

2006

Guide to the **CSQA COMMON BODY OF KNOWLEDGE**

▶ **Start**

Software
CERTIFICATIONS

Letter to CSQA Candidate

Dear CSQA Candidate:

Thank you for your interest in the Certified Software Quality Analyst (CSQA) Program. I am sure you already know the CSQA designation is quickly becoming the standard for IT software quality professionals around the world. Many companies are requiring certification for hiring or advancement. There have been over 27,000 IT professionals worldwide that have sought our professional certifications.

The CSQA Certification Board updates the CSQA Common Body of Knowledge (CBOK) approximately every three years. You can be assured that if you become competent in this material, you will be well prepared for today's software quality challenges. If you have extensive experience in software quality within IT, the examination should not be difficult for you. If your experience is minimal, or is limited to only certain areas of quality management, you should seek additional study material beyond those recommended in this guide.

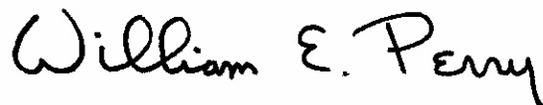
The CSQA certification examination is based upon the skill categories identified in the 2006 CSQA CBOK Outline. As such, this guide to the CBOK was designed for you to use as material in preparation for the CSQA exam. The examination presumes that you have had broad exposure to quality practices. It is expected that you have reviewed and read current literature available on software quality and testing. I urge you to read this guide carefully.

The guide to the 2006 CSQA Common Body of Knowledge has been released after careful review by software quality professionals and editors. As an organization based upon quality principals and theories, we welcome any feedback from you regarding content and structure. Please feel free to email your comments and suggestions to certify@softwarecertifications.org.

Best wishes in preparing for, and taking, the examination. For additional information regarding the 2006 CSQA CBOK, the CSQA Designation, or this program, please visit our Web site at www.softwarecertifications.org.

We also encourage you to become a part of the IT Quality community by visiting the Quality Assurance Institute Web site at www.qaiworldwide.org.

Sincerely,

A handwritten signature in black ink that reads "William E. Perry". The signature is written in a cursive, slightly slanted style.

William E. Perry, CSQA, CSTE
CEO
Quality Assurance Institute

Table of Contents

Introduction to the CSQA Program	1
Software Certification Overview	2
Program History	3
Why Become Certified?	3
Benefits of Becoming a CSQA	3
Meeting the CSQA Qualifications	7
Prerequisites for Candidacy	7
Code of Ethics	9
Submitting the Initial Application	12
Application-Examination Eligibility Requirements	13
Arranging to Sit and Take the Examination	14
Scheduling to Take the Examination	15
Receiving the Admission Ticket	16
Checking Examination Arrangements	16
Arriving at the Examination Site	16
How to Maintain Competency and Improve Value	16
Continuing Professional Education	17
Advanced CSQA Designations	17
Preparing for the CSQA Examination	21
Assess Your CSQA 2006 CBOK Competency	21
Complete the CSQA Skill Assessment Worksheet	22
Calculate Your CSQA CBOK Competency Rating	24
Understand the Key Principles Incorporated Into the Examination	25
Review the List of References	25
Initiate a Self-Study Program	26
Take the Sample Examination	26
CSQA 2006 Skill Assessment Worksheet	29

Skill Category 1

Quality Principles	41
Vocabulary of Quality	41
The Different Views of Quality	43
The Two Quality Gaps	44
Quality Attributes for an Information System	45
Quality Concepts and Practices	46
PDCA Cycle	48
Cost of Quality	49
Six Sigma Quality	54
Baselining and Benchmarking	54
Earned Value	55
Quality Control and Quality Assurance	55
Quality Control	55
Understanding and Using the Just-In-Time (JIT) Technique	56
Quality Assurance	57
Differentiating Between Quality Control and Quality Assurance	58
Quality Pioneers Approach to Quality	59
Dr. W. Edwards Deming	59
Philip Crosby	62
Dr. Joseph Juran	65
Total Quality Management	66

Skill Category 2

Quality Leadership	69
Leadership Concepts	69
Executive and Middle Management	69
Quality Champion	72
New Behaviors for Management	73
Empowerment of Employees	81
Quality Management Infrastructure	81
Quality Council	83
Management Committees	83
Teams and Work Groups	84
Process Improvement Teams	94
Quality Environment	95
The Six Attributes of an Effective Quality Environment	95
Setting the Proper "Tone" at the Top	105
Code of Ethics and Conduct	106
Open Communications	106

Implementing a Mission, Vision, Goals, Values, and a Quality Policy 112
 Monitoring Compliance to Organizational Policies and Procedures 116
 Enforcement of Organizational Policies and Procedures 120

Skill Category 3

Quality Baselines 125

 Quality Baseline Concepts 125

 Baselines Defined 125

 Types of Baselines 126

 Conducting Baseline Studies 126

 Methods Used for Establishing Baselines 130

 Customer Surveys 130

 Benchmarking to Establish a Baseline Goal 131

 Assessments against Management Established Criteria 133

 Assessments against Industry Models 136

 Model and Assessment Fundamentals 136

 Purpose of a Model 136

 Types of Models (Staged and Continuous) 137

 Model Selection Process 137

 Using Models for Assessment and Baselines 138

 Industry Quality Models 139

 Software Engineering Institute Capability Maturity Model Integration (CMMI®) 140

 Malcolm Baldrige National Quality Award (MBNQA) 143

 ISO 9001:2000 148

 ISO/IEC 12207: Information Technology – Software Life Cycle Processes 152

 ISO/IEC 15504: Process Assessment 156

 Post-Implementation Audits 164

Skill Category 4

Quality Assurance 165

 Establishing a Function to Promote and Manage Quality 165

 The Challenges of Implementing a Quality Function 167

 How the Quality Function Matures Over Time 168

 Support in Corporate Quality Management Environment 172

 Implementing an IT Quality Function 173

 Quality Tools 185

 Management Tools 186

 Statistical Tools 197

 Presentation Tools 209

 Process Deployment 212

Getting Buy-In for Change through Marketing	213
The Formula for Effective Behavior Change	215
The Deployment Process	216
Critical Success Factors for Deployment	223
Internal Auditing and Quality Assurance	224
Types of Internal Audits	224
Differences in Responsibilities	225
Skill Category 5	
Quality Planning	227
Planning Concepts	227
The Management Cycle	228
The Planning Cycle	229
Integrating Business and Quality Planning	231
The Fallacy of Having Two Separate Planning Processes	231
Planning Should be a Single IT Activity	232
Prerequisites to Quality Planning	233
The Planning Process	234
Planning Process Overview	234
The Six Basic Planning Questions	236
The Common Activities in the Planning Process	238
Planning to Mature IT Work Processes	240
QAI Model and Approach to Mature IT Work Processes	242
How to Plan the Sequence for Implementing Process Maturity	270
Skill Category 6	
Define, Build, Implement and Improve Work Processes	281
Process Management Concepts	281
Definition of a Process	281
Why Processes are Needed	282
Process Workbench and Components	283
Process Categories	285
The Process Maturity Continuum	286
How Processes are Managed	288
Process Template	289
Process Management Processes	289
Planning Processes	291
Do Processes	294
Check Processes	297
Act Processes	302

Skill Category 7

Quality Control Practices	309
Testing Concepts	309
The Testers' Workbench	309
Test Stages	310
Independent Testing	311
Static versus Dynamic Testing	312
Verification versus Validation	312
Stress versus Volume versus Performance	316
Test Objectives	316
Reviews and Inspections	317
Developing Testing Methodologies	320
Acquire and Study the Test Strategy	320
Determine the Type of Development Project	321
Determine the Type of Software System	321
Determine the Project Scope	323
Identify the Tactical Risks	324
Determine When Testing Should Occur	324
Build the System Test Plan	325
Build the Unit Test Plans	325
Verification and Validation Methods	325
Management of Verification and Validation	325
Verification Techniques	326
Validation Techniques	327
Structural and Function Testing	331
Software Change Control	332
Software Configuration Management	332
Change Control Procedures	333
Defect Management	333
Defect Management Process	333
Defect Reporting	334
Severity versus Priority	335
Using Defects for Process Improvement	337

Skill Category 8

Metrics and Measurement	339
Measurement Concepts	339
Standard Units of Measure	339
Metrics	340
Objective and Subjective Measurement	340

Types of Measurement Data	341
Measures of Central Tendency	342
Attributes of Good Measurement	342
Using Quantitative Data to Manage an IT Function	343
Key Indicators	344
Measurement in Software	344
Product Measurement	345
Process Measurement	348
Variation and Process Capability	348
The Measurement Program	349
Installing the Measurement Program	350
Common and Special Causes of Variation	353
Variation and Process Improvement	355
Process Capability	356
Risk Management	357
Defining Risk	357
Characterizing Risk	358
Managing Risk	358
Software Risk Management	365
Risks of Integrating New Technology	366
Implementing a Measurement Program	367
The Need for Measurement	367
Prerequisites	368
Skill Category 9	
Internal Control and Security	371
Principles and Concepts of Internal Control	371
Internal Control and Security Vocabulary and Concepts	372
Preventive, Detective, and Corrective Controls	377
Risk and Internal Control Models	386
COSO Enterprise Risk Management (ERM) Model	387
COSO Internal Control Framework Model	389
CobiT Model	392
Building Internal Control	392
Perform Risk Assessment	392
Building Adequate Security	395
Where Vulnerabilities in Security Occur	396
Establishing a Security Baseline	400
Security Awareness Training	411
Security Practices	418

Skill Category 10

Outsourcing, COTS and Contracting Quality 421

- Quality and Outside Software 421
 - Purchased COTS Software 422
 - Outsourced Software 422
- Selecting COTS Software 425
 - Assure Completeness of Needs Requirements 425
 - Define Critical Success Factor 426
 - Determine Compatibility with Hardware, Operating System, and
Other COTS Software 427
 - Assure the Software can be Integrated into Your Business System Work Flow 429
 - Demonstrate the Software in Operation 430
 - Evaluate People Fit 432
 - Acceptance Test the Software Process 432
- Selecting Software Developed by Outside Organizations 432
 - Contracting Life Cycle 432
 - Developing Selection Criteria 438
- Contracting for Software Developed by Outside Organizations 439
 - Contract Negotiations 439
- Operating for Software Developed by Outside Organizations 444
 - Acceptance Testing 444

How to Take the CSQA Examination 451

- CSQA Examination Overview 451
- Guidelines to Answer Questions 452
- Sample CSQA Examination 455

Appendix A

Vocabulary 479

Appendix B

References 495

Introduction to the CSQA Program

The Certified Software Quality Analyst (CSQA) program was developed by leading software quality professionals as a means of recognizing software quality analysts who demonstrate a predefined level of quality assurance competency. The CSQA program is directed by an independent Certification Board and administered by the Quality Assurance Institute (QAI). The program was developed to provide value to the profession, the individual, the employer, and co-workers.

The CSQA certification entails an aggressive educational program that tests the level of competence in the principles and practices of quality assurance and control in the Information Technology (IT) profession. These principles and practices are defined by the Certification Board as the Common Body of Knowledge (CBOK). The Certification Board will periodically update the CBOK to reflect changing software quality assurance and control, as well as changes in computer technology. These updates should occur approximately every three years.

<i>Software Certification Overview</i>	<i>2</i>
<i>Meeting the CSQA Qualifications</i>	<i>7</i>
<i>Arranging to Sit and Take the Examination</i>	<i>14</i>
<i>How to Maintain Competency and Improve Value</i>	<i>16</i>

Be sure to check the Software Certifications web site for up-to-date information on the CSQA program and examination sites and schedules, and What's New:

www.softwarecertifications.org

Using this product does not constitute, nor imply, the successful passing of the CSQA certification examination.

Software Certification Overview

Software Certification is recognized worldwide as the standard for IT quality assurance professionals. Certification is a big step; a big decision. Certification identifies an individual as a quality assurance leader and earns the candidate the respect of colleagues and managers. It is formal acknowledgement that the IT recipient has an overall understanding of the disciplines and skills represented in a comprehensive Common Body of Knowledge (CBOK) for a respective software discipline.

The CSQA program demonstrates the following objectives to establish standards for initial qualification and continuing improvement of professional competence. This certification program helps to:

1. Define the tasks (skill categories) associated with software quality assurance duties in order to evaluate skill mastery.
2. Demonstrate an individual's willingness to improve professionally.
3. Acknowledge attainment of an acceptable standard of professional competency.
4. Aid organizations in selecting and promoting qualified individuals.
5. Motivate personnel having software quality assurance responsibilities to maintain their professional competency.
6. Assist individuals in improving and enhancing their organization's software quality programs (i.e., provide a mechanism to lead a professional).

In addition to CSQA, Software Certifications also offer the following software certifications. See "How to Maintain Competency and Improve Value" on page 15 for more information on the certifications for advanced and master levels.

Software Quality Analysts

- Certified Software Quality Analyst (CSQA)
- Advanced Software Quality Analyst (ASQA)
- Master Software Quality Analyst (MSQA)

Software Testers

- Certified Software Tester (CSTE)
- Advanced Software Tester (ASTE)
- Master Software Tester (MSTE)

Software Project Manager

- Certified Software Project Manager (CSPM)

One or more of these certifications is frequently a prerequisite for promotion or acquiring a new position. See www.qaiworldwide.org and www.softwarecertifications.org for detailed information on all software certifications available including:

- Preparation Courses
- Examination Schedules
- Conferences and Seminars
- In-house Training Courses

Contact Us

Software Certifications
Phone: (407)-472-8100
Fax: (407)- 398-6817
CSQA questions? E-mail: certify@softwarecertifications.org

Program History

QAI was established in 1980 as a professional association formed to represent the software quality assurance industry. The first certification began development in 1985 and the first formal examination process was launched in 1990. Today, Software Certifications, administered by QAI, is global. Since its inception, Software Certifications has certified over 20,000 IT professionals in Australia, Barbados, Belgium, Bermuda, Brazil, Canada, China, Egypt, Hong Kong, India, Israel, Korea, Mexico, New Zealand, Puerto Rico, Saudi Arabia, Singapore, South Africa, United Kingdom, United Arab Emirates, and the United States.

Why Become Certified?

As the IT industry becomes more competitive, management must be able to distinguish professional and skilled individuals in the field when hiring. Certification demonstrates a level of understanding in carrying out software testing principles and practices that management can depend upon.

Acquiring the designation of CSQA indicates a professional level of competence in software quality assurance. CSQAs become members of a recognized professional group and receive recognition for their competence by businesses and professional associates, potentially more rapid career advancement, and greater acceptance in the role as advisor to management.

Benefits of Becoming a CSQA

As stated above, the CSQA program was developed to provide value to the profession, the individual, the employer, and co-workers. The following information is data collected from CSQAs

in the IT industry – a real testimonial to the benefits and reasons to make the effort to become a CSQA.

Value Provided to the Profession

Software quality assurance is often viewed as a software project task, even though many individuals are full-time quality assurance professionals. The CSQA program was designed to recognize software quality assurance professionals by providing:

- **Common Body of Knowledge (CBOK)**
The Certification Board defines the skills upon which the software quality assurance certification is based. The current CBOK includes 10 skill categories fully described in this preparation guide – see Skill Category 1 through Skill Category 10.
- **Examination Process to Evaluate Competency**
The successful candidate must pass a four-part examination that is based on the CBOK. You must receive a grade of 75%, or greater on each part. Only 31% of the pre-qualified applicants pass the examination the first time, making this a prestigious certification to obtain. See “How to Take the CSQA Examination” for a sample examination and answers to help you prepare for the actual examination.
- **Code of Ethics**
The successful candidate must agree to abide by a professional Code of Ethics as specified by the Certification Board. See “Code of Ethics” on page 9 for an explanation of the ethical behaviors expected of all certified professionals.

Value Provided to the Individual

The individual obtaining the CSQA certification receives the following values:

- **Recognition by Peers of Personal Desire to Improve**
Approximately, eighty percent (80%) of all CSQAs stated that a personal desire for self-improvement and peer recognition was the main reason for obtaining the CSQA certification. Fifteen percent (15%) were required by their employer to sit for the examination, and 10% were preparing themselves for an improved quality-related position.

Many CSQAs indicated that while their employer did not require CSQA certification, it was strongly encouraged.
- **Increased Confidence in Personal Capabilities**
Eighty-five percent (85%) of the CSQAs stated that passing the examination increased their confidence to perform their job more effectively. Much of that confidence came from studying for the examination.

- Recognition by IT Management for Professional Achievement

Most CSQAs stated that their management greatly respects those who put forth the personal effort needed for self-improvement. IT organizations recognize and reward individuals in the following ways:

- Thirteen percent (13%) received an immediate average one-time bonus of \$610, with a range of \$250 to \$2,500.
- Twelve percent (12%) received an immediate average salary increase of 10%, with a range of 2% to 50%.

Non-monetary recognitions were:

- Thirty-six percent (36%) were recognized in staff meetings.
- Twenty percent (20%) in newsletters or email.
- Many received rewards, management visits or calls, and lunch with the boss.

Within the first 18 months after receipt of the CSQA certification, of the successful candidates:

- Twenty-seven percent (27%) received an average salary increase of 23%, with a range of 2% to 100%.
- Twenty-three percent (23%) were promoted, 25% received a better assignment and 13% a new assignment.

Value Provided to the Employer

With the need for increased software quality and reliability, employing CSQAs provides value in these ways:

Increased Confidence by IT Users and Customers

IT users and customers expressed confidence in IT to effectively build or acquire software when certified quality assurance practitioners were involved.

Improved Processes to Build/Acquire/Maintain, Operate and Measure Software

CSQAs use their knowledge and skills to continuously improve the IT work processes. CSQAs know what to measure, how to measure it, and then prepare an analysis to aid in the decision-making process.

Independent Assessment of Quality Assurance Competencies

The CSQA program is directed by a Certification Board of independent quality assurance experts. Through examination and recertification, they provide an independent assessment of the CSQA's quality assurance competencies, based on a continuously strengthening Common Body of Knowledge for quality assurance practitioners.

Quality Assurance Competencies Maintained Through Recertification

Yesterday's quality assurance competencies are inadequate for today's challenges. CSQA recertification is a process that helps assure the CSQA's skills remain current. The

recertification process requires CSQAs to obtain 40 hours of quality assurance related training per year in topics specified by the Certification Board.

From an IT director's perspective, this is employee-initiated quality assurance training. Most, if not all CSQAs, do this training during their personal time. IT organizations gain three benefits from CSQA recertification: 1) employees initiate improvement; 2) quality assurance practitioners obtain competencies in quality assurance methods and techniques; and 3) employees train during personal time.

Value Provided to Co-Workers

The drive for self-improvement is a special trait that manifests itself in providing these values to co-workers:

Mentoring the Testing Staff

Forty-five percent (45%) of the CSQAs mentor their testing colleagues by conducting training classes; encouraging staff to become certified; and acting as a resource to the staff on sources of IT quality related information.

Testing Resource to "IT" Staff

CSQAs are recognized as experts in quality assurance and are used heavily for advice, counseling, and for recommendations on software construction and testing.

Role Model for Quality Assurance Practitioners

CSQAs are the IT role models for individuals with quality responsibilities to become more effective in performing their job responsibilities.

How to Improve Quality Assurance Effectiveness Through CSQA Certification

A "driver" for improved IT effectiveness is the integration of the CSQA certification program in your "IT" career development plan. This can be accomplished by:

- Creating an awareness of the CSQA Program and its benefits to your quality assurance practitioners.
- Requiring or encouraging your quality assurance practitioners to become certified.
- Recognizing and rewarding successful candidates.
- Supporting recertification as a means of maintaining quality assurance competency.

QAI, as CSQA program administrators, will assist you in this effort.
See www.qaiworldwide.org for detailed information.

Meeting the CSQA Qualifications

To become certified as a Certified Software Quality Analyst, every candidate must first meet these qualifications:

1. Satisfy all of the prerequisites required prior to applying for candidacy – educational and professional prerequisites including non-U.S. prerequisites, recommendations for preparing for the examination, and understanding what will be expected once you are a CSQA.
2. Subscribe to the Code of Ethics as described on page 9.
3. Submit a completed Certification Candidacy Application. See “Submitting the Initial Application” on page 11 for information on all the materials needed to submit your application.

Prerequisites for Candidacy

Before you submit your application, first check that you satisfy the educational and professional prerequisites described below and understand what is expected of the CSQA after certification.

Educational and Professional Prerequisites

To qualify for candidacy, each applicant must meet one of three credentials:

1. A bachelor's degree from an accredited college-level institution.
2. An associate degree and two years of experience in the information services field.

OR

3. Six years of experience in the information services field.

Non-U.S. Prerequisites

Educational requirements for Software Certifications are stated following the terms, customs, and requirements typically encountered in the United States. However, authority has been given to specific organizations sponsoring the examination process outside the United States to examine and modify educational and experience criteria within their countries. Each country's criteria will be based on the following framework:

- Candidates should possess qualifications equal to other professionals of similar status.
- Candidates should possess the superior knowledge and skills needed to carry out all designated responsibilities in a preeminent manner.
- Candidates' education and experience must be broadly based to meet a wide range of responsibilities and expectations.

- Successful candidates must be able to execute suitable quality assurance principles and practices in an array of diverse assignments and clearly communicate appropriate conclusions and recommendations.

Note: When submitting academic qualifications, the candidate must ensure that the materials are in sufficient detail so that the Software Certifications Board can determine equivalency. The Board is the final judge of acceptability of any alternative educational or experience-based criteria submitted by any applicant.

Expectations of the CSQA

Knowledge within a profession doesn't stand still. Having passed the CSQA examination, a certificant has demonstrated knowledge of the designation's CBOK at the point in time of the examination. In order to stay current in the field as knowledge and techniques mature, the certificant must be actively engaged in professional practice, and seek opportunities to stay aware of, and learn, emerging practices.

The CSQA is required to submit 120 credit hours of Continuing Professional Education (CPE) every three years to maintain certification or take an examination for recertification. Any special exceptions to the CPE requirements are to be directed to the Certification Director. Certified professionals are generally expected to:

- Attend professional conferences to stay aware of activities and trends in the profession.
- Take education and training courses to continually update skills and competencies.
- Develop and offer training to share knowledge and skills with other professionals and the public.
- Publish information in order to disseminate personal, project, and research experiences.
- Participate in the profession through active committee memberships and formal special interest groups.

The CSQA is expected not only to possess the skills required to pass the CSQA examination but also to be a change agent: someone who can change the culture and work habits of individuals (or someone who can act in an advisory position to upper management) to make quality in software quality assurance happen.

Professional Skill Proficiency Responsibilities

In preparing yourself for the profession of IT software quality assurance and to become more effective in your current job, you need to become aware of the three C's of today's workplace:

- Change – The speed of change in technology and in the way work is performed is accelerating. Without continuous skill improvement, you will become obsolete in the marketplace.
- Complexity – Information technology is becoming more complex, not less complex. Thus, achieving quality, with regard to software quality assurance in the information technology environment, will become more complex. You must update your skill proficiency in order to deal with this increased complexity.

- **Competition** – The ability to demonstrate mastery of multiple skills makes you a more desirable candidate for any professional position. While hard work does not guarantee you success, few, if any, achieve success without hard work. CSQA certification is one form of achievement. CSQA certification is proof that you've mastered a basic skill set recognized worldwide in the information technology arena.

Develop a Lifetime Learning Habit

Become a lifelong learner in order to perform your current job effectively and remain marketable in an era of the three C's. You cannot rely on your current knowledge to meet tomorrow's job demands. The responsibility for success lies within your own control.

Perhaps the most important single thing you can do to improve yourself professionally and personally is to develop a lifetime learning habit.
REMEMBER: If it is going to be—it's up to me.

Code of Ethics

An applicant for certification must subscribe to the following Code of Ethics that outlines the ethical behaviors expected of all certified professionals. Software Certifications includes processes and procedures for monitoring certificant's adherence to these policies. Failure to adhere to the requirements of the Code is grounds for decertification of the individual by the Software Certifications Board.

Purpose

A distinguishing mark of a profession is acceptance by its members of responsibility to the interests of those it serves. Those certified must maintain high standards of conduct in order to effectively discharge their responsibility.

Responsibility

This Code of Ethics is applicable to all certified by Software Certifications. Acceptance of any certification designation is a voluntary action. By acceptance, those certified assume an obligation of self-discipline beyond the requirements of laws and regulations.

The standards of conduct set forth in this Code of Ethics provide basic principles in the practice of information services quality assurance. Those certified should realize that their individual judgment is required in the application of these principles.

Those certified shall use their respective designations with discretion and in a dignified manner, fully aware of what the designation denotes. The designation shall also be used in a manner consistent with all statutory requirements.

Those certified who are judged by the Software Certifications Board to be in violation of the standards of conduct of the Code of Ethics shall be subject to forfeiture of their designation.

Professional Code of Conduct

Software Certifications certificate holders shall:

1. Exercise honesty, objectivity, and diligence in the performance of their duties and responsibilities.
2. Exhibit loyalty in all matters pertaining to the affairs of their organization or to whomever they may be rendering a service. However, they shall not knowingly be party to any illegal or improper activity.
3. Not engage in acts or activities that are discreditable to the profession of information services quality assurance or their organization.
4. Refrain from entering any activity that may be in conflict with the interest of their organization or would prejudice their ability to carry out objectively their duties and responsibilities.
5. Not accept anything of value from an employee, client, customer, supplier, or business associate of their organization that would impair, or be presumed to impair, their professional judgment and integrity.
6. Undertake only those services that they can reasonably expect to complete with professional competence.
7. Be prudent in the use of information acquired in the course of their duties. They shall not use confidential information for any personal gain nor in any manner that would be contrary to law or detrimental to the welfare of their organization.
8. Reveal all material facts known to them that, if not revealed, could either distort reports of operation under review or conceal unlawful practices.
9. Continually strive for improvement in their proficiency, and in the effectiveness and quality of their service.
10. In the practice of their profession, shall be ever mindful of their obligation to maintain the high standards of competence, morality, and dignity promulgated by this Code of Ethics.
11. Maintain and improve their professional competency through continuing education.
12. Cooperate in the development and interchange of knowledge for mutual professional benefit.
13. Maintain high personal standards of moral responsibility, character, and business integrity.

Grounds for Decertification

Revocation of a certification, or decertification, results from a certificant failing to reasonably adhere to the policies and procedures of Software Certifications as defined by the Software Certifications Board. The Board may revoke certification for the following reasons:

- Falsifying information on the initial application and/or a CPE reporting form,
- Failure to abide by and support the Software Certifications Code of Ethics,
- Failure to submit the required continuing education credits toward recertification as required, or
- Failure to submit the required recertification fees as required.

Upon revocation, the certificant is requested to return their current certification credentials. A certificant may appeal a revocation at any time by communicating, in writing, directly with the Board.

Submitting the Initial Application

A completed Certification Candidacy Application must be submitted for entrance to Software Certifications as a candidate for any particular certification. Software Certifications strongly recommends that you submit your application only if you are prepared to sit and pass the CSQA examination. Submit the application **only if you have**:

- Satisfied all of the prerequisites for candidacy as stated on page 7.
- Subscribed to the Code of Ethics as described on page 9.
- Reviewed the CBOK and identified those areas that require additional studying.

The entire CBOK is provided in Skill Category 1 through Skill Category 10. A comprehensive list of related references is listed in Appendix B.

- Current experience in the field covered by the certification designation.
- Significant experience and breadth to have mastered the basics of the entire CBOK.
- Prepared to take the required examination and therefore ready to schedule and take the examination.

It should not be submitted by individuals who:

- Have not met all of the requirements stated above.
- Are not yet working in the field but who have an interest in obtaining employment in the field.
- Are working in limited areas of the field but would like to expand their work roles to include broader responsibilities.
- Are working in IT but have only marginal involvement or duties related to the certification.

- Are interested in determining if this certification program will be of interest to them.

Candidates for certification who rely on only limited experience, or upon too few or specific study materials, typically do not successfully obtain certification. Many drop out without ever taking the examination. Fees in this program are nonrefundable. **Do not apply unless you feel confident that your work activities and past experience have prepared you for the examination process.**

Applicants already holding a certification from Software Certifications must still submit a new application when deciding to pursue an additional certification. For example, an applicant already holding a CSTE or CSPM certification must still complete the application process if pursuing the CSQA certification.

All application forms and required fees must be filed with the Director of Certification **at least 60 calendar days** prior to any examination date selected. The candidate must sign the application form agreeing to support and abide by the Software Certifications Code of Ethics. Applications will not be processed if they are incomplete, incorrectly completed, or fees have not been paid. See www.softwarecertifications.org for application fee information. The candidate has sole responsibility to ensure that materials are submitted in a timely and orderly manner.

When sending an application, please allow two weeks for processing. There is no need to contact the administrative office during this period to check on the status of the application. In fact, to protect the integrity of the examination and certification processes, all correspondence related to certification policies and procedures must be in writing, using e-mail, fax, or first-class postal service. Information and status obtained through telephone conversations with the administering body shall be considered unofficial and off-the-record.

Correcting Application Errors

The accuracy and correctness of applications, documentation, or payments are the responsibility of the applicant. Incomplete or erroneous paperwork is returned to the applicant for correction and resubmission. Common defects requiring paperwork to be returned to the applicant include:

- Required information is missing.
- Incorrect form was used.
- Payment is missing or invalid.
- Unable to read required portions of application.
- Required signature is not present.
- Application received too late to be processed for selected examination.

Once corrected, materials can be resubmitted. This correction cycle does not waive the requirement that all processing be completed at Software Certifications at least 60 days before any scheduled examination. Applicants are strongly advised to not delay submission of materials until close to that deadline.

Submitting Application Changes

It is critical that candidates submit changes to their candidacy application and keep their program records up to date. Many candidates change their residence or job situations during their certification candidacy. Others change their name as a result of marriage or divorce. If any such changes occur, it is the candidate's responsibility to notify the certification administrator using the Change of Records Form.

Application-Examination Eligibility Requirements

The candidate must take the initial exam within 12 months after acceptance. After the 12-month period, the candidate must resubmit the application, supporting documents, and any additional fees that may have been incurred. **A second or third sitting, if required, must be completed within 24 months of acceptance of the original application.** After the 24-month period, the candidate must reapply for candidacy to begin the process again.

The candidate may withdraw from the CSQA program at any time by submitting a Candidate Withdrawal Form to the certification administrator.

Candidates for certification must pass a four-part written examination in order to obtain certification. The examination tests the candidate's knowledge and practice of the competency areas defined in the CBOOK. **Candidates who do not successfully pass the examination may resit for the examination up to two times** by submitting an Examination Retake Application (see Filing a Retake Application below) and paying all required fees. Subsequent additional examination efforts require reinitiating the entire application process.

The Software Certifications Board requires unsuccessful candidates to wait six months or more between examination sittings. Candidates who rapidly resit for examination parts are rarely successful. Adequate study and learning time needs to be spent in order to resit for missed examination parts successfully.

Technical knowledge becomes obsolete quickly; therefore the board has established these eligibility guidelines. The goal is to test on a consistent and comparable knowledge base worldwide. The eligibility requirements have been developed to encourage candidates to prepare and pass all portions of the examination in the shortest time possible.

Filing a Retake Application

Candidates who have taken the examination but not passed are eligible to resit for the examination on up to two subsequent examination dates. **Candidates must wait at least six months between examination sittings.** Candidates who rapidly resit for examination parts are rarely successful. Adequate study and learning time needs to be spent in order to resit for missed examination parts successfully.

A written Examination Retake Application must be submitted for each desired retake. As with the initial application, the application to reschedule and associated fees must be filed with the Director

of Certification **at least 60 calendar days** before any examination date is selected. See www.softwarecertifications.org for application fee information.

Arranging to Sit and Take the Examination

When you have met all of the prerequisites as described above, you are ready to arrange to sit (or schedule) and take the CSQA examination. See “How to Take the CSQA Examination” for information on what you need to do once you have scheduled the examination. This section also includes an example test with answers.

To schedule the CSQA examination, every candidate must:

- Satisfy all of the qualifications as described in “Meeting the CSQA Qualifications” starting on page 7. Be certain that you are prepared and have studied the CBOOK, the vocabulary in Appendix A, and the references in Appendix B.
- Schedule to take the examination. If you've studied enough that you feel you can commit to a specific examination date, visit www.softwarecertifications.org for dates or call Software Certifications. CSQA examinations are administered in various cities in the United States and all over the world. Submit a complete Examination Selection Form.
- Follow up on your examination schedule. After scheduling your examination you should receive an admission ticket to the specific examination you indicated on your Examination Selection Form. See Receiving the Admission Ticket on page 15. Check www.softwarecertifications.org for your specific scheduled examination during the days leading up to the examination sitting for any changes to the schedule.
- Be sure to arrive at the examination early. See “Arriving at the Examination Site” on page 15 for a few tips, and what happens if you do not show up as scheduled.

Scheduling to Take the Examination

When you believe you are close to being prepared to take the examination, schedule to take the examination. To select an examination date and location that meets your needs submit an Examination Selection Form. Public certification examinations are scheduled periodically throughout the United States. A complete up-to-date schedule is on the Software Certifications web site; see **Current Examination Schedule at www.softwarecertifications.org**.

Examination seating is limited, and seats are assigned on a first-come-first-served basis. **An Examination Selection Form must be submitted at least 60 days before the selected examination date in order to reserve a seat in the selected examination.** The earlier you apply the better chances of reserving a seat. The examination schedule can change on a weekly basis, so check www.softwarecertifications.org for any changes.

Examinations are held primarily by QAI Federation chapters, at major QAI conference programs, and by local QAI affiliates around the world. It is recommended that you contact the Director of Certification for site requirements, fees, and other details.

The certification examinations are typically available in Australia, Canada, Hong Kong, India, New Zealand, Saudi Arabia, Singapore, South Africa, United Arab Emirates, and the United States. As the worldwide acceptance of Software Certifications designations continues to grow, more locations will be hosting the exam. Please contact www.softwarecertification.org to inquire about examination locations.

Rescheduling the Examination Sitting

From time to time, candidates need to reschedule their intended examination date. This is known as a *deferral*, and is accomplished using the **Examination Deferral Form that must be submitted to the certification administrator at least 30 days before the originally scheduled examination**. If done in this manner, the **Examination Selection Form can be used to schedule the new examination as long as it is received at least 60 days before the new requested date**.

Deferrals received within 30 days of an examination date cannot be processed because examination materials have already been sent to the field. These candidates are considered "no shows" on the day of the examination and must use the Examination Retake Application in order to schedule a new examination date. As with the initial application, **the Examination Retake Application and associated fees must be filed with the Director of Certification at least 60 days before any examination date is selected**.

Receiving the Admission Ticket

Each candidate should receive an Examination Admission Ticket. You should bring this ticket to the examination site along with photo identification to gain entry. When the ticket is received, verify the examination information to assure that you have been scheduled for the examination selected, and that your contact information is all correct. If not received three weeks before a scheduled sitting, check the Current Examination Schedule for possible changes, or contact Software Certifications via e-mail for confirmation or correction.

Checking Examination Arrangements

Candidates are strongly encouraged to check www.softwarecertifications.org for your specific scheduled examination during the days leading up to the examination sitting. While Software Certifications makes every possible effort to provide examinations as scheduled, last minute changes have been sometimes unavoidable in the past. Previous disruptions have included inclement weather and building closures. The Current Examination Schedule is kept as up-to-date as possible when such situations occur.

Arriving at the Examination Site

Candidates should arrive at the examination location at least 30 minutes before the scheduled start time of the examination. Candidates must have their Confirmation Letter and photo identification with them in order to register and gain admission to the examination.

No-shows

Candidates who fail to appear for a scheduled examination – initial or retake – automatically fail the examination and must submit the Examination Retake Application to apply for a new examination date. Candidates who have filed a deferral after the 30-day advance deadline are considered to be no-shows as well.

How to Maintain Competency and Improve Value

Maintaining your personal competency is too important to leave to the soul discretion of your employer. In today's business environment you can expect to work for several different organizations, and to move to different jobs within your own organization. In order to be adequately prepared for these changes you must maintain your personal competency in your field of expertise.

Continuing Professional Education

Most professions recognize that a minimum of 40 hours of continuing professional education is required to maintain competency of your skills. There are many ways to get this training, including attending professional seminars and conferences, on-the-job training, attending professional meetings, taking e-learning courses, and attending professional association meetings.

You should develop an annual plan to improve your personal competencies. Getting 40 hours of continuing professional education will enable you to recertify your CSQA designation, but it will not necessarily improve your competencies. For example, you may get 24 hours CPE credit for attending a 3-day seminar, but if you're already competent in the seminar topic, it will not add to your personal capabilities.

The Common Body of Knowledge (CBOK) for the CSQA should be your guide for improving your personal competencies. A self-assessment of your competencies in the CBOK is provided in "CSQA 2006 Skill Assessment Worksheet." This assessment is designed to help you identify areas in which additional training would be beneficial to both you and your employer. After taking this competency assessment, you can use the results to create a personal plan of action for you to ensure that you maintain the necessary professional competency to prepare you for change and/or promotion.

Advanced CSQA Designations

You can use your continuing professional education plan to improve and demonstrate your value to your employer. You can obtain your professional education credits while applying for an advanced certification. Your employer may have difficulty assessing improved competencies attributable to the continuing professional education you are acquiring. However, if you can use that continuing education effort to obtain an advanced degree, you can demonstrate to your employer your increased value to the organization by acquiring an advanced degree.

There are two levels of advanced degrees you will be eligible for once you obtain your CSQA designation:

- **Advanced Software Quality Analyst (ASQA)**

This advanced designation is designed to demonstrate your knowledge of how to do the quality assurance tasks you may be assigned. The CSQA designation is focused much more on “what” you must know in order to practice quality assurance. The ASQA designation is designed for those who can demonstrate they know “how” to perform quality assurance tasks.

- **Master Software Quality Analyst (MSQA)**

This is the highest designation attainable in the IT quality assurance field. It is reserved for those who can demonstrate quality assurance qualities and professional responsibilities.

The drivers for improving performance in IT are the quality assurance and quality control (testing) professionals. Dr. W. Edward Deming recognized this “do-check” partnership of quality professionals in his “14 points” as the primary means for implementing the change needed to mature. Quality control identifies the impediments to quality and quality assurance facilitates the fix. Listed below is the certification level, emphasis of each certification, and how you can demonstrate that competency.

What is the Certification Competency Emphasis?

CSQA

- Demonstrate competency in knowing what to do.
- Study for, and pass, a four-part examination developed by peers to evaluate the candidate’s knowledge of the principles and concepts incorporated into the CBOK, plus the ability to relate those principles and concepts to the challenges faced by IT organizations.

ASQA

- Demonstrate competency in knowing how to do it.
- Candidates must demonstrate their ability to develop real solutions to challenges in their IT organizations, by proposing a solution to a real-world problem. If accepted by the Certification Board, to develop and submit for evaluation the step-by-step solution the candidate developed for that IT challenge. This must be done for five CBOK categories. Each accepted solution will be awarded a certificate of competency for that CBOK category.

MSQA – Master Software Quality Analyst

- Demonstrate competency in knowing how to break through quality and productivity barriers.

- Candidates must demonstrate the ability to innovate beyond current practice in solving IT challenges, as well as, demonstrate public service in the IT Quality profession. (Note: this certification available starting in 2006.)

Figure 1 illustrates how you can improve your personal competencies.

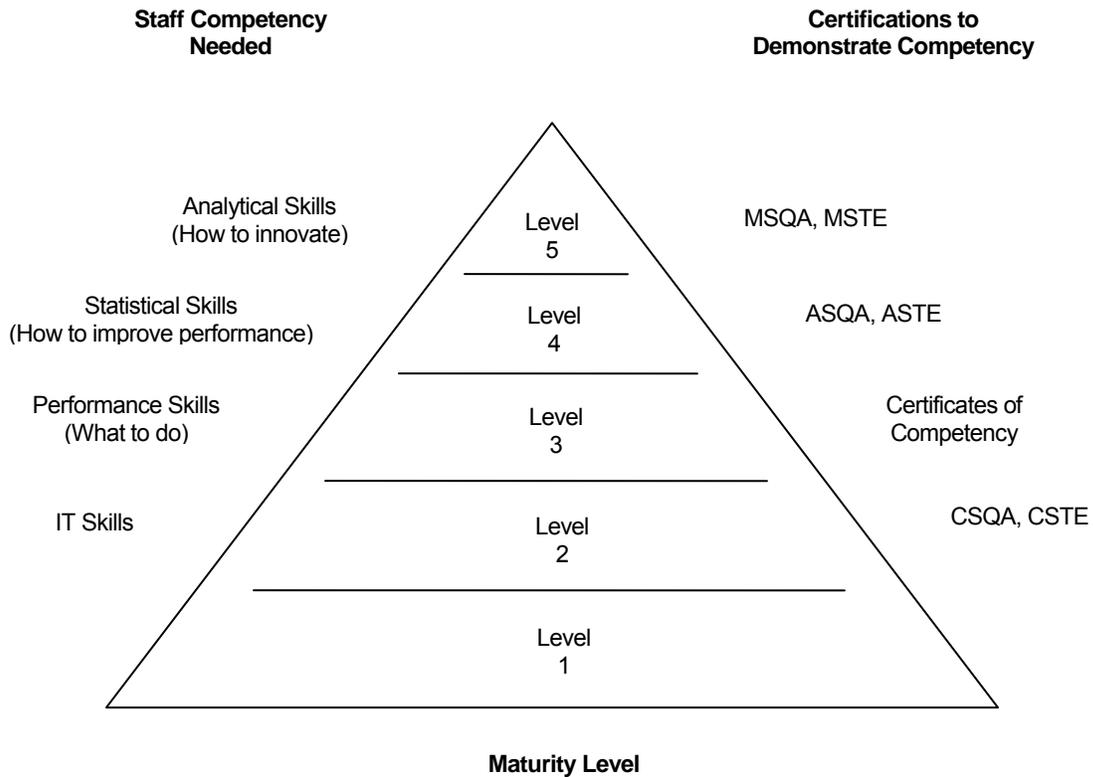


Figure 1. Maturing Your Professional Competencies

For more information on the type of training that is applicable toward your continuing professional education requirements, and information on the advanced quality assurance certifications and how to apply for them, visit www.softwarecertifications.org.

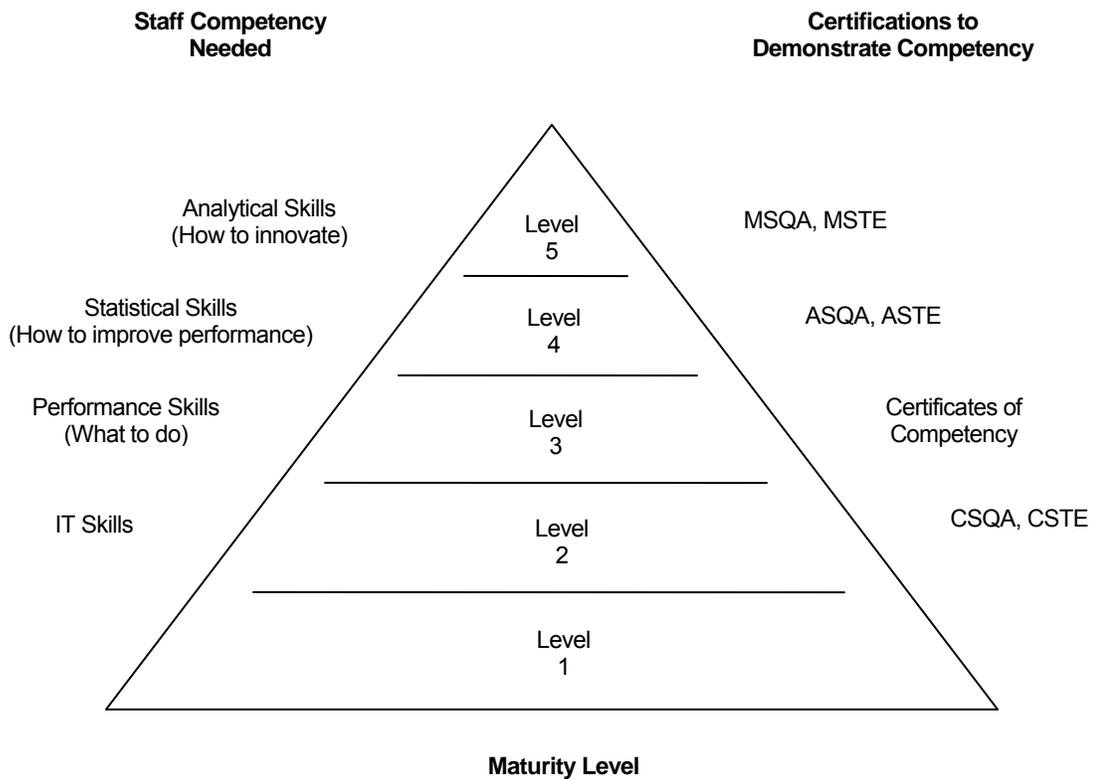


Figure 1. Maturing Your Professional Competencies

For more information on the type of training that is applicable toward your continuing professional education requirements, and information on the advanced quality assurance certifications and how to apply for them, visit www.softwarecertifications.org.

Preparing for the CSQA Examination

The CSQA examination is designed to evaluate your knowledge of the principles and practices of software quality analysis. The principles primarily will involve vocabulary. This is to ensure that you understand what quality in an IT function is attempting to accomplish. The second half of the examination is on the application of those principles. This is to ensure that you can recognize good software quality practices when they occur.

Preparing for any standardized examination should be considered a serious undertaking. Begin preparing and studying well in advance. Remember that the minimum requirement for submitting your application is **60 calendar days** prior to the exam date. When you know you will be applying for the examination, submit your application and fees and begin studying. Avoid “cramming,” as it is rarely beneficial in the long term. See the “Introduction” for detailed information on submitting your application.

<i>Assess Your CSQA 2006 CBOK Competency</i>	<i>21</i>
<i>Understand the Key Principles Incorporated Into the</i>	<i>25</i>
<i>Review the List of References</i>	<i>25</i>
<i>Initiate a Self-Study Program</i>	<i>26</i>
<i>Take the Sample Examination</i>	<i>26</i>

Assess Your CSQA 2006 CBOK Competency

The Common Body of Knowledge (CBOK) for the CSQA is in effect a job description for a world-class IT quality assurance analyst. The CSQA Certification Board has defined the skills within the CBOK as those skills that would enable an IT quality assurance analyst to perform the tasks needed to meet today’s IT quality challenges.

Many human resource organizations use the CSQA CBOK as the basis for writing job descriptions for IT quality assurance analysts. To properly prepare yourself to be proficient in the practice of IT quality assurance, you should develop a personal plan of action that would enable you to assess your competency in the 2006 CSQA CBOK. It is recognized that many software quality analysts do not need to be competent in all of the skill categories to fulfill their current job responsibilities.

The current CBOK includes 10 skill categories that are fully described in this guide:

- Skill Category 1 Quality Principles and Concepts
- Skill Category 2 Quality Leadership
- Skill Category 3 Quality Baselines (Assessments and Audits)
- Skill Category 4 Quality Assurance
- Skill Category 5 Quality Planning
- Skill Category 6 Define, Build, Implement and Improve Work Processes
- Skill Category 7 Quality Control Practices
- Skill Category 8 Metrics and Measurements
- Skill Category 9 Internal Control and Security
- Skill Category 10 Outsourcing, COTS, and Contracting Quality

Skill Categories 1-8 should be common to all quality-related assignments and therefore, most of the certification examination focuses on categories 1 through 8. However, you should have a basic knowledge of Skill Categories 9 and 10 to remain current of software quality competencies. Candidates are examined at high levels on categories 9 and 10.

Complete the CSQA Skill Assessment Worksheet

To assess your competency of the CSQA CBOK, complete the worksheet, “CSQA 2006 Skill Assessment Worksheet” starting on page 27. Follow these guidelines on how to use the worksheet to rate your competency and identify those areas that you need to better understand to successfully pass the CSQA examination:

1. Assess your competency of each skill listed on the worksheet. Carefully read each skill within the skill category. Based on your reading of the skill, assess your competency in one of the following three categories and place a checkmark (“✓”) in the appropriate column on the CSQA 2006 CBOK Competency Rating Table:

Not Competent – “None”

Either you do not understand this skill, or if you do understand it you do not know “what” is required to perform this skill. For example, you may know that an IT quality plan is needed, but you do not know what is included in an IT quality plan.

Some Competency – “Some”

This assessment means that you know “what” is needed to accomplish a specific skill. For example, you may know what is to be included within an IT quality plan, but you have never actually prepared an IT quality plan. In other words, you have book knowledge, but not how-to knowledge.

Fully Competent – “Full”

This assessment means that you not only know what is required to perform a specific skill, but you have actually used that skill in performing day-to-day work tasks. For example, you have written an IT quality plan.

Note that Skill Category 1 focuses on the vocabulary of IT quality assurance and the basic concepts on which the quality assurance profession is built. In assessing this category for a quality term such as *reliability* a “not competent” response means you could not define the term; a “some competency” response means you could define the term; and a “fully competent” response means that you use the term in the performance of your day-to-day work.

2. Study those skills you rated “None.” After you complete the assessment worksheet, you will have designated some of the skills included in the CBOK as: None, Some, and Full. The objective in preparing for the CSQA examination should be to have “some competency” in *all* of the skills within the CBOK. You need not be fully competent in any skill to qualify you to pass the CSQA examination.

Note that the CSQA designation focuses on individuals knowing “what to do” in order to effectively perform IT quality assurance. To provide maximum value to your employer, and to enable you to obtain either an Advanced Software Quality Assurance (ASQA) or Master Software Quality Assurance (MSQA) designation you need to be “fully competent” in most of the CBOK skills areas.

3. Reassess those skills you studied after a rating of “None.” If you now believe your rating changes to “Some,” then change your checkmark for the related skill on that category assessment table. Continue reassessing as you study.

Proceed *only* when you believe you are ready to submit your application for the CSQA certification examination.

Calculate Your CSQA CBOK Competency Rating

Follow these steps to calculate your competency rating for the CSQA 2006 CBOK. This rating will help you determine if you are ready to submit your application for the CSQA examination or if, and in what areas, you need further study in order to pass the examination. Use the CBOK Skill Category Competency Rating Table on page 38 to perform each step below.

1. Total the number of skills you have checked in each of the three columns for each skill category. Write your numbers in the space provided for each skill category on the worksheet. These are your competency rating totals for that skill category.
2. Transfer the three competency rating totals for each skill category to the corresponding column (“Full,” “Some,” and “None”) in the CSQA Skill Category Competency Ratings table provided.
3. Tally each column in the table to determine each Ratings Total.
4. Multiply each column by the indicated number to determine the Column Total.
5. Add the Column Totals together to determine the Sum of the Rows Total.
6. Divide the Sum of the Rows Total by 155 (the number of total skills in the CSQA 2006 CBOK) to determine your CSQA CBOK Competency Rating. This number will be between 1 and 3.

Now you are able to determine if you are ready to submit your application and take the certification examination or if you need further study. Use your CSQA 2006 CBOK Competency Rating from step 6 above and the following key to interpret your competency rating:

- The closer your score is to “3,” the more competent you are in software quality assurance.
- If your score is a “3,” you are a world-class software quality analyst and ready to submit your application.
- If your score is between “2” and “3”, you are a competent quality analyst and ready to submit your application.

See the “Introduction” for information on submitting your application for the CSQA 2006 certification examination.

- If your score is between “1” and “2”, you do not have the basic skills necessary to perform software quality assurance. Study those skills that you rated “None” and then reassess your skills.
- If your score is a “1”, you are not competent in the CBOK. Study those skills that you rated “None” and then reassess your skills.

Using this product does not constitute, nor imply, the successful passing of the CSQA certification examination.

Understand the Key Principles Incorporated Into the Examination

This step is to provide you some insight into what will be emphasized on the examination. This should not be used in place of the CBOK. It is intended to emphasize some of the key concepts included within the CBOK.

In studying these key principles, two guidelines should be followed:

- Learn the vocabulary.

A major part of the CSQA examination and a major part of being an effective quality analyst is to understand and use the quality analysis vocabulary. If you do not know the quality analysis vocabulary, study Appendix A, “Vocabulary,” before beginning any other CSQA examination preparatory activity. Note that understanding the vocabulary is essential to pass the examination.

- Learn how to apply the quality principles to everyday practice.

As you study the quality principles, think carefully how you would apply those principles to your day-to-day work challenges.

Review the List of References

Use the following lists of references to help you prepare for the CSQA examination:

- Appendix B of this preparation guide lists numerous books recommended in the quality analysis field.
- Software Certifications web site – www.softwarecertifications.org (click on Index and then click on Body of Knowledge, CSQA) lists references compiled by the Certification Board and used in preparing the examination.

It is each candidate's responsibility to stay current in the field and to be aware of published works and materials available for professional study and development. Software Certifications recommends that candidates for certification continually research and stay aware of current literature and trends in the field. The lists referenced above are suggestions; they are not intended to be all-inclusive.

Use these lists of references in the following ways:

- Search your library for availability. If you have these books in your reference library, company library, or ready access, set them aside for exam preparation.
- Use your assessment results (e.g., skills marked “Not Competent”) from the previous step to determine which books would help you build your skills in those areas. Note that while studying, look for principles as opposed to learning detailed how-to skills.
- Review the list of references from the perspective of the types of materials that might be included on the examination. The references give you insight into the topics that will be included on the examination.

Initiate a Self-Study Program

This guide contains a variety of skill areas designed to be representative of the types of skills needed by software quality analysts, and representative of the skill categories addressed in the CSQA examination. You may decide to start or join a self-study group in your area.

In developing a self-study program, you should:

- Assess your skills against the CSQA 2006 CBOK and complete the assessment worksheet.
- Study the key reference documents from the previous step. Use a representative sample of the reference books for study; if you do not have the specific reference book, use a similar book on the same topic.
- Attend formal courses, seminars, local quality-oriented chapter meetings, and quality conferences to gain a comprehension of the practice of quality assurance. Be sure to visit www.qaiworldwide.org for up-to-date information on courses, seminars, and conferences. QAI offers a preparation course for the CSQA.

Self-study becomes more effective if you can work with one or more other candidates for the examination. If no other candidates are available to form a study group, locate a CSQA to become your mentor during your self-study period.

Take the Sample Examination

We have provided a sample CSQA examination for you to use in your preparations. See “How to Take the CSQA Examination” for the following useful information:

- **CSQA Examination Overview** including how the test is structured and the number of questions, plus general information on how the test is administered at the test site.
- **Guidelines to Answer Questions** including useful steps to answer all questions, tips on responses to essay questions, and what to do if you do not know the answer to a question.

- **Sample CSQA Examination** including multiple-choice questions and essay questions. These give you examples of the types of questions on the examination. Also provided is an answer key to help you study and show you the types of essay responses expected.

CSQA 2006 Skill Assessment Worksheet

Assess Your Skills against the CSQA 2006 CBOK

The Skill Categories 1-8 should be common to all quality-related assignments and therefore, most of the certification examination focuses on categories 1 through 8. However, you should have a basic knowledge of Skill Categories 9 and 10 to remain current of software quality competencies. Candidates are examined at high levels on categories 9 and 10.

The 2006 Common Body of Knowledge for the software quality analyst certificate includes these ten skill categories:

Skill Category 1 – Quality Principles and Concepts

Skill Category 2 – Quality Leadership

Skill Category 3 – Quality Baselines

Skill Category 4 – Quality Assurance

Skill Category 5 – Quality Planning

Skill Category 6 – Define, Build, Implement and Improve Work Processes

Skill Category 7 – Quality Control Practices

Skill Category 8 – Metrics and Measurement

Skill Category 9 – Internal Control and Security

Skill Category 10 – Outsourcing, COTS, and Contracting Quality

See the “Introduction” for detailed instructions on how to use the worksheet and competency rating table.

Skill Category 1 – Quality Principles and Concepts

Before an organization can begin to assess the quality of its products and services, and identify opportunities for improvement, it first must have a working knowledge of quality principles and basic concepts. This category will test the CSQA candidate's ability to understand and apply these principles, which include the quality vocabulary, various ways of defining quality, key concepts, distinguishing between quality control and quality assurance, and the contributions of quality pioneers.

Skill Category 1 – Quality Principles and Concepts		Competency Rating		
Skill #	Skill Description	Full	Some	None
1.1	<i>Vocabulary of Quality</i> Understand the vocabulary of quality.			
1.2	<i>The Different Views of Quality</i> The two quality gaps			
1.3	Quality attributes for an information system			
1.4	<i>Quality Concepts and Practices</i> PDCA Cycle			
1.5	Cost of quality			
1.6	Six sigma quality			
1.7	Baselining and benchmarking			
1.8	Earned value			
1.9	<i>Quality Control and Quality Assurance</i> Understand quality control and quality assurance			
1.10	Understanding and using the Just-in-Time (JIT) Technique			
1.11	Differentiating between Quality Control and Quality Assurance			
1.12	<i>Quality Pioneers Approach to Quality</i> Includes Dr. W. Edwards Deming, Philip Crosby, and Dr. Joseph Juran.			
1.13	Total Quality Management			

Competency Rating Totals (total each "✓" in each column):

Skill Category 2 – Quality Leadership

The most important prerequisites for successful implementation of any major quality initiative are leadership and commitment from executive management. Management must create a work environment supportive of quality initiatives. It is management's responsibility to establish strategic objectives and build an infrastructure that is strategically aligned to those objectives. This category will cover the management processes used to establish the foundation of a quality-managed environment, as well as commitment, new behaviors, building the infrastructure, techniques, approaches and communications.

Skill #	Skill Category 2 – Quality Leadership Skill Description	Competency Rating		
		Full	Some	None
2.14	<i>Leadership Concepts</i> Executive and middle management commitment			
2.15	Quality Champion			
2.16	New Behaviors for Management – traditional management versus quality management, leadership, the importance of establishing mentoring relationships, and establishing trust			
2.17	Empowerment of employees			
2.18	<i>Quality Management Infrastructure</i> Quality council			
2.19	Management committees			
2.20	Teams and work groups			
2.21	Process improvement teams			
2.22	<i>Quality Environment</i> The six attributes of an effective quality environment			
2.23	Setting the proper “tone” at the top			
2.24	Code of ethics and conduct			
2.25	Open communication			
2.26	Implementing a mission, a vision, goals, values and a quality policy			
2.27	Monitoring compliance to organizational policies and procedures			
2.28	Enforcement of organizational policies and procedures			

Competency Rating Totals (total each “✓” in each column):

Skill Category 3 – Quality Baselines

Organizations need to establish baselines of performance for quality, productivity and customer satisfaction. These baselines are used to document current performance and document improvements by showing changes from a baseline. In order to establish a baseline, a model and/or goal must be established for use in measuring against to determine the baseline.

Skill #	Skill Category 3 – Quality Baselines Skill Description	Competency Rating		
		Full	Some	None
3.29	<i>Quality Baseline Concepts</i> Baselines defined			
3.30	Types of baselines			
3.31	Conducting baseline studies			
3.32	<i>Methods Used for Establishing Baselines</i> Customer surveys			
3.33	Benchmarking to establish a baseline goal			
3.34	Assessments against management established criteria			
3.35	Assessments against industry models			
3.36	<i>Model and Assessment Fundamentals</i> Purpose of a model			
3.37	Types of models (staged and continuous)			
3.38	Model selection process			
3.39	Using models for assessment and baselines			
3.40	<i>Industry Quality Models</i> Software Engineering Institute Capability Maturity Model/CMMI			
3.41	Malcolm Baldrige National Quality Award			
3.42	ISO 9001:2000			
3.43	ISO/IEC 12207			
3.44	ISO/IEC TR 15504			
3.45	Post-implementation audits			

Competency Rating Totals (total each “✓” in each column):

Skill Category 4 – Quality Assurance

Quality Assurance is a professional competency whose focus is directed at the critical processes used to build products and services. The profession is charged with the responsibility for tactical process improvement initiatives that are strategically aligned to the goals of the organization. This category addresses the understanding and application of quality assurance practices in support of the strategic quality direction of the organization. The quality practitioner should understand the importance of a quality function, how to implement a quality function and how it matures over time, as well as how to create a quality plan, the use of quality tools, process deployment, and differentiating between internal auditing and quality assurance.

Skill Category 4 – Quality Assurance		Competency Rating		
Skill #	Skill Description	Full	Some	None
4.46	<i>Establishing a Function to Promote and Manage Quality</i> The challenges of implementing a quality function			
4.47	How the quality function matures over time			
4.48	Support in corporate quality management environment			
4.49	Implementing an IT quality function			
4.50	<i>Quality Tools</i> Management tools			
4.51	Statistical tools			
4.52	Presentation tools			
4.53	<i>Process Deployment</i> Getting buy-in for change through marketing			
4.54	The formula for effective behavior change			
4.55	The deployment process			
4.56	Critical success factors for deployment			
4.57	<i>Internal Auditing and Quality Assurance</i> Types of internal audits			
4.58	Differences in responsibilities			

Competency Rating Totals (total each “✓” in each column): _____

Skill Category 5 – Quality Planning

Executive management establishes the vision and strategic goals. Planning is the process that describes how those strategic goals will be accomplished. Quality planning should be integrated into the IT plan so that they become a single plan. In simplistic terms, the IT plan represents the producer and the quality plan represents the customer.

Skill #	Skill Category 5 – Quality Planning Skill Description	Competency Rating		
		Full	Some	None
5.59	<i>Planning Concepts</i> The management cycle			
5.60	The planning cycle			
5.61	<i>Integrating Business and Quality Planning</i> The fallacy of having two separate planning processes			
5.62	Planning should be a single IT activity			
5.63	<i>Prerequisites to Quality Planning</i>			
5.64	<i>The Planning Process</i> Planning process overview			
5.65	The six basic planning questions			
5.66	The common activities in the planning process			
5.67	<i>Planning to Mature IT Work Processes</i> QAI model and approach to mature IT work processes			
5.68	How to plan the sequence for implementing process maturity			

Competency Rating Totals (total each “✓” in each column):

Skill Category 6 – Define, Build, Implement and Improve Work Processes

The world is constantly changing. Customers are more knowledgeable and demanding, therefore, quality and speed of delivery are now critical needs. Companies must constantly improve their ability to produce quality products that add value to their customer base. Defining and continuously improving work processes allows the pace of change to be maintained without negatively impacting the quality of products and services. This category addresses process management concepts, including the definition of a process, the workbench concept and components of a process. Additionally, it will address the understanding of definitions and continuous improvement of a process through the process management PDCA cycle.

Skill Category 6 – Define, Build, Implement and Improve Work Processes		Competency Rating		
Skill #	Skill Description	Full	Some	None
6.69	<i>Process Management Concepts</i> Definition of a process			
6.70	Why processes are needed			
6.71	Process workbench and components			
6.72	Process categories			
6.73	The process maturity continuum			
6.74	How processes are managed			
6.75	Process template			
	<i>Process Management Processes</i>			
	<i>Planning processes:</i>			
6.76	Process inventory			
6.77	Process mapping			
6.78	Process planning			
	<i>Do processes:</i>			
6.79	Process definition			
	<i>Check processes:</i>			
6.80	Identify control points			
6.81	Process measurement			
6.82	Testing			
	<i>Act processes:</i>			
6.83	Process improvement teams			
6.84	Process improvement process			

Competency Rating Totals (total each "✓" in each column):

Skill Category 7 – Quality Control Practices

Quality control practices should occur during product development, product acquisition, product construction at the end of development/acquisition and throughout product change and operation. During development, the quality control process is frequently called verification and at the conclusion of development, it is called validation. This category will address the various types of controls and when they are best used in the process. The quality practitioner should also be familiar with verification and validation techniques, the framework for developing testing tactics, change control and configuration management.

Skill #	Skill Category 7 – Quality Control Practices Skill Description	Competency Rating		
		Full	Some	None
	<i>Testing Concepts</i>			
7.85	The tester's workbench			
7.86	Test stages			
7.87	Independent testing			
7.88	Static versus dynamic testing			
7.89	Verification versus validation			
7.90	Stress versus volume versus performance			
7.91	Test objectives			
7.92	Reviews and inspections			
	<i>Developing Testing Methodologies</i>			
7.93	Acquire and study the test strategy			
7.94	Determine the type of development project			
7.95	Determine the type of software system			
7.96	Determine the project scope			
7.97	Identify the tactical risks			
7.98	Determine when testing should occur			
7.99	Build the system test plan			
7.100	Build the unit test plans			
	<i>Verification and Validation Methods</i>			
7.101	Management of verification and validation			
7.102	Verification techniques			
7.103	Validation techniques			
7.104	Structural and functional testing			
	<i>Software Change Control</i>			
7.105	Software configuration management			
7.106	Change control procedures			
	<i>Defect Management</i>			
7.107	Defect management process			
7.108	Defect reporting			
7.109	Severity versus priority			
7.110	Using defects for process improvement			

Competency Rating Totals (total each "✓" in each column): _____

Skill Category 8 – Metrics and Measurement

A properly established measurement system is used to help achieve missions, visions, goals, and objectives. Measurement data is most reliable when it is generated as a by-product of producing a product or service. The QA analyst must ensure that quantitative data is valued and reliable, and presented to management in a timely and easy-to-use manner. Measurement can be used to gauge the status, effectiveness and efficiency of processes, customer satisfaction, product quality, and as a tool for management to use in their decision-making processes. This category addresses measurement concepts, the use of measurement in a software development environment, variation, process capability, risk management, the ways measurement can be used and how to implement an effective measurement program.

Skill #	Skill Category 8 – Metrics and Measurement Skill Description	Competency Rating		
		Full	Some	None
8.111	<i>Measurement Concepts</i> Standard units of measure			
8.112	Metrics			
8.113	Objective and subjective measurement			
8.114	Types of measurement data			
8.115	Measures of central tendency			
8.116	Attributes of good measurement			
8.117	Using quantitative data to manage an IT function			
8.118	Key indicators			
8.119	<i>Measurement in Software</i> Product measurement			
8.120	Process measurement			
8.121	<i>Variation and Process Capability</i> The measurement program			
8.122	Installing the measurement program			
8.123	Common and special causes of variation			
8.124	Variation and process improvement			
8.125	Process capability			
8.126	<i>Risk Management</i> Defining risk			
8.127	Characterizing risk			
8.128	Managing risk			
8.129	Software risk management			
8.130	Risks of integrating new technology			
8.131	<i>Implementing a Measurement Program</i> The need for measurement			
8.132	Prerequisites			

Competency Rating Totals (total each "✓" in each column):

Skill Category 9 – Internal Control and Security

Privacy laws and increased accessibility to data have necessitated increased security. Accounting scandals and governmental regulation such as the Sarbanes-Oxley Act have placed increased importance on building and maintaining adequate systems of internal control. The quality assurance function can contribute to meeting those objectives by assuring that IT has adequate processes governing internal control and security.

Skill #	Skill Category 9 – Internal Control and Security Skill Description	Competency Rating		
		Full	Some	None
9.133	<i>Principles and Concepts of Internal Control</i> Internal control and security vocabulary and concepts			
9.134	Preventive, detective, and corrective controls			
9.135	<i>Risk and Internal Control Models</i> COSO enterprise risk management (ERM) model			
9.136	COSO internal control framework model			
9.137	CobiT model			
9.138	<i>Building Internal Controls</i> Perform risk assessment			
9.139	<i>Building Adequate Security</i> Where vulnerabilities in security occur			
9.140	Establishing a security baseline			
9.141	Security awareness training			
9.142	Security practices			

Competency Rating Totals (total each “✓” in each column): _____

Skill Category 10 – Outsourcing, COTS, and Contracting Quality

Organizations can assign software development work responsibilities to outside organizations through purchasing software or contracting services; but they cannot assign the responsibility for quality. Quality of software remains an internal IT responsibility regardless of who builds the software. The quality professionals need to assure that those quality responsibilities are fulfilled through appropriate processes for acquiring purchased software and contracting for software services.

Skill Category 10 – Outsourcing, COTS, and Contracting Quality		Competency Rating		
Skill #	Skill Description	Full	Some	None
10.143	<i>Quality and Outside Software</i> Purchased COTS software			
10.144	Outsourced software			
10.145	<i>Selecting COTS Software</i> Assure completeness of needs requirements			
10.146	Define critical success factor			
10.147	Determine compatibility with hardware, operating system, and other COTS software			
10.148	Assure the Software can be Integrated into Your Business System Work Flow			
10.149	Demonstrate the Software in Operation			
10.150	Evaluate People Fit			
10.151	Acceptance Test the Software Process			
10.152	<i>Selecting Software Developed by Outside Organizations</i> Contracting life cycle			
10.153	Developing selection criteria			
10.154	<i>Contracting for Software Developed by Outside Organizations</i> Contract negotiations			
10.155	<i>Operating for Software Developed by Outside Organizations</i> Acceptance testing			

Competency Rating Totals (total the “√” in each column):

CSQA 2006 CBOOK Competency Rating Table

CBOOK Skill Category Competency Ratings			
	Full	Some	None
Skill Category 1			
Skill Category 2			
Skill Category 3			
Skill Category 4			
Skill Category 5			
Skill Category 6			
Skill Category 7			
Skill Category 8			
Skill Category 9			
Skill Category 10			
Ratings Total			
Factor for Multiplication	x 3	x 2	x 1
Columns Total			
Sum of the Rows Total			
Number of CSQA Skills			÷ 155
Your CSQA 2006 CBOOK Competency Rating			

Quality Principles

Before an organization can begin to assess the quality of its products and services, and identify opportunity for improvement, it first must have a working knowledge of quality principles and basic concepts. This category tests the CSQA candidate's ability to understand and apply these principles, which include the following:

<i>Vocabulary of Quality</i>	41
<i>The Different Views of Quality</i>	43
<i>Quality Concepts and Practices</i>	46
<i>Quality Control and Quality Assurance</i>	55
<i>Quality Pioneers Approach to Quality</i>	59

Vocabulary of Quality

The quality language is the way quality professionals describe the principles, concepts, and approaches used for improving quality. Until the vocabulary is learned and its use encouraged in the organization, quality becomes a difficult program to achieve. For example, when the words “process” or “defect” are used, there must be a common understanding of what is meant by those terms.

Appendix A provides a glossary of definitions for terminology used in the quality language. This terminology is also referred to as the vocabulary of quality. Some of the more widely used terms are:

Defect

From the producer's viewpoint, a defect is a product requirement that has not been met, or a product attribute possessed by a product or a function performed by a

product that is not in the statement of requirements that define the product. From the customer's viewpoint, a defect is anything that causes customer dissatisfaction, whether in the statement of requirements or not.

Policy

Managerial desires and intents concerning either processes (intended objectives) or products (desired attributes).

Procedure

The step-by-step method followed to ensure that standards are met.

Process

(1) The work effort that produces a product. This includes efforts of people and equipment guided by policies, standards, and procedures. (2) A statement of purpose and an essential set of practices (activities) that address that purpose. A process or set of processes used by an organization or project to plan, manage, execute, monitor, control, and improve its software related activities.

Productivity

The ratio of the output of a process to the input, usually measured in the same units. It is frequently useful to compare the value added to a product by a process, to the value of the input resources required (using fair market values for both input and output).

Quality

Operationally, the word quality refers to products. A product is a quality product if it is defect free. To the producer, a product is a quality product if it meets or conforms to the statement of requirements that defines the product. This statement is usually shortened to: quality means meets requirements. To the customer, a product is a quality product if it meets the customer's needs, regardless of whether the requirements were met. This is referred to as fit for use.

Quality – Producer View

The producer's view of quality has these four characteristics: Doing the right thing, Doing it the right way, Doing it right the first time, and Doing it on time without exceeding cost.

Quality – Customer View

Meeting requirements is a producer's view of quality. This is the view of the organization responsible for the project and processes, and the products and services acquired, developed, and maintained by those processes.

Standard

A requirement of a product or process. For example: 100 percent of the functionality must be tested.

The Different Views of Quality

Industry accepted definitions of quality are “conformance to requirements” (from Philip Crosby) and “fit for use” (from Dr. Joseph Juran and Dr. W. Edwards Deming). These two definitions are not inconsistent.

Meeting requirements is a producer’s view of quality. This is the view of the organization responsible for the project and processes, and the products and services acquired, developed, and maintained by those processes. Meeting requirements means that the person building the product does so in accordance with the requirements. Requirements can be very complete or they can be simple, but they must be defined in a measurable format, so it can be determined whether they have been met. The producer’s view of quality has these four characteristics:

- Doing the right thing
- Doing it the right way
- Doing it right the first time
- Doing it on time without exceeding cost

Being fit for use is the customer’s definition. The customer is the end user of the products or services. Fit for use means that the product or service meets the customer’s needs regardless of the product requirements. Of the two definitions of quality, fit for use, is the more important. The customer’s view of quality has these characteristics:

- Receiving the right product for their use
- Being satisfied that their needs have been met
- Meeting their expectations
- Being treated with integrity, courtesy and respect

In addition to the producer and customer views of quality, the organizational infrastructure also includes a provider and a supplier view. These views are as follows:

- Provider view – This is the perspective of the organization that delivers the products and services to the customer.
- Supplier view – This is the perspective of the organization (that may be external to the producer’s company, such as an independent vendor) that provides either the producer and/or the provider with products and services needed to meet the requirements of the customer.

The infrastructure for quality products and services is illustrated in Figure 2. The figure shows the requirements coming from the customer to the producer/provider, who uses them to create the

products and services needed by the customer. This process works because of the two-way measurement process established between the involved parties.

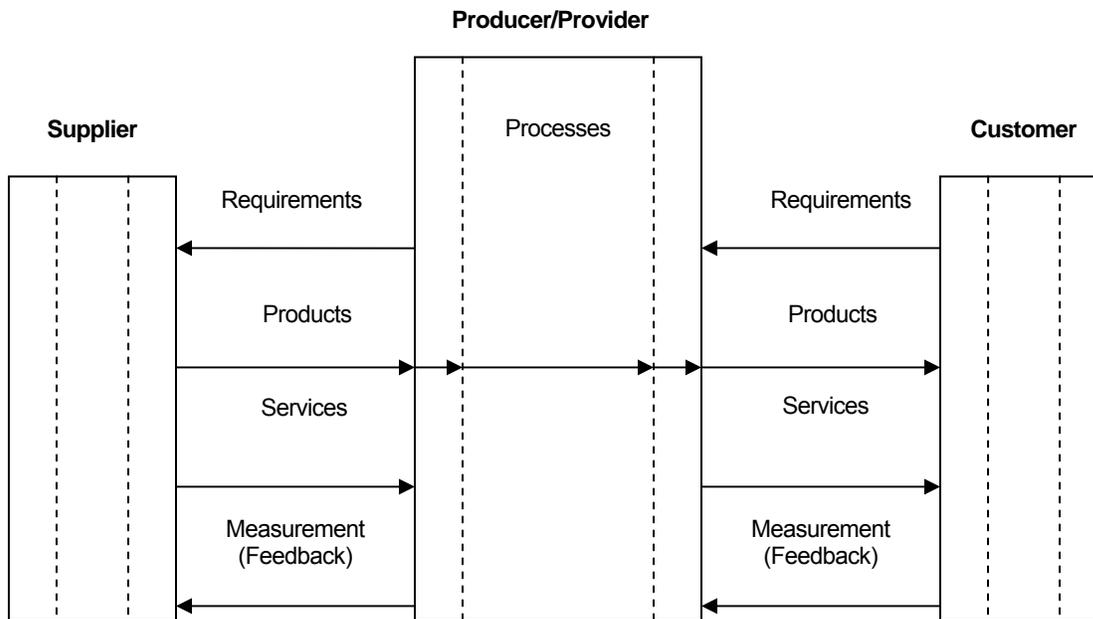


Figure 2. Infrastructure for Software Quality Products and Services

This infrastructure has been presented simplistically. In reality, the producer is the customer for the supplier, making the supplier the producer for the intermediate producer, and there may be a long chain of producers/providers and their customers. However, the quality characteristics by which an interim producer evaluates supplier products are really producer quality characteristics and not end user/customer quality characteristics.

The Two Quality Gaps

Most Information Technology (IT) groups have two quality gaps: the producer gap and the customer gap as shown in Figure 3. The producer gap is the difference between what is specified (the documented requirements and internal standards) versus what is delivered (what is actually built). The customer gap is the difference between what the producers actually delivered versus what the customer wanted.

Closing these two gaps is the responsibility of the quality function (see Skill Category 4). The quality function must first improve the processes to the point where the producer can develop the products according to requirements received and its own internal standards. Closing the producer's gap enables the IT function to provide its customers consistency in what it can produce. This has been referred to as the "McDonald's effect" - at any McDonald's in the world, a Big Mac should taste the same. It doesn't mean that every customer likes the Big Mac or that it meets everyone's needs, but rather, that McDonald's has now produced consistency in its delivered product.

Closing the second gap requires the quality function to understand the true needs of the customer. This can be done by customer surveys, Joint Application Development (JAD) sessions, and more user involvement through the process of building information products. The processes can then be changed to close the customer gap, keeping consistency while producing products and services needed by the customer.

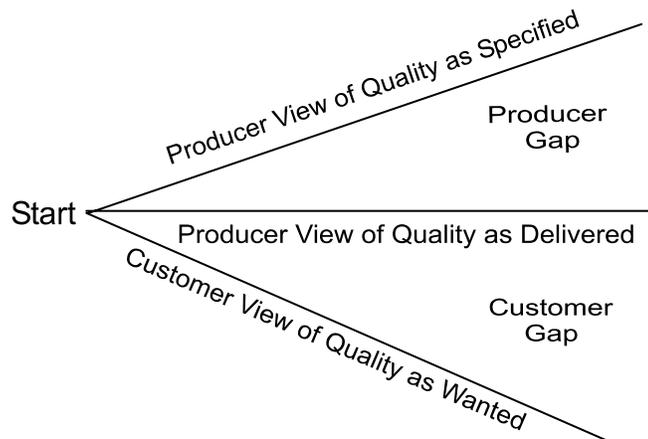


Figure 3. Two Quality Gaps

Quality Attributes for an Information System

Quality is a multifaceted concept driven by customer requirements. The level of quality can vary significantly from project to project and between organizations. In IT, the attributes of quality are examined in order to understand the components of quality, and as a basis for measuring quality. Some of the commonly accepted quality attributes for an information system are described in Figure 4.

Management needs to develop quantitative, measurable "standards" for each of these quality criteria for their development projects. For example, management must decide the degree of maintenance effort that is acceptable, the amount of time that it should take for a user to learn how to use the system, etc. Skill Category 8 covers this topic in more detail.

Attributes	Definition
Correctness	Extent to which a program satisfies its specifications and fulfills the user's mission objectives.
Reliability	Extent to which a program can be expected to perform its intended function with required precision.
Efficiency	The amount of computing resources and code required by a program to perform a function.
Integrity	Extent to which access to software or data by unauthorized persons can be controlled.
Usability	Effort required learning, operating, preparing input, and interpreting output of a program.
Maintainability	Effort required locating and fixing an error in an operational program.
Testability	Effort required testing a program to ensure that it performs its intended function.
Flexibility	Effort required modifying an operational program.
Reusability	Extent to which a program can be used in other applications – related to the packaging and scope of the functions that programs perform.
Interoperability	Effort required to couple one system with another.

Figure 4. Commonly Accepted Quality Attributes (Critical Success Factor) for Information Systems

"The Paul Revere Insurance Group believes that if a customer does not perceive quality, the program is not accomplished."

Charles E. Soule, Past Executive Vice President
Paul Revere Insurance Group

Quality Concepts and Practices

People say they want quality; however, their actions may not support this view for the following reasons:

- Many think that defect-free products and services are not practical or economical, and thus believe some level of defects is normal and acceptable. (This is called acceptable quality level, or AQL.) Quality experts agree that AQL is not a suitable definition of quality. As long as management is willing to "accept" defective products, the entire quality program will be in jeopardy.
- Quality is frequently associated with cost, meaning that high quality is synonymous with high cost. (This is confusion between quality of design and quality of conformance.) Organizations may be reluctant to spend on quality assurance, as they do not see an immediate payback.

- Quality by definition calls for requirements/specifications in enough detail so that the products produced can be quantitatively measured against those specifications. Few organizations are willing to expend the effort to produce requirements/specifications at the level of detail required for quantitative measurement.
- Many technical personnel believe that standards inhibit their creativity, and thus do not strive for compliance to standards. However, for quality to happen there must be well-defined standards and procedures that are followed.

The contributors to poor quality in many organizations can be categorized as either lack of involvement by management, or lack of knowledge about quality. Following are some of the specific contributors for these two categories:

Lack of involvement by management

- Management's unwillingness to accept full responsibility for all defects
- Failure to determine the cost associated with defects (i.e., poor quality)
- Failure to initiate a program to "manage defects"
- Lack of emphasis on processes and measurement
- Failure to enforce standards
- Failure to reward people for following processes

Lack of knowledge about quality

- Lack of a quality vocabulary, which makes it difficult to communicate quality problems and objectives
- Lack of knowledge of the principles of quality (i.e., what is necessary to make it happen)
- No categorization scheme for defects (i.e., naming of defects by type)
- No information on the occurrence of defects by type, by frequency, and by location
- Unknown defect expectation rates for new products
- Defect-prone processes unknown or unidentified
- Defect-prone products unknown or unidentified
- An economical means for identifying defects unknown
- Proven quality solutions are unknown and unused

If achieving quality (i.e., defect-free products and services) were easy, it would have been accomplished years ago. Quality is very difficult to accomplish – it requires the close cooperation of management and staff. Achieving quality requires a commitment and the establishment of an environment in which quality can flourish. Skill Category 2 focuses on management commitment and a quality management environment.

The bottom line is that making quality happen is a monumental challenge. Dr. Ishikawa, Japan's leading quality expert, best expressed this when he stated that accomplishing quality requires "a thought revolution by management." Thought revolutions do not come easy.

As a result of his experiences in turning around the Japanese economy, Dr. W. Edwards Deming found that it takes 20 years to change a culture from an emphasis on productivity to an emphasis on quality. Twenty years might be excessive, but management must be prepared to invest 2-5 years before the really large paybacks occur. Quality is a long-term strategy, which must be continually nurtured by the quality function and management.

The answer to the question, "Can we afford quality?" is: "You cannot afford to ignore it." Harold S. Geneen, past CEO at ITT, stated that quality "is the most profitable product line we have." What this means is that preventing and/or detecting defects early results in huge savings. Studies by Dr. Barry W. Boehm at GTE, TRW, and IBM in the late 1980s showed geometric escalation in the cost to fix a problem as the software life cycle progressed. Boehm concluded that errors are typically 100 times more expensive to correct in the maintenance phase on large projects than in the requirements phase. Boehm also stated that the total economic impact is actually much larger in operational systems because of the user costs incurred. Recent studies show that with today's more complex systems, Boehm's estimates are conservative.

PDCA Cycle

A major premise of a quality management environment is an emphasis on continuous improvement. The approach to continuous improvement is best illustrated using the PDCA cycle, which was developed in the 1930s by Dr. Shewhart of the Bell System. The cycle comprises the four steps of Plan, Do, Check, and Act as shown in Figure 5. It is also called the Deming Wheel, and is one of the key concepts of quality.

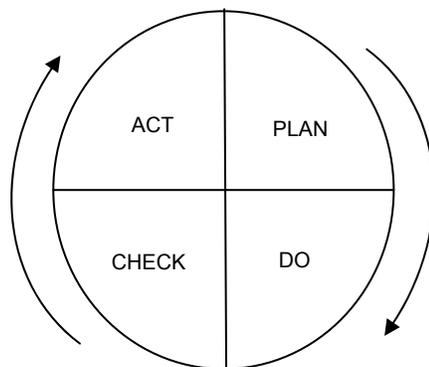


Figure 5. PDCA Concept

- Plan (P): Devise a plan - Define the objective, expressing it numerically, if possible. Clearly describe the goals and policies needed to attain the objective at

this stage. Determine the procedures and conditions for the means and methods that will be used to achieve the objective.

- Do (D): Execute the plan - Create the conditions and perform the necessary teaching and training to ensure everyone understands the objectives and the plan. Teach workers the procedures and skills they need to fulfill the plan and thoroughly understand the job. Then perform the work according to these procedures.
- Check (C): Check the results – As often as possible, check to determine whether work is progressing according to the plan and whether the expected results are obtained. Check for performance of the procedures, changes in conditions, or abnormalities that may appear.
- Act (A): Take the necessary action - If the check reveals that the work is not being performed according to plan, or if results are not what were anticipated, devise measures for appropriate action. Look for the cause of the abnormality to prevent its recurrence. Sometimes workers may need to be retrained and procedures revised. The next plan should reflect these changes and define them in more detail.

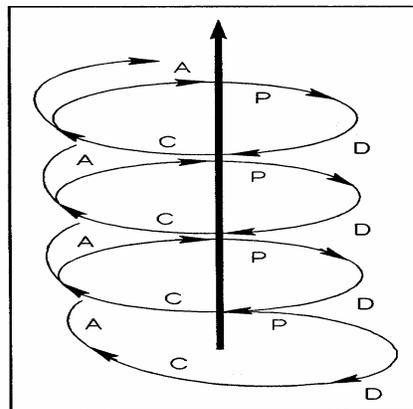


Figure 6. Ascending Spiral

The PDCA procedures ensure that the quality of the products and services meets expectations, and that the anticipated budget and delivery date are fulfilled. Sometimes preoccupation with current concerns limits the ability to achieve optimal results. Repeatedly going around the PDCA circle can improve the quality of the work and work methods, and obtain the desired results. This concept can be seen in the ascending spiral of Figure 6.

Cost of Quality

Quality is an attribute of a product or service. Productivity is an attribute of a process. They have frequently been called two sides of the same coin because one significantly impacts the other. There are two ways that quality can drive productivity. The first, which is an undesirable method, is to lower or not meet quality standards. For example, if testing and rework components of a system development process were eliminated or reduced, productivity as measured in lines of code per hours worked would increase. This is often done under the guise of completing projects

on time. The second and more desirable method to improve productivity through quality is to improve processes so that defects do not occur, thus minimizing the need for testing and rework. Quality improvement should be used to drive productivity.

The *cost of quality* (COQ) is the money spent beyond what it would cost to build a product right the first time. If every worker could produce defect-free products the first time, the COQ would be zero. Since this situation does not occur, there are costs associated with getting a defect-free product produced.

There are three COQ categories:

- Prevention - Money required preventing errors and to do the job right the first time is considered prevention cost. This category includes money spent on establishing methods and procedures, training workers and planning for quality. Prevention money is all spent before the product is actually built.
- Appraisal – Appraisal costs cover money spent to review completed products against requirements. Appraisal includes the cost of inspections, testing and reviews. This money is spent after the product or subcomponents are built but before it is shipped to the user.
- Failure – Failure costs are all costs associated with defective products. Some failure costs involve repairing products to make them meet requirements. Others are costs generated by failures, such as the cost of operating faulty products, damage incurred by using them and the costs incurred because the product is not available. The user or customer of the organization may also experience failure costs.

Figure 7 shows a few examples of the three costs of quality that illustrate the types of activities in each of the categories. Experience has shown that a small group of knowledgeable people can develop estimates for the COQ categories. The estimate does not have to be highly precise because the amounts will be so large that even errors of plus or minus 50% would not affect identifying the actions that need to be taken to reduce the COQ.

Prevention Costs	
<p style="text-align: center;">In IT Area</p> <ul style="list-style-type: none"> • Quality audits • Selling top management • Planning quality improvement • Quality training • Systems assurance consultation 	<p style="text-align: center;">In User Area</p> <ul style="list-style-type: none"> • Installing a project selection process • Installing a planning database • Installing improved programming techniques
Appraisal Costs	
<p style="text-align: center;">In IT Area</p> <ul style="list-style-type: none"> • Preparation for reviews • Inspections • Systems assurance reviews 	<p style="text-align: center;">In User Area</p> <ul style="list-style-type: none"> • Phase reviews • Preparation for tests • Systems test
Failure Costs	
<p style="text-align: center;">In IT Area</p> <ul style="list-style-type: none"> • Project rework • Overtime • Maintenance costs • Lost IT credibility • Reruns 	<p style="text-align: center;">In User Area</p> <ul style="list-style-type: none"> • Alternative services • Lost management time • Complaints, rebates, damage claims • Lost assets • Lost opportunity • Unrealized savings

Figure 7. Cost of Quality Sample

The cost of building a product is comprised of the cost of production, which is the cost if the product could be built defect free, plus the three COQ categories. Added together, the production cost and COQ become the cost to build a product. The three COQ categories are sometimes called the cost of nonconformance, meaning COQ is the failure to conform to a process that enables defect-free products to be produced.

The quality function attempts to reduce the cost of quality. This is usually accomplished by increasing the prevention and/or the appraisal costs in order to reduce the failure costs more than the increase in the prevention and appraisal costs. Figure 8 illustrates this phenomenon. It shows that initiating new appraisal programs such as inspections and reviews in software development, or new preventive programs such as staff training, can reduce the failure costs, which include such things as rework, so there is a net reduction in the cost to build a product.

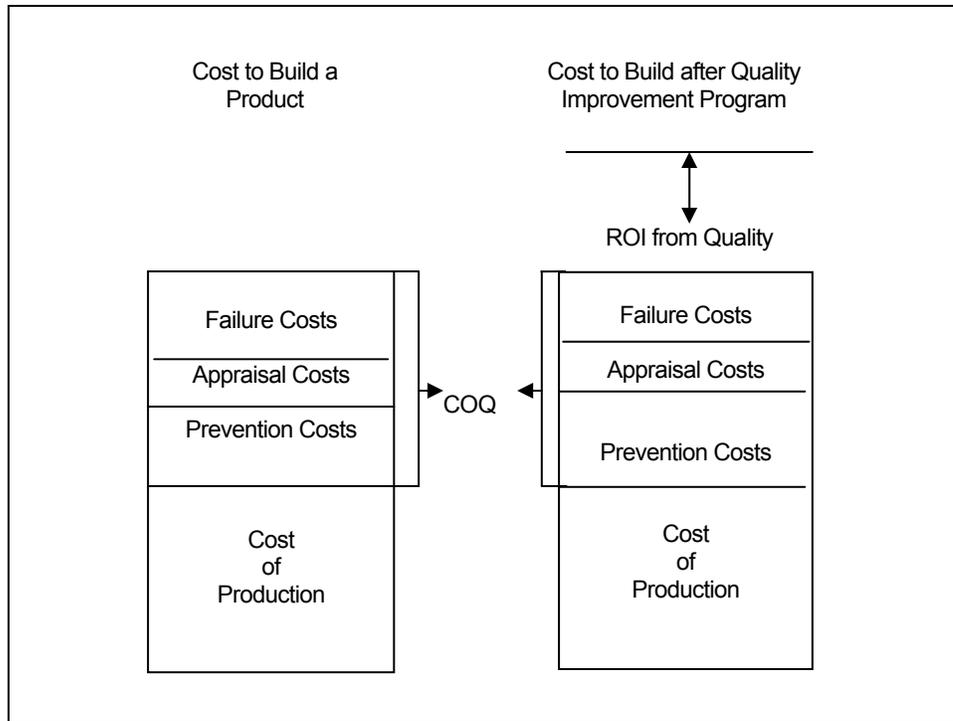


Figure 8. Examples of Cost of Quality

Studies show that the COQ in IT is approximately 50% of the total cost of building a product. Of the 50% COQ, 40% is failure, 7% is appraisal, and 3% is prevention. Other studies have shown that \$1 spent on appraisal costs will reduce failure costs threefold; and each dollar spent on prevention costs will reduce failure costs tenfold. Obviously, the right appraisal and prevention methods must be used to get these benefits.

For example, the cost of adding unidentified requirements during system or acceptance testing is much more costly than identifying those requirements during the requirements-gathering phase. Once individuals understand the cost of "overlooking requirements" they might be more willing to use techniques such as requirements reviews, JAD sessions, and improved processes to avoid the overtime, stress, and excessive cost of building those overlooked requirements late in the development process.

"Quality is free, but it is not a gift."

Philip B. Crosby in *Quality is Free*

The Three Key Principles of Quality

Everyone is responsible for quality, but senior management must emphasize and initiate quality improvement, and then move it down through the organization to the individual employees. The following three quality principles must be in place for quality to happen:

1. Management is responsible for quality.

Quality cannot be delegated effectively. Management must accept the responsibility for the quality of the products produced in their organization; otherwise, quality will not happen. A quality function is only a catalyst in making quality happen. The quality function assists management in building quality information systems by monitoring quality and making recommendations to management about areas where quality can be improved. As the quality function is a staff function, not management, it cannot dictate quality for the organization. Only management can make quality happen.

2. Producers must use effective quality control.

All of the parties and activities involved in producing a product must be involved in controlling the quality of those products. This means that the workers will be actively involved in the establishment of their own standards and procedures.

3. Quality is a journey, not a destination.

The objective of the quality program must be continuous improvement. The end objective of the quality process must be satisfied customers.

The Quality Solution

The action that must be taken by management to make quality happen is as simple as 1-2-3:

1. Define quality
2. Control quality
3. Assure quality

After management becomes committed to the quality principles, the most effective method for making these three actions happen is to establish a quality function. While a quality function is not necessary to make quality happen, quality rarely happens without adequate attention devoted to the quality objectives. As stated in the first key principle above, the quality function should be that catalytic group which initiates quality improvement programs in order to make quality happen. Management commitment is covered in Skill Category 2 and Skill Category 4 discusses the quality function.

Best Practices

A practice is a specific implementation of a work process. For example, a practice would be one organization's process for estimating the amount of resources required for building a system.

A Best Practice is one of the most effective practices for performing a specific process. Best Practices are normally identified by benchmarking, or by an independent assessment. Best Practices are also identified through winners of quality competitions such as the Malcolm Baldrige National Quality Award, Deming Prize, etc.

Six Sigma Quality

Most people spend twelve or more years in an educational system in which grades of 90% or higher, are considered excellent. However, in industry, 90% is not a good quality record. For example, if one out of every ten tires fails, you have a 90% quality rating, but that is totally unacceptable to tire customers.

Motorola developed a concept called “Six Sigma Quality” that focuses on defect rates, as opposed to percent performed correctly. “Sigma” is a statistical term meaning one standard deviation. “Six Sigma” means six standard deviations. At the Six Sigma statistical level, only 3.4 items per million are outside of the acceptable level. Thus, the Six Sigma quality level means that out of every one million items counted 999,996.6 will be correct, and no more than 3.4 will be defective.

Experience has shown that in most systems, a Four Sigma quality level is the norm. At the Four Sigma level there are 6,120 defects per million parts, or about 6 defects per 1,000 opportunities, to do a task correctly.

The key focus of companies implementing a Six Sigma program is to develop a good business strategy that balances the cost, quality, features and availability considerations for products. A valuable lesson learned is that decisions made must be tied to the bottom line for the company. Companies should take care to use correct measurements for each situation, and to consider measuring output of a process over time (not just a snapshot).

When considering a project to improve using Six Sigma, the following characteristics are desirable. If one or more of these characteristics is missing, there will likely be barriers to success.

- The project should clearly connect to business priorities.
- The problem being solved should be very important, such as a 50% process improvement.
- The importance of the project should be clear to the organization.
- The project should have a limited scope that can be completed in less than six months.
- The project should have clear, quantitative measures that define success.
- Management should support and approve the project to ensure resources are available, barriers are removed, and that the project continues over time.

Baselining and Benchmarking

Baselining is defining the current level of performance. For example, the number of defects currently contained in a thousand lines of code is usually calculated quantitatively and can be used to measure improvement.

Benchmarking involves a comparison of one organization’s, or one part of an organization’s, process for performing a work task to another organization’s process, for the purpose of finding

best practices or competitive practices that will help define superior performance of a product, service or support process.

Skill Category 3 provides additional details on benchmarking, including types of benchmarking and a four-step process for conducting benchmarking.

Earned Value

It is important that quality professionals be able to demonstrate that their work provides value to their organization. Return-on-investment (ROI) that demonstrates the dollars returned for the dollars invested is one of the more popular means to demonstrate value returned. However, ROI is not designed to measure subjective values such as customer loyalty. There is no generally accepted best way to measure the “value earned” from quality initiatives. It is recommended that quality professionals use the method(s) recommended by their accounting function for calculating earned value.

Quality Control and Quality Assurance

Very few individuals can differentiate between quality control and quality assurance. Most quality assurance groups, in fact, practice quality control. This section differentiates between the two, and describes how to recognize a control practice from an assurance practice.

Quality means meeting requirements and meeting customer needs, which means a defect-free product from both the producer’s and the customer’s viewpoint. Both quality control and quality assurance are used to make quality happen. Of the two, quality assurance is the more important.

Quality is an attribute of a product. A product is something produced, such as a requirement document, test data, source code, load module or terminal screen. Another type of product is a service that is performed, such as meetings with customers, help desk activities and training sessions. Services are a form of products, and therefore, also contain attributes. For example, an agenda might be a quality attribute of a meeting.

A process is the set of activities that is performed to produce a product. Quality is achieved through processes. Processes have the advantage of being able to replicate a product time and time again. Even in data processing, the process is able to replicate similar products with the same quality characteristics.

Quality Assurance (QA) is associated with a process. Once processes are consistent, they can “assure” that the same level of quality will be incorporated into each product produced by that process.

Quality Control

Quality Control (QC) is defined as the processes and methods used to compare product quality to requirements and applicable standards, and the action taken when a nonconformance is detected. QC uses reviews and testing to focus on the detection and correction of defects before shipment of products.

Quality Control should be the responsibility of the organizational unit producing the product and should be integrated into the work activities. Ideally the same group that builds the product performs the control function; however, some organizations establish a separate group or department to check the product.

Impediments to QC include the following:

- Quality Control is often viewed as a police action
- IT is often considered an art
- Unclear or ineffective standards and processes
- Lack of process training

Quality Control is the focus of Skill Category 7.

Understanding and Using the Just-In-Time (JIT) Technique

Just-in-time (JIT) is a revolutionary production system developed by Taiichi Ohno, a Toyota vice-president. He examined and challenged a known manufacturing principle, and developed a disciplined system that placed Toyota a quantum step ahead of their rivals in the Western countries. This system, now known as, “the Toyota production system,” has set the standard for world-class manufacturing. The concept may be adopted within IT.

The ultimate goal of JIT production is to supply each process with exactly the required items, in exactly the required quantity, at exactly the required time. There are two conditions necessary to reach this situation: large amounts of production flexibility, and very short lead times.

The basic difference between the old method of supply and the new system is that the concept of a one-process department is eliminated. The same work tasks are no longer all performed in the same work area. These highly specialized departments are replaced with mixed lines of processing capabilities laid out in the sequence required to make the part or groups of parts. Parts having similar size, shape, material, and processing sequence are allocated to those lines by a system known as “group technology.” Parts are processed over these lines one at a time in very small batches.

The key to JIT production is the ability of the production area to quickly switch from job to job within a few moments. The manufacturing technique for doing this is known as SMED (single-minute exchange of dies). This technique of quickly changing work capabilities can significantly reduce the cost of doing work.

Another change from the traditional processing method is that a JIT shop uses a system called “kanban,” which is a Japanese word for “visible record.” Kanban cards are like money in a cash-and-carry store. The goods produced in one area can be viewed as for sale to other areas in the process. Instead of producing work in one area and pushing them or giving them to the next operation, the goods stay with the producing department until the next step in the process comes to the preceding operation, and takes only what is needed. The traditional method of a “push”

system, in which the work is pushed through the operation from beginning to end, is changed to a “pull” system, in which data is only moved forward when it is needed by the next operation.

Toyota considers inventory to be at the root of all evil in a manufacturing plant. Inventory is used as a protection, or buffer, for known trouble levels and schedule changes. It covers up systems inadequacies and costs associated with carrying inventory that are not always apparent.

An important part of JIT is a high level of quality. Traditional plants define quality control as only controlling the manufacturing process to ensure the product meets its specifications. Toyota’s view of total quality control is much broader. All departments, not just the manufacturing departments, focus their efforts on contributing to customer satisfaction, which is the ultimate measure of success. They believe that only a customer who is 100 percent satisfied with a product will return to buy another and advise his or her friends to do the same.

Of the three critical aspects of Toyota quality (war on waste, perfect quality, and employee involvement), the human element, or employee involvement, is the most important. Management in the work force forms a partnership where each party commits to the mutual success. While the system is a revolutionary way to approach manufacturing, and is not just an inventory control system, the implementation of JIT is driven by the goal of inventory reduction.

Just-in-time principles can be used in IT in the following ways:

- Systems development and maintenance tasks become driven when the user of an internal or external product or service needs them. Programs would not be developed before they are needed for test or production.
- Systems analysts and programmers would not be given information and documents to store until they need them.
- Internal information processes would be designed so individuals can move from job to job with minimal delay. For example, programmers should be able to stop working on one program and start another within the JIT ten-minute turnover standard.

Quality Assurance

Quality Assurance (QA) is the set of activities (including facilitation, training, measurement and analysis) needed to provide adequate confidence that processes are established and continuously improved in order to produce products or services that conform to requirements and are fit for use.

QA is a staff function that prevents problems by heading them off, and by advising restraint and redirection at the proper time. It is also a catalytic function that should promote quality concepts, and encourage quality attitudes and discipline on the part of management and workers. Successful QA managers know how to make people quality conscious and to make them recognize the personal and organizational benefits of quality.

The major impediments to QA come from management, which is typically results oriented, and sees little need for a function that emphasizes managing and controlling processes. Thus, many of the impediments to QA are associated with processes, and include the following:

- Management does not insist on compliance to processes
- Workers are not convinced of the value of processes
- Processes become obsolete
- Processes are difficult to use
- Workers lack training in processes
- Processes are not measurable
- Measurement can threaten employees
- Processes do not focus on critical aspects of products

Differentiating Between Quality Control and Quality Assurance

QC is an activity that verifies whether or not the product produced meets standards. QA is an activity that establishes and evaluates the processes that produce the products. If there is no process, there is no role for QA. Assurance would determine the need for, and acquire or help install system development methodologies, estimation processes, system maintenance processes, and so forth. Once installed, QA would measure them to find weaknesses in the process and then correct those weaknesses to continually improve the processes.

It is possible to have quality control without quality assurance. For example, there might be a standard that “ALTER GO TO” statements in COBOL should not be used. Regardless of whether a program is produced using a system development process or done by an individual without a process, it could still be checked to determine whether or not “ALTER GO TOs” are in the program.

The following statements help differentiate QC from QA:

- QC relates to a specific product or service.
- QC verifies whether particular attributes exist, or do not exist, in a specific product or service.
- QC identifies defects for the primary purpose of correcting defects.
- QC is the responsibility of the worker.
- QA helps establish processes.
- QA sets up measurement programs to evaluate processes.
- QA identifies weaknesses in processes and improves them.
- QA is a management responsibility, frequently performed by a staff function.
- QA evaluates whether or not quality control is working for the primary purpose of determining whether or not there is a weakness in the process.
- QA is concerned with all of the products that will ever be produced by a process.

- QA is sometimes called quality control over quality control because it evaluates whether quality control is working.
- QA personnel should not ever perform quality control unless doing it to validate quality control is working.

Quality Pioneers Approach to Quality

Many individuals have contributed to the quality movement. Three individuals are highlighted here because they have either organized a business to promote their concepts, or have a following. These three are Dr. W. Edwards Deming, Philip Crosby and Dr. Joseph Juran.

Both Dr. Deming and Mr. Crosby have developed a set of quality principles. Dr. Juran is well known for his trilogy and distinction between “little-Q” quality and “big-Q” quality.

Dr. W. Edwards Deming

Dr. Deming defined 14 principles for quality, which formed the basis for the turnaround of the Japanese manufacturing industry. He believed that all 14 principles must be used concurrently to make quality happen. The 14 principles are discussed briefly below. Additional information can be found in his book *Out of the Crisis* (see the References in Appendix B).

1. Create Consistency of Purpose in the Company

In quality-oriented companies, quality should be the cornerstone of the corporation. All units within the organization should work toward common goals and purposes. Within IT this can be translated to mean that the training department teaches the standards, operations is working for the same goals as systems programming, and systems programming is dedicated to improving application performance. Assuring quality involves:

- Innovating new approaches and allocating resources for long-term planning, such as possible new service, new skills required, training and retraining of personnel, satisfaction of the user.
- Putting resources into research, education and maintenance.

2. Learn the New Philosophy

We are in a new economic age. We can no longer live with commonly accepted levels of mistakes, defects, material not suited to the job, people on the job that do not know what the job is and are afraid to ask, failure of management to understand the problems of the product in use; antiquated methods of training on the job; and inadequate and ineffective supervision. Acceptance of defective systems and poor workmanship as a way of life is one of the most effective roadblocks to better quality and productivity.

3. Require Statistical Evidence of Information Technology Quality

There is no other way to know the quality that is being delivered and no other way to achieve best economy and productivity. Managers must learn the statistical control of quality. They must proceed under the new philosophy; the right quality characteristics must be built in, without dependence on inspection.

4. Reduce the Number of Vendors

Requiring statistical evidence of quality control in the purchase of hardware and software will mean, in most companies, drastic reduction in the number of vendors with whom they deal. Companies will have to consider the cost of having two or more vendors for the same item. A company will be lucky to find one vendor that can supply statistical evidence of quality. A second vendor, if they cannot furnish statistical evidence of their quality, will have higher costs than the one that can furnish the evidence, or they will have to chisel on their quality, or go out of business. A person that does not know his/her costs or whether today's distribution of quality can be repeated tomorrow is not a good business partner.

5. Use Statistical Methods to Find Sources of Trouble

Which are user faults? Which faults belong to IT? Put responsibility where it belongs. Do not rely on judgment because it always gives the wrong answer on the question of where the fault lies. Statistical methods make use of knowledge of the subject matter where it can be effective, but supplant it where it is a hazard.

Constantly improve the system. This obligation never ceases. Most people in management do not understand that the system (their responsibility) is everything not under the governance of a user.

6. Institute Modern Aids to Training on the Job

Training must be totally reconstructed. Statistical methods must be used to learn when training is finished, and when further training would be beneficial. A person once hired and trained and in statistical control of his/her own work, whether it be satisfactory or not, can do no better. Further training cannot help. If their work is not satisfactory, move them to another job, and provide better training there.

7. Improve Supervision

Supervision belongs to the system, and is the responsibility of management.

- Project leaders must have more time to help people on the job.
- Statistical methods are vital aids to the project leader to indicate whether the fault lies locally or in the system.
- The usual procedure, by which a project leader calls the worker's attention to every defect or to half of them, may be wrong – is certainly wrong in most organizations – and defeats the purpose of supervision.

8. Drive Out Fear

Most people on a job, and even in management positions, do not understand what the job is, or what is right versus wrong. Moreover, it is not clear to them how to find out. Many of them are afraid to ask questions or to report trouble. The economic loss from fear is appalling. It is necessary, for better quality and productivity, that people feel secure.

Another related aspect of fear is the inability to serve the best interest of the company through necessity to satisfy specified rules, or to satisfy a production quota, or to cut costs by some specified amount.

One common result of fear is seen in inspection. An inspector records incorrectly the result of an inspection for fear of overdrawing the quota of allowable defectives of the work force.

9. Break Down Barriers Between Departments

People in user areas must learn about the problems encountered with various technologies and specifications in system design and operation. Otherwise, there will be losses in production from necessity of reruns and from attempts to use systems unsuited to the purpose. Why not have users spend time in the IT department to see the problems and hear about them?

10. Eliminate Numerical Goals, Slogans, Pictures, Posters Urging People to Increase Productivity, Sign Their Work as an Autograph, etc.

"Zero Defects" is an example. Posters and slogans like these never helped anyone do a better job. Numerical goals often have a negative effect through frustration. These devices are management's lazy way out. They indicate desperation and incompetence of management. There is a better way.

11. Look Carefully at Work Standards

Do they take account of quality, or only numbers? Do they help anyone do a better job? How old are they? Work standards are costing as much loss as poor materials and mistakes. In hundreds of companies any day, people stand around the last hour or two waiting for the whistle to blow. They have completed their quotas for the day. Is this good for the competitive position of any industry? Ask these people. They are unhappy doing nothing, and would rather work.

12. Institute a Massive Training Program for Employees in Simple but Powerful Statistical Methods

Thousands of people must learn rudimentary statistical methods. One in 500 must spend the necessary ten years to become a statistician. This training will be a costly affair.

13. Institute a Vigorous Program for Retraining People in New Skills

The program should keep up with changes in technology and methods and, if advantageous, new hardware.

14. Create a Structure in Top Management that Will Push Every Day on the Above 13 Points

Make maximum use of statistical knowledge and talent in your company. Top management will require guidance from an experienced consultant, but the consultant cannot take on obligations that only the management can carry out.

Philip Crosby

Philip Crosby has developed 14 steps for an organization to follow in building an effective quality program. These are:

1. Management Commitment

Clarify where management stands on quality. It is necessary to consistently produce conforming products and services at the optimum price. The device to accomplish this is the use of defect prevention techniques in the operating departments: engineering, manufacturing, quality control, purchasing, sales, and others. Management must ensure that no one is exempt.

2. The Quality Improvement Team

They run the quality improvement program. Since every function of an operation contributes to defect levels, every function must participate in the quality improvement effort. The degree of participation is best determined by the particular situation that exists. However, everyone has the opportunity to improve.

3. Quality Measurement

Communicate current and potential nonconformance problems in a manner that permits objective evaluation and corrective action. Basic quality measurement data is obtained from the inspection and test reports, which are broken down by operating areas of the plant. By comparing the rejection data with the input data, it is possible to know the rejection rates. Since most companies have such systems, it is not necessary to go into them in detail. It should be mentioned that unless this data is reported properly, it is useless. After all, their only purpose is to warn management of serious situations. They should be used to identify specific problems needing corrective action, and the quality department should report them.

4. The Cost of Quality

Define the ingredients of the COQ and explain its use as a management tool. See “Quality Concepts” on page 46 where COQ was defined and examples provided.

5. Quality Awareness

Provide a method of raising the personal concern felt by all personnel in the company toward the conformance of the product or service and the quality reputation of the company. By the time a company is ready for the quality awareness step, they should have a good idea of the types and expense of the problems being faced. The quality measurement and COQ steps will have revealed them.

6. Corrective Action

Provide a systematic method of permanently resolving the problems that are identified through previous action steps. Problems that are identified during the acceptance operation or by some other means must be documented and then resolved formally.

7. Zero Defects Planning

Examine the various activities that must be conducted in preparation for formally launching the Zero Defects (ZD) program - The quality improvement task team should list all the individual action steps that build up to ZD day in order to make the most meaningful presentation of the concept and action plan to personnel of the company. These steps, placed on a schedule and assigned to members of the team for execution, will provide a clean energy flow into an organization-wide ZD commitment. Since it is a natural step, it is not difficult, but because of the significance of it, management must make sure it is conducted properly.

8. Supervisor Training

Define the type of training supervisors need in order to actively carry out their part of the quality improvement program. The supervisor, from the board chairman down, is the key to achieving improvement goals. The supervisor gives the individual employees their attitudes and work standards, whether in engineering, sales, computer programming, or wherever. Therefore, the supervisor must be given primary consideration when laying out the program. The departmental representatives on the task team will be able to communicate much of the planning and concepts to the supervisors, but individual classes are essential to make sure that they properly understand and can implement the program.

9. ZD Day

Create an event that will let all employees realize through personal experience, that there has been a change. Zero Defects is a revelation to all involved that they are embarking on a new way of corporate life. Working under this discipline requires personal commitments and understanding. Therefore, it is necessary that all members of the company participate in an experience that will make them aware of this change.

10. Goal Setting

Turn pledges and commitments into action by encouraging individuals to establish improvement goals for themselves and their groups. About a week after ZD day,

individual supervisors should ask their people what kind of goals they should set for themselves. Try to get two goals from each area. These goals should be specific and measurable.

11. Error-Cause Removal

Give the individual employee a method of communicating to management the situations that make it difficult for the employee to fulfill the pledge to improve. One of the most difficult problems employees face is their inability to communicate problems to management. Sometimes they just put up with problems because they do not consider them important enough to bother the supervisor. Sometimes supervisors don't listen anyway. Suggestion programs are some help, but in a suggestion program the worker is required to know the problem and also propose a solution. Error-cause removal (ECR) is set up on the basis that the worker need only recognize the problem. When the worker has stated the problem, the proper department in the plant can look into it. Studies of ECR programs show that over 90% of the items submitted are acted upon, and fully 75% can be handled at the first level of supervision. The number of ECRs that save money is extremely high, since the worker generates savings every time the job is done better or quicker.

12. Recognition

Appreciate those who participate. People really don't work for money. They go to work for it, but once the salary has been established, their concern is appreciation. Recognize their contribution publicly and noisily, but don't demean them by applying a price tag to everything.

13. Quality Councils

Bring together the professional quality people for planned communication on a regular basis. It is vital for the professional quality people of an organization to meet regularly just to share their problems, feelings, and experiences, with each other. Primarily concerned with measurement and reporting, isolated even in the midst of many fellow workers, it is easy for them to become influenced by the urgency of activity in their work areas. Consistency of attitude and purpose is the essential personal characteristic of one who evaluates another's work. This is not only because of the importance of the work itself but because those who submit work unconsciously draw a great deal of their performance standard from the professional evaluator.

14. Do it Over Again

Emphasize that the quality improvement program never ends. There is always a great sign of relief when goals are reached. If care is not taken, the entire program will end at that moment. It is necessary to construct a new quality improvement team, and to let them begin again and create their own communications.

Dr. Joseph Juran

Dr. Juran believed that managing for quality required the same attention that other functions typically receive. To ensure that adequate attention was given, he developed a trilogy consisting of three interrelated, basic managerial phases/processes: quality planning, quality control and quality improvement. These are known as “The Juran Trilogy” or “The Quality Trilogy.”

- **Quality Planning**

The purpose of this phase is to create a process that enables goals to be met. In developing the process, quality planning should identify customers and their needs, and then incorporate those needs into the product and process designs. The planning process should also attempt to avoid costly deficiencies, such as rework, and optimize the company performance. This phase occurs before the process is used to produce a product.

- **Quality Control**

Quality control takes place at all levels in the organization, with everyone using the same feedback loop. Dr. Juran believed that in order to achieve control, processes must have numerical measures and adjustment capabilities. When products are produced from the process, there will always be some acceptable (inherent) variation; and the occasional spikes (those representing special causes of variation) should be investigated. Management should strive to give process users the capability of making the necessary adjustments to control the process. He refers to this as “self-control”. When the process is being designed, control should be part of the planning process. Typically quality control will be performed to prevent defects from worsening; it will not focus on the process.

- **Quality Improvement**

At some point in the quality control phase, the continuous loop of product deficiencies will be traced back to the planning process, and the problems will be seen as an opportunity to improve. Improvements will be made to revise the process, and problems will become less than originally planned.

Dr. Juran believed that business processes presented a major opportunity for improvement. He developed a structured approach for improvement, which included the following list of responsibilities for senior managers that could not be delegated:

- Create an awareness of the need and an opportunity for improvement
- Mandate quality improvement; make it part of every job description
- Create the infrastructure: establish a quality council, select projects to improve, appoint teams and provide facilitators
- Provide training in how to improve quality
- Review progress regularly
- Recognize winning teams
- Propagandize the results

In addition to his Trilogy, Dr. Juran is also known for his distinction between “little-Q” quality and “big-Q” quality. Prior to TQM there was only “little-Q”, which he called the narrow focus on quality. “Little-Q” quality is considered important, but it has a limited scope and impact, such as a team of people and their manager improving a specific work process. Dr. Juran referred to “big-Q” quality as the new focus on quality. An example of “big-Q” quality is cross-functional teams throughout an organization working to prevent problems. While the scope of “little-Q” quality is a specific departmental mission, the scope of “big-Q” quality emphasizes the coordination of various activities conducted in other functions and groups so that all plans contribute to achieving the organization’s goals for quality.

Total Quality Management

A Managerial Philosophy Based on the Work of the Pioneers

Total quality management (TQM) is the term used by many to indicate an organization-wide effort of continuous process improvement. The Federal Quality Institute defines TQM as a strategic, integrated management system for achieving customer satisfaction, which involves all managers and employees and uses quantitative methods to continuously improve an organization's processes.

Some additional thoughts on TQM include:

- The improved performance is directed toward satisfying such cross-functional goals as quality, cost, schedule, mission, need and suitability.
- TQM is a process of controlled change.
- Central to the TQM approach is the change in management philosophy regarding the "responsibility for quality." Formerly it rested with a separate group of individuals in a department/directorate/division often designated as quality assurance. Under TQM, the responsibility rests with every employee, beginning with top management. Skill Category 2 provides additional insight into the change in management philosophy, new behaviors for management and leadership.
- TQM is accomplished using a team organization; both management and the employees are members of "quality teams" (also called process improvement or process action teams), that focus on continuous process improvement (see Skill Category 2 regarding teams in general, and Skill Category 6 regarding process improvement teams). Suggestions to improve the quality of a particular process should come from the employees and the managers who work in the process, as they know it best.
- Communication must be encouraged to allow employees and management to work together to reach the mutual goal of continuous process improvement. Skill Category 2 contains several topics related to communication.
- The timing, sequence, method of implementation, and integration of these elements will vary from one organization to another.

- Professional literature addresses concepts by different names. Thus, terms like TQM or quality management, which are popular one year, may fall out of favor the next year. Readers should associate these activities with the term that is used within your organization.

Quality Leadership

The most important prerequisite for successful implementation of any major quality initiative is commitment from executive management. It is management's responsibility to establish strategic objectives and build an infrastructure that is strategically aligned to those objectives. This category describes the management processes used to establish the foundation of a quality-managed environment:

<i>Leadership Concepts</i>	69
<i>Quality Management Infrastructure</i>	81
<i>Quality Environment</i>	95

Leadership Concepts

Quality management is a philosophy and a set of guiding principles that represent the foundation of a continuously improving organization. Quality management is the application of quantitative methods and human resources to improve the products and services supplied to an organization, all processes within an organization, and the degree to which the current and future needs of the customer are met. Quality management integrates fundamental management techniques, existing improvement efforts, and technical tools under a disciplined approach focused on continuous improvement. It is a culture change.

Executive and Middle Management Commitment

Management commitment is the single most important requirement for successful implementation of quality management. There is no precedent of successful quality improvement without executive management and the management team leading the effort. Having management commitment does not guarantee quality management success; it only improves the odds for successful implementation. The entire organization must eventually become committed to quality management.

Managers need to set the tone for the organization by driving the process and incorporating the philosophy of quality management into their management styles. They must be prepared for an environmental change, making quality a key responsibility. The top-down implementation model (waterfall effect), starting with executive management, then middle management, line management, and, finally, employees, has proven successful in many organizations.

Committing to quality management implementation means that all management must be willing to:

- Understand the concept of quality management
- Adopt behaviors required to show commitment
- Accept the need to change to participative leadership
- Lead in the development of a quality management implementation plan
- Lead the formation of the implementation organization
- Lead the planning for process improvement teams
- Provide funds for training
- Provide time for training and meetings
- Identify quality standards and measures
- Publicize and reward results
- Monitor and measure progress
- Provide personnel and other resources

Commitment can take many forms. First, it is action that can be measured in time, effort, and money. It is a full-time commitment that is not delegated. Commitment begins by putting quality at the top of every agenda. Looking at management's calendars is a way to measure their current commitment to quality. Another method of finding out where they stand is to survey them before implementation starts. At a minimum, the survey should cover the areas of job satisfaction, organization satisfaction, management satisfaction, quality productivity, and the work environment. The currently perceived relative priorities of cost, schedule, and quality should be determined as well as the actual values. The survey results should be fed back to the managers of the organizations that are surveyed and used to develop the quality management implementation strategy and plan. Possible candidates for the quality management champion may be identified based on the results.

Willingness to participate in the implementation of quality management can also be analyzed by reviewing these key questions:

- Are you willing to change your organization?
- Will you create the environment for change?
- Will you train others and commit resources for that purpose?
- Will you demonstrate commitment by your actions?

- Will you positively reinforce progress?
- Will you commit resources to promote quality management in your organization?
- Do you discuss quality in daily conversations and include it in presentations?
- Do you require quality as part of performance appraisals and reviews?
- Do you review quality in the various aspects of your job?

Starting at the top, management must sincerely believe that the organization can, and must, do better. Employees measure management's commitment by observing their actions. Eventually, everyone in the organization makes a commitment to the perfection of goods and services. Commitment brings the following changes:

Quality comes first among equals of quality, schedule and cost. "How good?" must precede "How many?" and "How much?" Having a product developed on schedule, which meets cost, will not do well in the marketplace if it performs poorly.

Satisfying internal customers as well as external customers becomes a new priority for management and a main objective for the organization. Internal customer-supplier relationships are established if they are not currently recognized. Top management leads this effort by personally maintaining close contact with customers. Knowledge of customers' needs and expectations is a prerequisite to satisfying them.

Management must acquire new skills and perspectives, including the language of statistics. The use of quantitative techniques becomes second nature to the entire organization. Statistical analysis instead of opinion and "gut feelings," becomes the basis for decision-making. This quantitative based management approach will be one of the long-term effects of Quality.

Continuous process improvement techniques are applied to all processes in the organization. The focus of problem solving changes from people to processes, which stresses the need to find and fix root-causes of problems.

Management becomes more active in recognizing success. Implementing quality management is difficult for everyone. By looking for every opportunity to thank people for their contributions, management helps maintain the momentum required for the long haul. Employees are recognized in their work areas, instead of management's office.

Implementing quality management is a long-term effort, requiring a long-term commitment. Management will need to sustain its interest until quality management becomes the way of life in the organization. This change can take a minimum of five to eight years to start and may take decades to complete. Such a long-term commitment will be difficult because management will most likely change during this period of time. New managers must accept and continue the cause of quality management as soon as possible after they enter the organization. A change in executive management causes the implementation efforts of many organizations to fail. Attempting to implement quality management at the bottom of the organization before securing executive management commitment almost always results in frustration. Ideally, quality management should proceed through the rest of the organization after executive management commitment is obtained.

Executive Management Commitment

While overall management commitment is necessary to the success of quality management, commitment from the organization's executives is vital. Executive management sets the tone for the whole effort by visibly supporting quality management. Every employee, including other managers, will wait to see where executives prioritize quality management. If quality improvement is not "number one" with executive management, it will not be with anyone else.

After receiving their initial quality management training, executive managers should operate as a Process Improvement Team (PIT) to improve at least one management process before moving quality management down the organization. (See Skill Category 6 for a discussion of PITs and an eight-step improvement process.) This is a visible sign of commitment and the best way for management to understand what quality management means. Ideally, each layer of management will do the same thing. A challenging process to study is the preparation of the annual budget, but starting with a simpler process may be more appropriate.

Executive management should develop a quality policy and mission, vision, goals, and values statements (see "Implementing a Mission, Vision, Goals, Values, and Quality Policy" on page 112 for more information). They also show commitment to quality management by establishing new quality standards calling for error-free performance. This goal will not be pursued or achieved, if executive management does not establish the need. The standard must be applied to management before it is deployed to the rest of the organization. Executive management must personally strive for perfection, measure progress, and recognize those who contribute to error-free performance. Employees strive to meet expectations established for them, and model their behavior after management's actions.

Middle Management Commitment

As the slowest group to accept the process, middle management is the weakest link in most quality management efforts. Special effort is required to assure them they have a role as important players. They should have input to the statements that executive management prepares, and be included in all aspects of quality management planning and implementation. One way to assure their support is to assign them the task of determining their own role. How to include middle management is an important consideration for executive management, because obtaining quality management support from first-line managers and employees is relatively easy.

Quality Champion

There is a need for one or more people to champion the cause of quality management. Ideally a champion will emerge during the planning for quality management implementation. This is the person who accepts personal responsibility for the success of quality management without being assigned the responsibility. The champion will be emotionally committed to quality management and will see it as a cause. The champion should be respected in the organization, have high quality standards and believe that the organization needs to improve. This "can do" attitude may be the most important consideration. A quality management champion may assume the day-to-day management responsibility for successfully implementing quality management.

Champions happen naturally; they are not appointed. The enthusiasm and energy of champions are important factors in the success of an organization. Ideally, the initial champion would be the top executive, but several managers may assume this role at different times. The need for a champion lasts a minimum of two to three years.

New Behaviors for Management

Implementing quality management requires a culture change. Learning new types of behavior leads to that change. Some new behavior modes are discussed below.

Traditional Management versus Quality Management

Most managers practice traditional management. They have been taught to control their organization and employees, using an “I’ll tell you what to do, and you’ll do it” mentality. Many managers look at the short-term because their commitment to the organization is short range.

The key differences in philosophy between traditional management and quality management environments are illustrated in Table 1.

Table 1. Traditional versus Quality Management Philosophy

Traditional Management Philosophy	Quality Management Philosophy
Controls each result	Use the process
Who made the error?	What allowed the error?
Correct the error	Reduce variation and prevent the error
Employees are the problem	Refine the process
Management accountable to their manager	Management accountable to the customer
Competition between organizations	Teamwork
Motivation from fear of failure	Motivation from within (self)
Management of outputs (results) – focusing on detection of defects	Management of process inputs – methods or sources of variation that focus on preventing defects
Fire fighting	Continuous process improvement
Accomplishment from meeting quotas, the monthly or quarterly bottom line	Accomplishment from long-term impact of improving processes

The culture change required to build a quality management environment is significant. Management must change its philosophy, practices, and assumptions about work and people. The biggest mistake usually made when implementing a quality management environment is underestimating the cultural changes that must occur and the time required for accomplishing these changes. It is usually felt that only a few control charts are needed, and little effort is made to change the culture of the organization.

The programs needed to change from a traditional to quality management culture must be customized for an organization and its current culture. Table 2 illustrates cultural changes that can be made.

Table 2. Quality Management Cultural Changes

Category	Traditional Culture	Quality Management Culture
Mission	Maximum return on investment (ROI), management by objectives (MBO)	Ethical behavior and customer satisfaction, climate for continuous improvement, ROI as a measure of performance
Customer Requirements	Incomplete or ambiguous understanding of customer requirements	Uses a systematic approach to seek out, understand, and satisfy both internal and external customer requirements
Suppliers	Undirected relationship	Partnership
Objectives	Orientation to short-term objectives and actions with limited long-term perspective	Deliberate balance of long-term goals with successive short-term objectives
Improvement	Acceptance of process variability and subsequent corrective action as the norm	Understanding and continually improving the process
Problem-Solving	Unstructured individualistic problem-solving and decision-making	Predominantly participative and interdisciplinary problem-solving and decision-making based on substantive data
Jobs and People	Functional, narrow scope, management controlled	Management and employee involvement, work teams, integrated functions
Management Style	Management style with uncertain objectives that instills fear of failure	Open style with clear and consistent objectives, encouraging group-derived continuous improvement
Role of Manager	Plan, organize, assign, control and enforce	Communicate, consult, delegate, coach, mentor, remove barriers, and establish trust
Rewards & Recognition	Pay by job, few team incentives	Individual and group recognition and rewards, negotiated criteria
Measurement	Orientation toward data gathering for problem identification	Data used to understand and continuously improve processes

Leadership

Leadership and management are two different things. While a manager works within the system following the accepted practices of the system, a leader determines where the organization needs to be, and then does what is necessary to get there. In a business context, leadership is the ability to build the commitment of employees, to endow an organization with a positive perception of itself, and to give employees a positive perception of their role within the business. While programming experience, technical prowess, and management ability may be important qualifications for top-level IT management, leadership ability is the critical element.

Traditional management techniques focus on employee behavior, not the employee. Feelings of achievement, recognition for good work, and a sense of meaningful professional advancement are foreign to many workers, as managers do not know how to make employees feel valuable. Without such messages, it is impossible to build employee commitment. Too many managers lack the formal training, or sometimes even the common sense, to understand that most employees need realistic feedback on their performance. While performance appraisals can be used as a tool

to encourage better work behavior, true leaders provide this type of feedback, naturally, in their day-to-day actions.

In the process of creating desirable standards for business performance, there are three fundamental mistakes a would-be IT leader can make:

- Isolation

Leaders must maintain regular, if not frequent, contact with a significant percentage of the people they manage. Typical criticism centers on their preoccupation with technology, their inability to see the big picture, and their fondness for tasks rather than a genuine interest in the organization.

- Inability to reward

Managers who lack the ability or do not take the time to reward are never able to build employee commitment. When performance is measured against possibility, extraordinary rewards are warranted.

- Lack of business perspective

Business perspective is the ability to take advantage of opportunities, articulate goals, effectively deploy resources, take risks and accept responsibility for the outcome of actions. Business perspective also involves strict attention to cost.

In the business world, no employee consciously defines leadership, but each one instinctively knows what it is and if it is absent. Leadership requires the establishment of a vision, a strategy for achieving that vision, the cooperation required for achieving the vision and, finally, putting in place the organization to continue the vision. The vision management must create, is that of a new culture, which encourages and accepts change. This new culture allows all people to work together to maximize their contributions to quality improvement.

Quality leadership begins with quality management knowledge. It moves towards a strong focus on quality. Leadership includes focusing on the importance of the customer and on teamwork. New behaviors required of quality leaders include modeling, coaching, and reinforcing.

- Modeling

Modeling consists of “Do as I do.” Quality leaders show their employees the preferred way of acting. They use the quality management language, manage with statistics, and participate on process improvement teams. They expect their subordinate managers to do the same modeling.

- Coaching

Coaching means helping others implement quality management correctly. It can take the form of instructing, directing, or prompting others toward desired outcomes. Coaches:

- Instruct when they see others are unsure of how to do something, and help them proceed. Instructions can include correcting, consulting, or reviewing.

- Direct when employees do not know which action to take next. Direction provides the “what” and the “why” by setting priorities for employees or teams.
- Prompt when others need a hint as to the next quality management step to take. Prompting usually follows a request or question.

Using managers as quality management instructors is the ultimate form of coaching and an excellent way to demonstrate commitment. Quality leaders look for situations where one of these coaching behaviors can be used.

- Reinforcing

Positive reinforcement is the deliberate effort to praise people for their accomplishments. Management must make a conscious effort to look for opportunities to praise employees accomplishing quality management. The praise should be immediate and specific to the improvement action. Positive feedback should become second nature to management. Reinforcement is probably the best source of motivation that a leader can provide. A simple “thank you” may suffice. Never assume to know what employees consider praise. Ask them, and then provide the reinforcement they consider appropriate.

Practices that need to be used for IT leadership are:

- Assess the real business needs
A leader must know what the customer or business really wants. Leaders do not accept the stated requirements but probe until they believe the real business needs have been identified, and then develop solutions.
- Provide problem-solving vision
Leaders look for the most effective ways to solve problems, not the traditional ways.
- Understand and work with human nature
A leader understands that solutions come with, and through, people. To make this happen, leaders give workers feedback needed to do their job effectively, and provide them with appropriate rewards when warranted.
- Stay in touch with customers/producers
The leader constantly works with and calls on the parties who have a vested interest in the success of the project. The leader continues to have the pulse of the marketplace and the workers producing the products.
- Set priorities
The leader establishes the priority for work, knowing what must be done first, and ensuring that it is done.

- Set standards for performance

A leader never leaves it up to the workers to determine the required level of performance. A leader establishes clear performance standards, and then works side-by-side with the workers to ensure those standards are met.

Table 3 shows how quality leaders behave with regards to leadership characteristics.

Table 3. The Quality Leader Behavior Model

Characteristic	Behavior Demonstrated
Substance	Helps others achieve needed substance.
Growth	Helps others achieve personal and career growth.
Opportunities	Creates opportunities for others to make uninhibited contributions to the enterprise.
Environment	Creates an environment conducive to performance.
Empowerment	Empowers others.
Obstacles	Removes obstacles to performance.
Support	Helps others do what they decide is in their own best interest.
Coaching, training, education	Coaches, trains and educates others.
Coordination	Helps coordinate the work of others.
Market, Outlets	Creates a market and outlet for the talents of others.
Resources for others	Acquires the resources others need.
Uniquely Equipped	Does what is necessary for success, which others are not capable of doing.
Strategies	Creates a vision, communication and trust through positioning and deployment of self.
Persistent	Pursues tirelessly the mission of the organization through linkage with other leaders on strategic issues.
Ethical, open, honest	Maintains a totally open and honest state with others.

The Importance of Establishing Mentoring Relationships

Mentoring allows the more senior employees a means to pass their experience and insights to junior employees. Mentoring is normally not in a person's job description, but it is an activity that should be encouraged and rewarded. Many certified quality professionals, i.e., CSQA and CSTE, mentor certification candidates on how to study for, and pass, the certification examination.

Mentoring can include how to deal with organizational politics, how to prepare for job opportunities, and how to solve job work challenges. Quality professionals should actively promote mentoring programs in their IT organization.

Establishing Trust

One of the first questions employees ask during quality management implementation when management is asking them to help improve quality is, “Can we trust management?” Mutual trust is mandatory for building management-employee process improvement teams. Management must trust employees enough to ask for their help and to share with them the responsibility for continuous process improvement. Employees must trust management enough to respond to this request by contributing their knowledge and their ideas.

Actively listening down, not talking down, can be the first step in establishing trust. Listening down helps establish two-way communication, and carries two-way responsibility. Those being listened to should have something constructive to say. Ideas for improvement, not yesterday’s complaints, are needed. Those doing the listening need to be ready to react to the ideas that will be generated. Pushing responsibility down the organization also helps build trust. Decisions for process improvement should occur at the lowest possible level in the organization. Decisions that must be elevated should be responded to in a timely manner. If an idea cannot be used, an explanation should be given to those making the recommendation. A timely “no thank you because” will be accepted.

Competition to Cooperation

While competition is “the American Way,” within an organization competition is a hindrance to quality improvement – it stifles communication and creates barriers between functions.

Cooperation is required to improve quality and to implement quality management. Cooperation leads to increased creativity, reduces fear of censure, and helps the development of a better sense of belonging and acceptance. These are all characteristics of high-quality organizations. Teamwork, one of the cornerstones of quality management, is based on cooperation. To facilitate cooperation, the organization needs to focus on teams doing well, allow ample time for teams to operate and achieve improvements, use the language of quality management, practice reciprocity, share information, and act cooperatively. (Stages of team development are discussed in “Understanding Team Development Phases” on page 85.) Cooperation requires objectivity. An atmosphere based on “I win, you win,” as opposed to “I win, you lose,” must be established.

Awareness Training

Understanding precedes behavior change. It is difficult to understand and deal with a new concept without being aware of its existence. Awareness training should create knowledge of the defined topic and initiate some action associated with that topic; however, these objectives do not have to occur simultaneously. For example, create awareness about the effectiveness of a software inspection technique, and start the process to implement that technique a few weeks later. The delay allows time to assimilate the concept before beginning action. Until people understand an area and its impact, they are not prepared to act.

Developing a program for awareness training involves the following two steps. Each step consists of five tasks.

1. Prepare for Awareness Training

Task 1: Select awareness topic. The topic is usually a problem or a new approach to doing work, and is normally related to accomplishing the organization's mission. Sample topics include the number-one cause of operational problems, or a new work approach, such as implementing quality principles in the IT group.

Task 2: Identify the topic's customers. Individuals with a vested interest in the topic need to have the awareness training. These are the people who will benefit from or be adversely affected by the topic. For example, for software inspections, customers would be the individuals whose products are being inspected as well as those performing the inspection. Customers could also include users of those products because inspections affect delivery date and quality.

Task 3: Define objectives for awareness training. The objectives relate to the action that is initiated as part of the awareness training, indicating the outcome or results to be accomplished relating to the topic. In the topic of software inspections, the objective may be to initiate software inspections in the systems development process. The training would focus on accomplishing those objectives.

Task 4: Define customer benefits. Defining and selling the benefits to the customer is an important part of the training. To define the benefits, identify each category of customer, and then determine what benefits that customer would receive if the training objectives were accomplished. Some benefits will be negative, in which case the training must be supportive of that loss of benefit.

Task 5: Develop administrative training plan. A plan needs to be developed for conducting the training. The plan contains information related to the awareness topic as well as general administrative activities. The content of the awareness topic is discussed in Step 2; however, in actual practice, the specific topic would need to be developed in this task.

Administrative activities to consider when planning awareness training are: identifying attendees (limit sessions to 25 people), inviting attendees (homogenous groups work best), arranging a training room and equipment, preparing an agenda (limit training to 2 hours maximum), arranging handout materials, soliciting a high-ranking person (i.e., champion) to introduce the topic, and assigning training responsibilities.

2. Conduct Awareness Training

In addition to introducing the topic, attendees, and trainers, the awareness training should cover these five basic topical areas:

Task 1: Attendees' needs. Begin the session by stating the topic, and relating it to the attendees' needs. The needs should relate closely to the objectives and benefits of the awareness topic. Some research may be necessary to assure that the true needs of the attendees are identified.

Task 2: Awareness topic/product. Describe the product/activity that is to be addressed or is a solution to the problems identified in the need. For example, software inspection would be the activity to satisfy the need for on-time delivery and high quality, while a description of operational abnormal terminations would be the problem to be solved. The rest of the awareness session deals with solving the problem or getting a product/activity implemented.

Task 3: Identify objections to product/problem. Attendees may object to the severity of the problem or the product that is being proposed. For example, if abnormal terminations are discussed, they might consider the current level to be normal; and if software inspections were implemented, they would object to the time taken to conduct an inspection. The purpose of this task is to ensure that all the objections are clearly identified.

Task 4: Overcome objections. Having objections is a normal and positive step of change. Dealing with the objections is the key to initiating change effectively. In this step resolution of the customer's objections is addressed.

Task 5: Recommend course of action. So that subject matter is not forgotten, awareness training should always end with some action to take. If attendees are charged with performing some action, or will shortly become involved in some action, the training will become effective and practical.

Nurturing New Behaviors

The new behavior patterns discussed above must replace old habits and become management's customary practice. Establishing a structure or mechanism for ensuring this will happen is recommended. For example, devoting the first several minutes of each staff meeting to reviewing quality management progress will establish the habit (behavior) of tracking and measuring quality management progress. Adopting the habit of "management by walking around" reinforces the concept of "listening down." Management at all levels must continuously look for opportunities to review and change their behavior. While old habits are hard to break, the new behavior of management is a powerful sign to the rest of the organization that quality management is the new way of life.

After quality management has been implemented for two or three years, management's commitment should include the following:

- Continuing self-education
- Demonstrating by action the change to participative leadership
- Being patient
- Supporting teams with resources and recognition
- Randomly attending team meetings
- Responding promptly to issues elevated to their level
- Managing using quantitative techniques and statistical tools

- Institutionalizing quality management

"...quality and productivity are two sides of the same coin. Everything you do for quality improves your product."

Lee Iacocca, Past CEO, Chrysler Corporation

Empowerment of Employees

Empowerment is a process that transfers decision making from management to employees. It is a concept that enriches an employee's job and enables quick actions to be taken. Customer and user satisfaction is normally increased when they can get quick action from front line staff.

To be effective, empowerment must be part of a process. The empowered employee must know their limits of empowerment, and when and how they can make decisions. For example, a help desk employee may be empowered to hire a courier to deliver a report needed quickly by a user; but only up to a cost of \$100. Another empowerment example would be in a "charge-out" environment where a programmer can fix a program at no charge to the user if the programmer believes the problem was caused by incorrect programming.

Quality Management Infrastructure

The reason a quality management environment is established is to assure constancy of purpose in promoting quality as a major IT goal. There are two components to that environment: belief and commitment from management and staff, and the organizational structure and quality initiatives to support that environment. Part 1 focused on the belief and commitment of the management team. This part focuses on the infrastructure and initiatives.

No organization has a perfect quality management environment. All are striving to achieve the optimum management philosophy, and organizations can be anywhere along the quality management continuum. By forming a quality function, some level of commitment and organizational structure exists that could be called quality management.

There are three approaches to quality management implementation: bottom-up, middle-out and top-down.

- The Bottom-up Approach

This approach sends the message that quality management is something for the employees, but not necessarily for management. This approach is like swimming against the current, and leads to frustration because resources are not readily provided to teams when required. Process improvement can be accomplished, and it can be successful, but with an inordinate expenditure of time and effort.

- The Middle-out Approach

Starting in the middle of the organization and then progressing simultaneously to the top and bottom of the organization can be successful. The degree of success depends on how fast the organization proceeds to the top. It presents many of the same problems as the bottom-up approach.

- The Top-down Approach

Top-down has the highest probability for success, although success is not guaranteed. This model fosters management involvement - the single most important requirement for quality management success. In addition to commitment, it also requires that management lead the effort, providing a quality management vision and philosophy for the organization. Management must lead the cultural change required. Time, money, and people will be required, and the top-down approach assures the availability of the resource support that is crucial to quality management success. Quality improvement is not free; it is an investment.

In some organizations there is no choice but to begin implementation from the middle or the bottom. In these cases, demonstrated success will be required to get management's attention. While all three approaches have been used and have been successful, the top-down approach is recommended, and is used for the remainder of this discussion.

For a top-down implementation of the infrastructure in a quality management environment, begin the process with executive management. Then facilitate the downward flow of the goals, values, structure, and training established at upper levels, to succeeding levels. Each level is linked to the other by the common objective of making people capable of combined performance. Figure 9 shows the three levels of infrastructure normally needed.

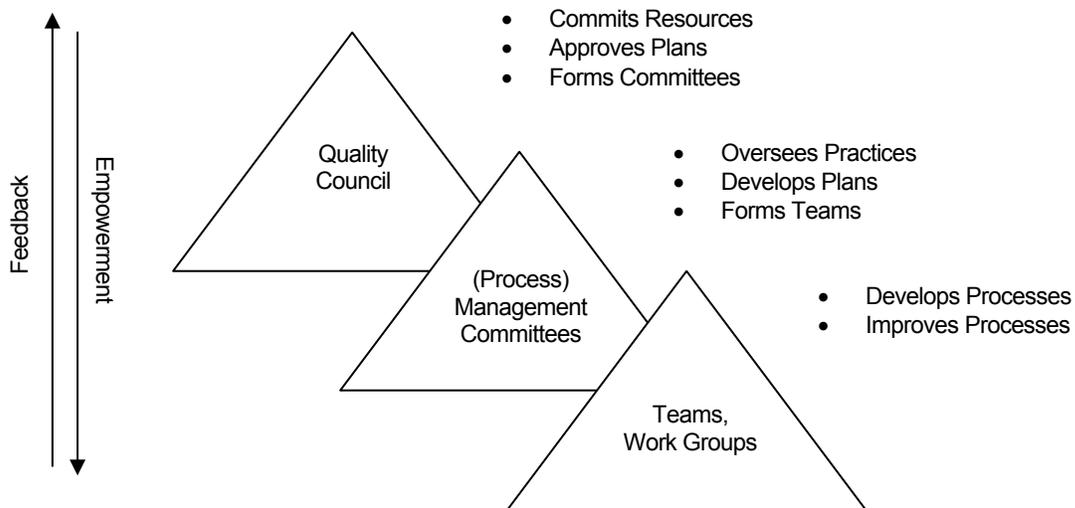


Figure 9. Quality Management Infrastructure

Quality Council

A Quality Council is composed of the organization's top executive and his or her direct reports. It may also be referred to as an Executive Council. The Quality Council acts as the steering group to develop the organization's mission, vision, goals, values, and quality policy. These serve as critical input to process mapping, planning, measurement, etc. Some large companies opt for more than one level of Quality Council. When multiple levels exist, each organization's mission and vision tie into that specified by the top council. Specifically, the Quality Council:

- Initiates and personally commits to the quality management philosophies and practices.
- Incorporates this decision into the strategic planning process, allocating resources in the budget for the deployment of quality management, and ensuring resources are available for both ongoing and upcoming IT projects and internal process improvement projects.
- Establishes committees at lower levels to focus on functional and cross-functional improvement efforts, to develop or revise processes, and to oversee and manage the quality management process on a daily basis. They may develop charters to serve as job descriptions for the committees, or approve the committees' charters.
- Defines and deploys policies.
- Recommends critical processes for analysis.
- Makes the decision regarding whether to approve, reject, or table (pending further investigation) new or changed processes.
- Acts on unresolved process problems and issues referred by the committees.
- Provides review and oversight of progress.

Management Committees

Management committees (also called Process Management Committees) are composed of middle managers and/or key staff personnel, and are responsible for deploying quality management practices throughout the organization. One or more committees may be needed depending on the organization's size and functional diversity. Committees should represent all the skills and functions needed to work on the specific processes or activities. They:

- Work with the Quality Council to understand the organization's mission, goals, and priorities. They either review the charter provided by the Quality Council or develop one. They also develop and maintain a deployment plan that identifies and prioritizes which key processes need to be defined and improved.
- Develop, or commission the development of, and maintain a process inventory and process maps (see Skill Category 7).
- Analyze processes, at the direction of the Quality Council, and identify those that need priority attention. This includes proposing new processes and/or revising existing processes.

- Establish teams or work groups (they may participate on the teams) to define and improve processes, and provide support to the teams (training, coaching, facilities, approaches, standards, tools, etc.). They monitor team progress and review/approve the resulting processes.

Teams and Work Groups

Teams are formed under any number of names depending on their purpose. Common names and functions are:

- **Process Development Teams** develop processes, standards, etc.
- **Process Improvement Teams** improve existing processes, standards, etc.
- **Work Groups** perform specific tasks such as JAD, inspection, or testing.

The process teams are composed of a representative group of process owners. Members of work groups will vary depending on the purpose, but suppliers and customers are likely to participate as team members or as reviewers. It may also be desirable for a QA analyst to participate on the team.

Process teams use standard approaches (such as flowcharts, checklists, Pareto analysis) to define, study, and improve processes, standards, procedures, and quality control methods. They may pilot new or revised processes help deploy them to the rest of the organization, and provide process training. They may also serve as process consultants to process owners and others using the process. Approaches and tools used by the team depend on its purpose.

Guidelines for teams include the following:

- The process development committee selects teams.
- Each team should have a chairperson.
- The core team should be small, containing 3-5 people.
- Each team should have a work plan that outlines the tasks to be performed and assigns tasks to team members.
- The team should meet regularly to review work performed by individual members and to review progress.
- Different team members may draft different portions of the processes and procedures.
- The team must reach consensus before submitting results to the process management committee for approval.

Teams Don't Always Work

When it's definitely determined by a team that something cannot be done, watch somebody go ahead and do it.

Understanding Team Development Phases

The use of teams is critical in a quality management environment. As a result, understanding the team life cycle is important in order to set proper expectations for the team and to help it communicate and function effectively. Teams go through four phases:

- **Forming**

In this first stage, teams are dominated by feelings of confusion and anxiety, and are not able to focus on their purpose for long. Individuals may come to the team proud to be selected, but wondering why, and wondering about the other members. Information will be solicited and shared, and hidden agendas add to the uncertainty. Key accomplishments of this phase are identifying roles for team members, clarifying responsibilities and accepted behavior, and defining the team's purpose.

- **Storming**

Conflict, defensiveness, and competition are key during this stage. Team members still think individually and wrestle with loyalties outside the team. As ideas emerge, they are attacked and defended. There may be confrontations, disagreements, and fluctuating attitudes over the likelihood of achieving the team's purpose. Barriers will be examined and the team will focus on well-known observations and common beliefs. Some people will not participate to prevent unfavorable responses, and others will test the leader's authority and form cliques.

- **Norming**

In this stage the individuals start to become a team. Personal agendas, concerns, and loyalties are minimized. People are discussed less often than the issues, conflicts are resolved constructively, and the team focuses on its real purpose. As trust develops, riskier ideas are proposed and feelings exchanged. The willingness to discuss for the sake of the team grows, which results in better communication and cooperation.

- **Conforming**

During this final stage, the team has matured into a cohesive unit. Individual strengths and weaknesses are understood and appreciated, leading to an overall satisfaction with the team membership. As steps are made toward the team's goals, there is individual learning and growth, and people feel satisfied with progress.

There are many variables that affect the length of time a team spends in each of these stages. The team experience of individual members is a big factor, and use of a facilitator can help. Clarity of the team's purpose and the level of management support are other factors. Teams may also get to the norming or conforming stages and fall back to earlier stages if assumptions are found to be incorrect or team membership changes.

Establishing Group Compatibility

Fundamental Interpersonal Relations Orientation (FIRO) is a technique to build group compatibility. This theory was created by W.C. Schutz to explain an individual's orientation toward others based on his or her interpersonal behavior. Understanding FIRO will help the QA

analyst select a good mix of people for a group activity, or to better understand the interpersonal relationships occurring in a group. The concept explains people orienting themselves toward others in certain characteristic patterns. Similar patterns among group members yield a group that is more compatible and efficient.

To understand the potential compatibility of group members, the interpersonal characteristics of each individual must be understood first. Characteristics of an individual can be explained in terms of the three interpersonal needs below:

- Inclusion

Inclusion is the need to associate with others and the need for togetherness. This need manifests itself through behaviors designed to attract attention. A person with a strong need for inclusion may be overly friendly, amiable, and possessive. They may punish friends who attempt to establish friendships with others.

- Control

Control refers to the decision-making process between people. The need for control varies from the need to dominate others versus being dominated by others. A person with a high need to be controlled is compliant and submissive to others; a person with a high need to control displays rebellion and refusal to be controlled.

- Affection

Affection refers to close personal and emotional feelings between two individuals. Love and hate represent the two extremes. A person with a strong need for affection will be friendly, make overtures to others, and generally tries to establish close emotional ties with others. A low-need person will avoid close interpersonal relations.

When two or more people interact, each one typically enacts in the need area of the characteristic behavior pattern that was developed in childhood. These patterns are often a direct result of the way a child was treated by his or her parents or other adults and how that child reacted. The interaction patterns of any two given individuals may be either compatible or incompatible. If they are compatible, then the interaction is likely to be easy and productive. If they are incompatible, the interaction is likely to be difficult and unproductive.

Three types of compatibility-incompatibility have been identified that could occur in each of the three need areas:

- Interchange Compatibility

This is based upon the mutual expression of inclusion, control, or affection. Interchange compatibility depends upon the degree to which those interacting agree on the desired amount of mutual interaction. Some people prefer a great deal of behavior exchange relevant to the need, while others prefer not to receive, or to send inclusion, control, or affection. Interchange compatibility exists when the two persons interacting desire a similar amount of exchange. People are incompatible when one prefers a high rate of exchange in the area of affection and the other prefers a low rate of exchange.

- **Originator Compatibility**

This derives from the originator-receiver dimension of interaction. Two persons are compatible to the degree that the expression of inclusion, control, or affection corresponds to that which the other person wishes to receive. For example, if one person needs to control and tries to dominate another person that needs to be submissive, they will be compatible. If both need to control and try dominating each other, they will be incompatible. Similarly, if one person initiates group activities for another person who wants to be included in the activities, they will be compatible. However, initiating group activities for a person who does not want to be included leads to incompatibility. The degree to which the activities originated by one person are in accord with the needs of the other member is important.

- **Reciprocal Compatibility**

This reflects the degree to which two persons reciprocally satisfy each other's behavior preferences (the degree to which each person's behavior is in accord with the other's needs). If one person wants the other to express much affection and the other does so, there is compatibility in the area of affection. But if one member is frustrated because the other doesn't express enough affection, incompatibility results.

The general assumption of Shutz's theory is that compatible groups will be more efficient than incompatible groups. This effect is reflected in the initial formation of groups, in the degree to which the groups are likely to continue to function, and in the productivity of groups.

Consensus

During problem solving and process improvements, teams are faced with making decisions on a number of activities: what problems need to be worked out, data collection, steps to use, conclusions, solutions, recommendations, presentations, etc.

With the consensus technique, each member must accept the agreed-upon resolution and be willing to support it, even if it was not their favorite choice. Everyone participates and there is no voting. Using consensus provides teams an opportunity to reach high-quality decisions with total team commitment. Team members have an in-depth understanding of the underlying concern or issues, trust, willingness to explore, and mutual respect for each other.

An effective consensus process is best accomplished in a relaxed environment away from the work place, with a trained facilitator and a team of between five and nine people. A facilitator helps to seek out participation, draw out information, keep the team focused, provide encouragement, suggest approaches, and maintain order. The facilitator needs to be impartial and neutral with no stake in the resolution.

Consensus is the most difficult decision-making process compared to authority, voting, or avoidance (no decision).

Controlling Meetings

The most vital process of any negotiation is to understand what goes wrong in meetings and to identify problem areas. Individuals must know the difference between the content and process of

the meeting. Meetings should be conducted by consensus, with each attendee interacting and contributing to its overall success. The following steps help ensure the success of meetings:

1. Prepare for the meeting. Plan the agenda, noting the time to be spent on each item.
2. Ensure attendees know their roles and are encouraged to carry them out at the meeting.
3. Identify issues. Use intervention techniques to handle difficult people.
4. Create a stimulating environment so that the participants do not feel threatened to voice their opinions.
5. Encourage attendees to contribute fully, asking questions to help.
6. Repeat and note suggestions so that everyone understands what is said.
7. Help reach clear agreement on the issues.
8. Repeat decisions so that there is no miscommunication.
9. Set a time to implement the decisions that are reached.
10. Identify responsibilities of who will do what, by when, to accomplish the tasks. Establish a time for a follow-up meeting.

Using Task Forces Effectively

A task force is a cross-functional activity organized for a specific purpose. Research has shown that task forces are better for decision-making than for solving problems. If problem solving is needed, a committee should be established, made up of people who work with the problem every day. Note that the task force's decision may be to organize a committee, or to hire a consultant to solve the problem.

The successful use of the task force requires implementing effectiveness principles. These principles can be divided into the following two areas.

- **Task Force Management Principles**
 - The task force leader should be an expert in leading groups (such as a trained facilitator), not necessarily an expert on the issue before the task force.
 - Task forces should be organized for a single purpose. Several related issues might be more appropriate for multiple task forces.
 - The task force needs to have a clear description of what is to be addressed.
 - The task force members should include all areas with a vested interest in the decision to be made. The individuals should be the best people available in the organization to deal with the issue. They should have knowledge of the situation and the business, open-mindedness, and some clout in making things happen.

- Output from a task force should be a short report containing a recommended course of action that is presented as a unanimous decision.
- The task force should be dissolved once the report has been completed.
- Task Force Organizational Principles
 - A task force should contain 3-8 members. Less than three prevents synergism from occurring. More than eight can cause organizational problems for the leader.
 - Someone should be appointed to record all significant information. The leader can do it or assign another team member.
 - The task force work should begin as soon as the need is recognized. It should not wait for all the facts since some of its responsibilities will be data gathering.
 - Meet on neutral ground, such as a conference room.
 - Meet as frequently as needed to finish the task force business as quickly as possible.
 - Do not discuss task force business with non-team members while the task force is meeting, and preferably not at all. The report should be the only task force output.

Personal Persuasion

People receive most information through visual intelligence. Image (visual intelligence about a person) is how others perceive a person. Their perception normally depends on how that person is viewed within the corporation. If management has a high image of that person, the probability of having his or her recommendations accepted and being promoted is significantly higher than when a negative image is projected. A person has an image problem if peers appreciate his or her skills more than superiors do.

Management relies on technical people for their technical capabilities. However, many managers believe it is easy to buy technical capabilities. It is difficult to obtain good managers, but the difference between a good technician and a good manager is frequently image.

Everyone has an image of what an executive looks like. That image is normally shaped through role models. By looking at the dress of corporate officers or other very successful people, an image is developed of how successful people should look and act. Being male or female is irrelevant regarding how an image is projected. One would expect some differences in dress and actions, but basically the same attributes of good image apply to both male and female.

James R. Baehler, author of *The New Manager's Guide to Success* defined these six basic attributes of executive management:

Purposeful	Stands and sits straight. Walks briskly. Knows where he or she is going and how to get there. Looks people directly in the eye.
Competent	Organizes thoughts before speaking; is brief, simple, specific, and direct. Stays calm. Does not appear rushed or harried.
Analytical	Does more asking than telling; listens to the answers. Does not accept generalities.
Decisive	States the problem, then the solution. Always talks straight. Does not waste time.
Confident	Talks about challenges, not obstacles. Is not tense with superiors. Knows the art of casual conversation.
Appearance	Dresses up one level.

Conformity Behavior of Individuals in a Group

Studies indicate that individuals tend to conform to a perceived group standard. This research finding has been demonstrated repeatedly and has become one of the "givens" of industrial psychology. While it is not a complex skill to understand and observe, it can be difficult to put into practice effectively.

Conformity behavior states that individuals in a group tend to conform to an acceptable standard of behavior by the group. For example, people walk into an elevator and face the door. Facing the door is a perceived group standard that has general acceptance, therefore, when one person in the elevator faces the door, others coming into the elevator will do the same.

Establishing the group standard is more difficult than conforming to it. A study by M. Blake Lefkowitz, as reported in the *Journal of Abnormal and Social Psychology*, indicated that there is greater conformity to a high-status person than to a low-status person. The study showed that if a high-status person crossed a street against the light, more people would follow that person than would follow a low-status person crossing against the light. This indicates that one factor used in establishing a group standard is the status of individuals setting that standard. In the Lefkowitz study, status was established by dress and image, but could be expanded to include such things as leadership, technical skill, and years of experience when equated to an IT environment.

The key elements in conformity behavior are:

- Identifying a group

Establish the group for which conformity of behavior is desired. In data processing people may be grouped into such categories as programmers, the data processing department, end users, etc.

- Desired group standard

Establish the standard for which conformity behavior is wanted. Groups at large tend to establish their own standard, but, in the performance of work, the manager likely sets the group standard. Or, it could be established by an independent party and then introduced into the group.

- How to deal with nonconformity

Society has its way of dealing with nonconformity, such as exclusion from the group, nasty comments, and so forth. However, when conformity behavior is used to change the environment, it must be dealt with.

To modify behavior, select a behavior change that can occur through group dynamics. Then select a respected leader and give him or her the desired standard. The leader now conforms to that standard, and, if group behavior works correctly, the group will follow that behavior without extensive instruction. For example, a quality management environment stresses the importance of communication. In support of this concept, senior management may choose to set a new standard of an open-door policy. Senior management would discuss the value of this with their managers and provide positive feedback to those opting to use the open-door policy. Managers would see the benefits and would in turn follow their manager's lead. Group dynamics would result in the majority of the management chain opting to institute the same open-door policy.

Resolving Customer Complaints

Complaints are the customers' way of indicating they are having a problem. Quality promotes turning problems into opportunities. Thus, while resolving a customer complaint the opportunity can be used to improve customer relationships.

Research shows that complaints must be resolved within four minutes and the customer should be receiving a solution to his or her problem. Dr. Leonard Zunin, a human relations consultant, in his book *Contact: The First Four Minutes*, states that unless a customer is satisfied within four minutes, the customer will give up on you. They will sense that you have not accepted the urgency of the problem they are expressing to you. You have not accepted the problem as your problem, and you are not the one to solve their problem.

To resolve the customer's problem, execute the following four-step complaint-resolution process:

1. Get on your customer's wavelength – The first step in the resolution process is to show concern about the customer's problem by performing the following acts:
 - Get on the same physical wavelength. Establish a position for mutual discussion. Stand if your customer is standing, or ask the customer to sit and then sit after the customer complies.
 - Give undivided attention to the customer. Comments to a secretary or receptionist, such as, "Do not interrupt us," show sincere interest.
 - Physically display interest. Assume a body position, gestures, and tone of voice that show concern.

- React positively to the customer's concern, showing empathy. For example, if the customer indicates you have caused great inconvenience to your customer's staff, apologize for causing this type of problem to occur.
2. Get the facts – The problem cannot be dealt with until the problem (not symptoms) is known. An angry person more likely tells symptoms than problems. As a second step:
 - Ask probing questions. Request an example of the problem, samples of defective products, and sources of information.
 - Take detailed notes. Write down names, amounts in question, order numbers, dates or times at which events happened, and specific products and parts of products where problems occurred.
 - Obtain feelings and attitudes. The problem may be more emotional than factual, but emotions need to be dealt with. Find out how a person feels about what has happened; find out what his or her colleagues or boss feels about the problem.
 - Listen carefully, through the words, to what is being said so that the real problem can be identified. See “Achieving Effective Listening” on page 108 for details.
 3. Establish and initiate an action program – Third, even if the complaint does not appear reasonable, action still needs to be taken to determine the validity of the facts, and to pacify the complainer. In taking action:
 - If you are responsible for the error, admit it and apologize for it. Do not minimize the seriousness of the error.
 - Negotiate a satisfactory resolution with the customer by suggesting a solution and getting agreement. State the solution again, to ensure customer agreement. The solution may be to conduct an investigation and follow-up with the customer to determine next steps.
 - Immediately take the action that was agreed to. Just as it is important to begin communicating a solution within four minutes, it is important to resolve the action quickly.
 - Note - if you are not personally responsible for the problem, still be empathetic; talk about the steps that you will take to get the issue resolved. If resolution of the issue requires another person, make sure to communicate the name of that person and his or her contact information to the customer.
 4. Follow up with the customer – Fourth, after the agreed upon action has been taken, follow up with the customer to ascertain that the result is satisfactory. If the customer remains unsatisfied, return to step 3 and renegotiate a solution. The problem could be a difference in believing what the solution was. Words do not always convey exactly what was meant.

Written Reports

While QA analysts write many types of documents, reports to management are the focus of this section, because written reports are often used to judge the QA analyst's ability to write.

Good ideas are of little value unless they are accepted and implemented. The QA report is designed to convey information and to change behavior. QA analysts write a report, distribute it, and follow up on the recommendations. The value of the quality function can be rated on whether management accepts the report. Thus, the report must be comprehensive, identifying the scope, explaining the factual findings, and suggesting recommendations. The report must be written clearly and effectively enough to cause action to be taken, and must include all information necessary to attain that end.

To write a good report the QA analyst should perform these ten tasks:

1. Establish report objectives and desired management actions

Writing a successful report requires a clear understanding of both the report objectives (what the QA analyst hopes the report will accomplish) and the desired action (what the QA analyst wants management to do after reading the report).

2. Gather factual data (i.e., findings) and recommendations

Ensure that relevant evidence supporting the data and recommendations is incorporated into the report. Failure to include this will adversely affect the credibility of the quality function and management will almost certainly disagree with the factual information in the report.

3. Develop a report outline

A good report has no more than three objectives and three actions. Too much data or too many requests overwhelm the reader. If several items need reporting to management, rank the objectives according to priority, and report only the three most important. List small items in an appendix or a supplemental letter.

4. Draft the report

General principles of writing any report apply to a QA report. Consider using the presentation tools discussed in Skill Category 4. The QA analyst should also remember the following potential problem areas:

- Keep the quality-oriented language at a level that can be understood by management, and explain any technical jargon of quality.
- Provide enough information to make implementing the recommendations possible.
- Ensure there is adequate time to write the report.

5. Review the draft for reasonableness

The author should review the report to verify that the data gathered adequately supports the findings and recommendations, and that the information is presented clearly.

6. Have the report reviewed for readability

At least one person other than the author should look at the report objectively, from the perspective of the target audience, to assess the impression the report will make on its readers, and the impact it will have in changing managerial behavior. Appearance, wording, and effectiveness of the report are evaluated by considering the following questions:

- Does the report appear to have been developed by a professional and knowledgeable group?
- Do I understand what the report is trying to tell me?
- Would a person associated with the report topic find the information in the report offensive or disparaging? If so, would they be more concerned with developing countermeasures than with implementing the recommendations?
- Does the report adequately build a case for implementing the recommendations?
- Does the report clearly differentiate between important and less critical items?

7. Review the report with involved parties

To recognize the importance of findings and recommendations, they should be discussed with affected parties before issuing the final report so that their support can be solicited.

8. Review the report with management

When the report is complete, the QA analyst should meet with management to explain the report and to obtain their concurrence. Any issues should be addressed and corrected.

9. Finalize the report

After incorporating any review comments make any final edits to the report.

10. Distribute the report and follow up

Distribute the final report to the appropriate parties, and follow up to ensure that appropriate action is taken.

Process Improvement Teams

Improving process is most effective when the users of that process are involved in improving the process. The users of the process know the strengths and weaknesses of the process; they know the type of process changes that can facilitate the use of the process.

Process improvement in most organizations is performed by a team of stakeholders. The stakeholders are those individuals who have a vested interest in improving the process. The

organization and operation of the team, is the same as any other team. The preceding section has addressed the team dynamics. They are applicable to process improvement teams.

Skill Category 6, which addresses defining, building, implementing and improving work processes, has a process to be followed to improve processes. Process improvement teams should use that or an equivalent process as a means for improving the work processes.

Quality Environment

The quality environment is the totality of practices that management uses that effects how workers performed. It is the attitudes, values, ethics, policies, procedures and behavior of management that sets the example for work in the organization. For example, if management is ethical and customers over pay their accounts, they will be refunded the overpayment. If IT management recognizes that they would not have any work task to perform for the users, then the users will be treated as very important people and their desires would be important to the IT organization. On the other hand, if IT users are viewed as not knowing the requirements and over demanding, they will be treated as unimportant to the IT organization.

In business and accounting literature, the environment is referred to in many different ways. In accounting literature it is sometimes called the “control environment,” other times the “management environment,” and sometimes just the environment in which work is performed. For the purpose of this skill category, we will refer to it as the “quality environment.” What is important to understand is that the environment significantly impacts the employee’s attitude about complying with policies and procedures. For example, if IT management conveys that they do not believe that following the system development methodology is important, project personnel most likely will not follow that system development methodology. Likewise, if IT management conveys a lack of concern over security, employees will be lax in protecting their passwords and securing confidential information.

The quality environment has a pervasive influence on the way business activities are structured, objectives established, and risks assessed.

The Six Attributes of an Effective Quality Environment

Five major accounting associations (i.e., Financial Executives International, American Institute of Public Accountants, American Accounting Association, The Institute of Internal Auditors, and the Institute of Management Accountants), which is referred to as COSO (Committee of Sponsoring Organizations), was organized to provide guidance on evaluating internal control. They issued this guidance as the COSO Internal Control Framework. The COSO Framework identified the six quality attributes. For each attribute, they listed several control objectives that if implemented would define each of the six attributes.

The six attributes are briefly described below together with the COSO control objectives for that attribute.

The following six attributes are the key attributes of an effective quality environment:

- Integrity and Ethical Values

Management must convey the message that integrity and ethical values cannot be compromised, and employees must receive and understand that message. Management must continually demonstrate, through words and actions, a commitment to high ethical standards. The control objectives for this attribute are:

- Existence and implementation of codes of conduct and other policies regarding acceptable business practice, conflicts of interest, or expected standards of ethical and moral behavior.
- Establishment of the “tone at the top” – including explicit moral guidance about what is right and wrong – and extent of its communication throughout the organization.
- Dealings with employees, suppliers, customers, investors, creditors, insurers, competitors, and auditors, etc. (e.g., whether management conducts business on a high ethical plane, and insist that others do so, or pay little attention to ethical issues).
- Appropriateness of remedial action taken in response to departures from approved policies and procedures or violations of the code of conduct. Extent to which remedial action is communicated or otherwise becomes known throughout the entity.
- Management’s attitude towards intervention or overriding established controls.
- Pressure to meet unrealistic performance targets – particularly for short-term results – and extent to which compensation is based on achieving those performance targets.

- Commitment to Competence

Management must specify the level of competence needed for particular jobs, and translate the desired levels of competence into requisite knowledge and skills. The control objectives for this attribute are:

- Formal or informal job descriptions or other means of defining tasks that comprise particular jobs.
- Analyses of the knowledge and skills needed to perform jobs adequately.

- Management’s Philosophy and Operating Style

The philosophy and operating style of management has a pervasive effect on an entity. These are, of course, intangibles, but one can look for positive or negative signs. The control objectives for this attribute are:

- Nature of business risks accepted, e.g., whether management often enters into particularly high-risk ventures, or is extremely conservative in accepting risks.

- Personnel turnover in key functions, e.g., operating, accounting, data processing, internal audit.
- Management's attitude toward the data processing and accounting functions, and concerns about the reliability of financial reporting and safeguarding of assets.
- Frequency of interaction between senior management and operating management, particularly when operating from geographically removed locations.
- Attitudes and actions towards financial reporting, including disputes over application of accounting treatments (e.g., selection of conservative versus liberal accounting policies, whether accounting principles have been misapplied, important financial information not disclosed, or records manipulated).

- **Organizational Structure**

The organizational structure shouldn't be so simple that it cannot adequately monitor the enterprise's activities nor so complex that it inhibits the necessary flow of information. Executives should adequately understand their control responsibilities and possess the requisite experience and levels of knowledge commensurate with their positions. The control objectives for this attribute are:

- Appropriateness of the entity's organizational structure, and its ability to provide the necessary information flow to manage its activities.
- Adequacy of definition of key managers' responsibilities, and their understanding of these responsibilities.
- Adequacy of knowledge and experience of key managers in light of responsibilities.
- Appropriateness of reporting relationships.
- Extent to which modifications to the organizational structure are made in light of changed conditions.

- **Assignment of Authority and Responsibility**

The assignment of responsibility, delegation of authority and establishment of related policies provide a basis for accountability and control, and sets forth-respective roles in the organization. The control objectives for this attribute are:

- Assignment of responsibility and delegation of authority to deal with organizational goals and objectives, operating functions and regulatory requirements, including responsibility for information systems and authorization for changes.
- Appropriateness of control-related standards and procedures, including employee job descriptions.
- Appropriate numbers of people, particularly with respect to data processing and accounting functions, with the requisite skill levels relative to the size of the entity and nature and complexity of activities and systems.
- Appropriateness of delegated authority in relation to assigned responsibilities.

- **Human Resource Policies and Practices**

Human resource policies are central to recruiting and retaining competent people to enable the entity's plans to be carried out so its goals can be achieved. The control objectives for this attribute are:

- Extent to which policies and procedures for hiring, training, promoting and compensating employees are in place.
- Appropriateness of remedial action taken in response to departures from approved policies and procedures.
- Extent to which personnel policies address adherence to appropriate ethical and moral standards.
- Adequacy of employee candidate background checks, particularly with regard to prior actions or activities considered to be unacceptable by the entity.
- Adequacy of employee retention and promotion criteria and information-gathering techniques (e.g., performance evaluations) and relation to the code of conduct or other behavioral guidelines.

The Core Values and Concepts Included in the Malcolm Baldrige National Quality Award Model

The Malcolm Baldrige National Quality Award Model was established by an act of the United States Congress. It is a model used by organizations to assess, or have assessed, their quality management system.

The core values and concepts in the Malcolm Baldrige National Quality Award Model are another way to define the quality environment. The core values used have been defined, extended and approved over a period of years.

The criteria for evaluating corporate performance excellence are designed to help organizations use an integrated approach to organizational performance management that results in:

- Delivery of ever-improving values to customers, contributing to marketplace success
- Improvement of overall organizational effectiveness and capabilities
- Organizational and personal learning

To a large degree, an organization is built on its core values. Core values help to define, to the Board of Directors, management and employees of the corporation and how they should perform their day-to-day activities.

Core Values and Concepts

The core values and concepts on which the Malcolm Baldrige National Quality Award model is based are:

- Visionary leadership

- Customer-driven excellence
- Organizational and personal learning
- Valuing employees and partners
- Agility
- Focus on the future
- Managing for innovation
- A management by fact
- Social responsibility
- Focus on results and creating value
- Systems perspective

These values and concepts, described below, are embedded beliefs and behaviors found in high-performing organizations. They are the foundation for integrating key business requirements within a results-oriented framework that creates a basis for action and feedback.

Visionary Leadership

An organization's senior leaders should set directions and create a customer focus, clear and visible values, and high expectations. The directions, values, and expectations should balance the needs of all stakeholders. Leaders should ensure the creation of strategies, systems, and methods for achieving performance excellence, stimulating innovation, building knowledge and capabilities, and ensuring organizational sustainability. The values and strategies should help guide all of the organization's activities and decisions. Senior leaders should inspire and motivate the entire workforce and should encourage all employees to contribute, to develop and learn, to be innovative, and to be creative. Senior leaders should be responsible to the organization's governance body for their actions and performance. The governance body should be responsible ultimately to all stakeholders for the ethics, actions, and performance of the organization and its senior leaders.

Senior leaders should serve as role models through their ethical behavior and their personal involvement in planning, communications, coaching, development of future leaders, review of organizational performance, and employee recognition. As role models, they can reinforce ethics, values, and expectations while building leadership, commitment, and initiative throughout the organization.

Customer-Driven Excellence

Quality and performance are judged by an organization's customers. An organization must take into account all product and service features and characteristics and all modes of customer access that contribute value to customers. Such behavior leads to customer acquisition, satisfaction, preference, referral, retention and loyalty, and business expansion. Customer-driven excellence has both current and future components: understanding today's customer desires and anticipating future customer desires and marketplace potential.

Value and satisfaction may be influenced by many factors throughout the customers' overall purchase, ownership, and service experiences. These factors include the organization's relationships with customers that help to build trust, confidence and loyalty.

Customer-driven excellence means much more than reducing defects and errors, meeting specifications, or reducing complaints. Nevertheless, these factors contribute to the customers' view of the organization and thus are important parts of customer-driven excellence. In addition, the organization's success in recovering from defects and mistakes is crucial to retaining customers and building customer relationships.

Customer-driven organizations address not only the product and service characteristics that meet basic customer requirements but also those features and characteristics that differentiate products and services from competing offerings. Such differentiation may be based upon new or modified offerings, combinations of product and service offerings, customization of offerings, multiple access mechanisms, rapid response, or special relationships.

Customer-driven excellence is thus a strategic concept. It is directed toward customer retention and loyalty, market share gain, and growth. It demands constant sensitivity to changing and emerging customer and market requirements and to the factors that drive customer satisfaction and loyalty. It demands listening to the customers. It demands anticipating changes in the marketplace. Therefore, customer-driven excellence demands awareness of developments in technology and competitors' offerings, as well as rapid and flexible response to customer and market changes.

Organizational and Personal Learning

Achieving the highest levels of business performance requires a well-executed approach to organizational and personal learning. Organizational learning includes both continuous improvement of existing approaches and significant change, leading to new goals and approaches. Learning needs to be embedded in the way the organization operates. This means that learning is:

1. A regular part of daily work;
2. Practiced at personal, work unit, and organizational levels;
3. A result of solving problems at their source ("root cause");
4. Focused on building and sharing knowledge throughout the organization; and
5. Driven by opportunities to effect significant, meaningful change.

Sources for learning include employees' ideas, research and development (R&D), customers' input, best practice sharing, and benchmarking.

Organizational learning can result in:

1. Enhancing value to customers through new and improved products and services;
2. Developing new business opportunities;

3. Reducing errors, defects, waste, and related costs;
4. Improving responsiveness and cycle time performance;
5. Increasing productivity and effectiveness in the use of all your resources; and
6. Enhancing the organization's performance in fulfilling its societal responsibilities and its service to the community as a good citizen.

Employees' success depends increasingly on having opportunities for personal learning and on practicing new skills. Organizations invest in employees' personal learning through education, training and other opportunities for continuing growth and development. Such opportunities might include job rotation and increased pay for demonstrated knowledge and skills. On-the-job training offers a cost-effective way to train and to better link training to your organizational needs and priorities. Education and training programs may benefit from advanced technologies, such as computer Internet-based learning and satellite broadcasts.

Personal learning can result in:

1. More satisfied and versatile employees who stay with the organization.
2. Organizational cross-functional learning.
3. The building of the organization's knowledge assets.
4. An improved environment for innovation.

Thus, learning is directed not only toward better products and services but also toward being more responsive, adaptive, innovative, and efficient – giving the organization marketplace sustainability and performance advantages and giving employees satisfaction and motivation to excel.

Valuing Employees and Partners

An organization's success depends increasingly on the diverse backgrounds, knowledge, skills, creativity, and motivation of all its employees and partners.

To value employees means committing to their satisfaction, development, and well-being. Increasingly, this involves more flexible, high-performance work practices tailored to employees with varying workplace and home life needs. Major challenges in the area of valuing employees include:

1. Demonstrating the organizational leaders' commitment to the employees' success,
2. Providing recognition that goes beyond the regular compensation system,
3. Offering development and progression within the organization,
4. Sharing the organization's knowledge so the employees can better serve the customers and contribute to achieving the strategic objectives,

5. Creating an environment that encourages risk taking and innovation, and
6. Creating a supportive environment for a diverse workforce.

Organizations need to build internal and external partnerships to better accomplish overall goals. Internal partnerships might include labor-management cooperation. Partnerships with employees might entail employee development, cross-training or new work organizations, such as high-performance work teams. Internal partnerships also might involve creating network relationships among the work units to improve flexibility, responsiveness, and knowledge sharing.

External partnerships might be with customers, suppliers, and education organizations. Strategic partnerships or alliances are increasingly important kinds of external partnerships. Such partnerships might offer entry into new markets or a basis for new products or services. Partnerships might permit the blending of the organization's core competencies and leadership capabilities with the complementary strengths and capabilities of partners.

Successful internal and external partnerships develop longer-term objectives, thereby creating a basis for mutual investments and respect. Partners should address the key requirements for success, means for regular communication, approaches to evaluating progress, and means for adapting to changing conditions. Joint education and training could offer a cost-effective method for employee development.

Agility

Success in globally competitive markets demands agility – a capacity for rapid change and flexibility. E-business requires and enables more rapid, flexible, and customized responses. Businesses face ever-shorter cycles for the introduction of new/improved products and services, as well as for faster and more flexible responses to customers. Major improvements in response times often require simplification of work units and processes or the ability for rapid changeover from one process to another. Cross-trained and empowered employees are vital assets in such a demanding environment.

A major success factor in meeting competitive challenges in the design-to-introduction (product or service initiation) or innovation cycle time. To meet the demands of rapidly changing global markets, organizations need to carry out stage-to-stage integration (such as concurrent engineering) of activities from research or concept to commercialization.

All aspects of time performance now are more critical, and cycle time has become a key process measure. Time improvements often drive simultaneous improvements in organization, quality, cost, and productivity.

Focus on the Future

In today's competitive environment, creating a sustainable organization requires understanding the short- and longer-term factors that affect the business and marketplace. Pursuit of sustainable growth and market leadership requires a strong future orientation and a willingness to make long-term commitments to key stakeholders – the customers, employees, suppliers and partners, stockholders, the public and the community.

The organization's planning should anticipate many factors, such as customers' expectations, new business and partnering opportunities, employee development and hiring needs, the increasingly global marketplace, technological developments, the evolving E-business environment, changes in customer and market segments, evolving regulatory requirements, community and societal expectations, and strategic moves by competitors. Strategic objectives and resource allocations need to accommodate these influences. A focus on the future includes developing employees and suppliers, doing effective succession planning, creating opportunities for innovation, and anticipating public responsibilities.

Managing for Innovation

Innovation means making meaningful change to improve an organization's products, services, processes, and operations and to create new value for the organization's stakeholders. Innovation should lead the organization to new dimensions of performance. Innovation is no longer strictly the purview of research and development departments; innovation is important for all aspects of the business and all processes. Organizations should be led and managed so that innovation becomes part of the learning culture. Innovation should be integrated into daily work and should be supported by the performance improvement systems.

Innovation builds on the accumulated knowledge of the organization and its employees. Therefore, the ability to rapidly disseminate and capitalize on this knowledge is critical.

Management by Fact

Organizations depend on the measurement and analysis of performance. Such measurements should derive from business needs and strategy, and they should provide critical data and information about key processes, outputs, and results. Many types of data and information are needed for performance management. Performance measurement should include customer, product, and service performance; comparisons of operational, market and competitive performance; supplier, employee, cost and financial performance; and corporate governance and compliance. Data should be segmented by, for example, markets, product lines, and employee groups to facilitate analysis.

Analysis refers to extracting larger meaning from data and information to support evaluation, decision-making and improvement. Analysis entails using data to determine trends, projections, and cause and effect that might not otherwise be evident. Analysis supports a variety of purposes, such as planning, reviewing the overall performance, improving operations, change management, and comparing the performance with competitors' or with "best practices" benchmarks.

A major consideration in performance improvement and change management involves the selection and use of performance measures or indicators.

The measures or indicators selected should represent the factors that lead to improved customer, operational, financial and ethical performance. A comprehensive set of measures or indicators tied to customer and organizational performance requirements represents a clear basis for aligning all processes with the organization's goals.

Measures or indicators themselves should be evaluated and changed to reflect changing conditions.

Social Responsibility

An organization's leaders should stress responsibilities to the public, ethical behavior, and the need to practice good citizenship. Leaders should be role models focusing on business ethics and protection of public health, safety and the environment. Protection of health, safety, and the environment includes the organization's operations, as well as the life cycles of its products and services. Also, organizations should emphasize resource conservation and waste reduction at the source. Planning should anticipate adverse impacts from production, distribution, transportation, use, and disposal of the products. Effective planning should prevent problems, provide for a forthright response if problems occur, and make available information and support needed to maintain public awareness, safety and confidence.

For many organizations, the product design stage is critical from the point of view of public responsibility. Design decisions impact the production processes and often the content of municipal and industrial waste. Effective design strategies should anticipate growing environmental concerns and responsibilities.

Organizations should not only meet all local, state, and federal laws and regulatory requirements, but they should treat these and related requirements as opportunities for improvement "beyond mere compliance." Organizations should stress ethical behavior in all stakeholder transactions and interactions. Highly ethical conduct should be a requirement of, and should be monitored by, the organization's governance body.

Practicing good citizenship refers to leadership and support – within the limits of an organization's resources – of publicly important purposes. Such purposes might include improving education and health care in the community, pursuing environmental excellence, practicing resource conservation, performing community service, improving industry and business practices, and sharing nonproprietary information. Leadership as a corporate citizen also entails influencing other organizations, private and public, the partner for these purposes.

Managing social responsibility requires the use of appropriate measures and leadership responsibility for those measures.

Focus on Results and Creating Value

An organization's performance measurements need to focus on key results. Results should be used to create and balance value for the key stakeholders – customers, employees, stockholders, suppliers and partners, the public, and the community. By creating value for the key stakeholders, the organization builds loyalty and contributes to growing the economy. To meet the sometimes conflicting and changing aims that balancing value implies, organizational strategy should explicitly include key stakeholder requirements. This will help ensure that plans and actions meet differing stakeholder needs and avoid adverse impacts on any stakeholders. The use of a balanced composite performance measure offers an effective means to communicate short- and longer-term priorities, monitor actual performance, and provide a clear basis for improving results.

Systems Perspective

The Baldrige Criteria provide a systems perspective for managing the organization and its key processes to achieve results – performance excellence. The seven Baldrige Categories and the Core Values form the Building Blocks and the Integrating Mechanism for the System.

Management of overall performance requires organization-specific synthesis, alignment, and integration. Synthesis means looking at the organization as a whole and builds upon key business requirements, including the strategic objectives and action plans. Alignment means using the key linkages among requirements given in the Baldrige Categories to ensure consistency of plans, processes, measures, and actions. Integration builds on alignment so that the individual components of the performance management system operate in a fully interconnected manner.

These concepts are depicted in the Baldrige framework. A systems perspective includes the senior leaders' focus on strategic directions and on the customers. It means that the senior leaders monitor, respond to, and manage performance based on the business results. A systems perspective also includes using the measures, indicators, and organizational knowledge to build the key strategies. It means linking these strategies with the key processes and aligning the resources to improve overall performance and satisfy customers.

Thus, a systems perspective means managing the whole organization, as well as its components, to achieve success.

Setting the Proper “Tone” at the Top

The quality environment sets the tone of an organization, influencing the control consciousness of its people to do the right thing at the right time. It is the foundation for all other components of internal control, providing discipline and structure. Quality environmental factors include:

- The integrity, ethical values and competence of the entity's people.
- Management's philosophy and operating style.
- The way management assigns authority and responsibility.
- The way management organizes and develops its people.
- The attention and direction provided by the Board of Directors.

The quality environment is established by executive management. What is important in understanding the quality environment is that the quality environment:

- Has been established by senior management and that senior management is committed to the implementation of that environment throughout the organization; and it
- Defines the standards for the performance of day-to-day activities.

Code of Ethics and Conduct

Employees need to know what is expected from them in the performance of their day-to-day activities. In most corporations, employees are trained in how to perform their job responsibilities; and either has, or is trained in, needed job skills. However, until recently, they were rarely trained in how to react in situations in which ethics and values are involved.

Many employees face situations where ethical conduct guidance is needed. For example, when suppliers invite them to lunch or offer them gifts; when they acquire a second job; when they're dealing with other employees; and when they represent the corporation in outside activities, such as sales and community activities.

The *Code of Conduct* of the corporation represents the manner in which the corporation expects the employees to act. It attempts to define the type of situations that employees may be faced with, and when they are faced with that situation, how they should respond to that situation. A Code of Conduct is one of the most important documents involved in corporate governance.

It is not enough to have a Code of Conduct. That Code of Conduct must be taught and senior officers must live by the Code of Conduct. The Code of Conduct applies to all officers and employees of the organization. However, if a code is to be effective, the senior offices of the corporation must set the example of how to perform.

Open Communications

If the "tone at the top" is to drive corporate governance, then that tone must be communicated to all involved. Communication would not only be to employees, but to partners, agents, suppliers, and other stakeholders involved with the corporation. Communication must not only be downward from senior management, but must include communication upward from the lowest levels to senior management.

Effective communication is planned, not spontaneous. Effective communication is repeatable. Repeatable meaning that if individuals in a specific job are changed, the same types of communication will occur with new individuals.

Corporate governance is a people problem, not a procedural problem. People must first be motivated and convinced that senior management wants the corporate governance practices implemented and enforced. Getting that message to employees and stakeholders is a communication activity.

Communication is inherent in information processing. Communication also takes place in a broader sense, dealing with expectations and responsibilities of individuals and groups. Effective communication must occur down, across and up an organization and with parties external to the organization.

COSO defined these control objectives for communication:

- Effectiveness with which employees' duties and control responsibilities are communicated.
- Establishment of channels of communication for people to report suspected improprieties.
- Receptivity of management to employee suggestions of ways to enhance productivity, quality or other similar improvements.
- Adequacy of communication across the organization (for example, between procurement and production activities) and the completeness and timeliness of information and its sufficiency to enable people to discharge their responsibilities effectively.
- Openness and effectiveness of channels with customers, suppliers and other external parties for communicating information on changing customer needs.
- Extent to which outside parties have been made aware of the entity's ethical standards.

Timely and appropriate follow-up action by management resulting from communications received from customers, vendors, regulators or other external parties.

Guidelines for Effective Communications

The following are some guidelines that quality assurance personnel can use to improve their communication effectiveness.

PROVIDING CONSTRUCTIVE CRITICISM

In giving constructive criticism, you should incorporate the following tactics:

- Do it Privately – Criticism should be given on a one-on-one basis. Only the individual being criticized should be aware that criticism is occurring. It is best done in a private location. Many times it is more effective if it is done in a neutral location, for example, in a conference room or while taking someone to lunch, rather than in the boss' office.
- Have the Facts – General statements of undesired performance are not very helpful. For example, statements such as "That proposal is not clear, fix it" or "Your program does not make best use of the language or technology" leave people feeling confused and helpless. Before criticizing someone's performance, have specific items that are causing the deficiency or undesirable performance.
- Be Prepared to Help the Worker Improve Their Performance – It is not good enough to ask the worker to "fix it." You must be prepared to help fix it. Be prepared to train the subordinate in the area of deficiency. For example, in a proposal, indicate that a return-on-investment calculation was not made; or if a program failed to use the language properly, state specifically how it should and should not be used. You should not leave an individual feeling that they have performed poorly or unsure as to how to correct that performance.

- Be Specific on Expectations – Be sure your subordinate knows exactly what you expect from him or her now and in the future. Your expectations should be as clear as possible so there can be no confusion. Again, in a proposal, indicate that you expect a return-on-investment calculation included in all proposals. Most people will try to do what they are expected to do—if they know what those expectations are.
- Follow a Specific Process in Giving Criticism – The specific process that is recommended is:
 - State the positive first. Before criticizing indicate what you like about their performance. Again, be as specific as possible in the things you like.
 - Indicate the deficiencies with products or services produced by the individual. Never criticize the individual, only the work performed by the individual. For example, never indicate that an individual is disorganized; indicate that a report is disorganized. People can accept criticism of their products and services; they have great difficulty when you attack their personal work ethic.
 - Get agreement that there is a problem. The individual being criticized must agree there is a problem before proper corrective action can be taken. Avoid accepting agreement just because you are the boss; probe the need for improvement with the subordinate until you actually feel there is agreement that improvement can be achieved. For example, if you believe a report or program is disorganized, get agreement from the individual on specifically why it might be disorganized.
 - Ask the subordinate for advice on how to improve their performance. Always try to get the employee to propose what needs to be done. If the employee's suggestion is consistent with what you have decided is a realistic method of improvement; you have finished the process.
 - If the subordinate is unable to solve the problem, suggest the course of action that you had determined before performing the actual criticism.
 - Make a specific "contract" regarding what will happen after the session. Be very specific in what you expect, when and where you expect it. If the employee is uncertain how to do it, the "contract" should include your participation, as a vehicle to ensure what will happen.

One last recommendation for criticism:

- Avoid making threats about what will happen if the performance does not change. This will not cause any positive behavior change to occur and normally produces negative behavior. Leave the individual with the assumption that he or she has the capability for improvement, and that you know he or she will improve.

ACHIEVING EFFECTIVE LISTENING

Throughout school, students are taught the importance of speaking, reading, writing, and arithmetic, but rarely is much emphasis placed on listening. The shift in society from industrial

production to information management emphasizes the need for good listening skills. This is particularly true in the practice of software testing – oral communication is rated as the number-one skill for the quality analyst.

Some facts about listening include:

- Many Fortune 500 companies complain about their workers' listening skills.
- Listening is the first language skill that we develop as children; however, it is rarely taught as a skill. Thus, in learning to listen, we may pick up bad habits.
- Listening is the most frequently used form of communication.
- Listening is the major vehicle for learning in the classroom.
- Salespeople often lose sales because they believe talking is more important than listening (thus, in ads a computer company emphasizes that they listen).

It is also important to understand why people do not listen. People do not listen for one or more of the following reasons:

- They are impatient and have other stimuli to respond to, such as random thoughts going through their mind.
- They are too busy rehearsing what they will say next, in response to someone.
- They are self-conscious about their communication ability.
- External stimuli, for example, an airplane flying overhead, diverts their attention.
- They lack the motivation and responsibility required of a good listener.
- The speaker's topic is not of interest to them.

The listener must be aware of these detriments to good listening so they can recognize them and devote extra attention to listening.

THE 3-STEP LISTENING PROCESS

The listening process involves three separate steps: 1) hearing the speaker, 2) attending to the speaker, and 3) understanding the speaker. The practice of listening requires these three listening steps to occur concurrently. Mastering each of these steps will help improve your listening abilities.

Step 1: Hearing the Speaker

Hearing the speaker requires an understanding of the five channels of communication incorporated into speech. Much of listening occurs beyond merely hearing the words. Let's look at the five channels through which a speaker delivers information to his/her audience:

Information Channel	The speaker's subject.
Verbal Channel	The words used by the speaker.
Vocal Channel	The tone of voice associated with the various words.

Body Channel	The body movements and gestures associated with the information being conveyed.
Graphic Channel	The pictures, charts, etc., that the speaker uses to emphasize or illustrate the material being discussed.

Speakers normally use the information, verbal, vocal, and body channels in speaking. In some instances, they also use the graphic channel. Listening requires that there is a meeting of the mind on the information channel. Speakers sometimes skip around to different subjects, making it easy to lose the subject being covered on the information channel. In Step 2, attending to the speaker, we will discuss the importance of feedback to confirm the subject being covered on the information channel.

The vocal and body channels impact the importance of the verbal channel. The verbal channel includes the choice of words used to present information, but the vocal and body channels modify or emphasize the importance of those words. For example, the words in the verbal channel may be, "John says he can do it." However, the tone of the vocal channel might indicate that John cannot do it, or the use of a thumbs-down body channel signal will also indicate that John cannot do it.

Hearing the speaker involves an awareness of all five channels, and listening to and watching the speaker to be sure we are receiving what the speaker is saying through all five channels. To master the hearing step, you must pay attention to all five channels. If you miss one or more of the channels, you will not hear what the person is saying. For example, if you are only paying partial attention to the speaker when the words, "John can do it" are stated, you may hear that John can do it, while the speaker said that John could not do it.

Step 2: Attending to the Speaker

Attending to the speaker is sometimes referred to as being an active listener. Devote your full attention to the speaker to confirm that what you heard is what the speaker intended you to hear. You must first understand yourself and your situation. You must evaluate your motivation for wanting to listen to this speaker. If the subject is important to you, but the speaker is boring, it will require significantly more effort on your part to be a good listener.

The most important part of attending to the speaker is establishing an active listening ability. Active listening involves a lot of response and dynamics. Some people view the listening process as a passive skill where you sit back and let the other person talk. This is fine for hearing the speaker, but not for confirming what the speaker has said. Feedback is very important to the listening process, particularly in this step. Feedback can be a nonverbal response, such as nodding your head, or a verbal response such as a question or a statement of confirmation.

It is very important to send the right type of feedback to the speaker. The wrong type of feedback not only doesn't confirm what the speaker said, but also can reduce or terminate the listening process. It is very irritating to a speaker who is providing information to have the listener stray from the subject. For example, the speaker might be describing a quality problem, and the listener changes the subject and asks where the speaker is going to have lunch that day.

Some suggestions to help in attending to the speaker are:

- Free your mind of all other thoughts and concentrate exclusively on the speaker's communication.
- Maintain eye contact with the speaker for approximately 80 percent of the time.
- Provide continuous feedback to the speaker.
- Periodically restate what you heard the speaker say, and ask the speaker to confirm the intent of the information spoken.
- Move periodically to the understanding step to ensure that the information passed has been adequately understood.

Step 3 - Understanding the Speaker

There are five types of listening. While people can listen several different ways concurrently, normally listening is limited to one of the five types. The type chosen will have an impact on the ability to understand what the speaker is saying. When one has deciphered the information channel (i.e., what the subject is) and related the importance of that subject to the audience, listening must be adjusted to ensure that we get the message we need.

The five types of listening and their impact on understanding are:

Type 1: Discriminative Listening

Directed at selecting specific pieces of information and not the entire communication. For example, one may be listening to determine if an individual did a specific step in the performance of a task. To get this, listen more to the nonverbal expressions rather than the verbal channel.

Type 2: Comprehensive Listening

Designed to get a complete message with minimal distortion. This type of listening requires a lot of feedback and summarization to fully understand what the speaker is communicating. This type of listening is normally done in fact gathering.

Type 3: Therapeutic Listening

The listener is sympathetic to the speaker's point of view. During this type of listening, the listener will show a lot of empathy for the speaker's situation. It is very helpful to use this type of listening when you want to gain the speaker's confidence and understand the reasons why a particular act was performed or event occurred, as opposed to comprehensive listening where you want to find out what has happened.

Type 4: Critical Listening

The listener is performing an analysis of what the speaker said. This is most important when it is felt that the speaker is not in complete control of the situation, or does not know the complete facts of a situation. Thus, the audience uses this type of

understanding to piece together what the speaker is saying with what has been learned from other speakers or other investigation.

Type 5: Appreciative or Enjoyment Listening

One automatically switches to this type of listening when it is perceived as a funny situation or an explanatory example will be given of a situation. This listening type helps understand real-world situations.

One must establish which type of understanding is wanted and then listen from that perspective.

Implementing a Mission, Vision, Goals, Values, and Quality Policy

The mission statement tells why a company or an organization exists. Organizations need to map their course of direction, which is the corporate vision. Goals convey how the vision will be achieved. Values are like an organization's code of ethics – they help establish the corporate culture and shape the foundation for making decisions. A quality policy is a statement of principles, and a broad guide to action.

The statements of mission, vision, goals, values, and quality policy must be what all levels of management truly believe and practice in their day-to-day activities. Developing the statements cannot be delegated, nor is it a quick task. Management may begin by:

- Using industry examples and models of these statements
- Understanding the current culture and beliefs in the organization
- Establishing an action plan to develop these statements that includes: roles and responsibilities of involved parties, specific tasks and dates to accomplish the tasks, and a method of reporting status of action plans.

Mission

A mission statement explains why a company, organization, or activity exists, and what it is designed to accomplish. It clearly and concisely describes the work that is done, providing direction and a sense of purpose. The mission should focus on products and services and be customer-oriented. During implementation, the mission is constrained by the vision and values.

Examples of mission statements include:

- From Arco Transportation Company Information Services: "The mission of information services is to provide the appropriate computing network, products, and services and support of the strategies, goals, and objectives of the company."
- From the Ford Motor Company (listed in their Ford Q-101 Quality Systems Standard, January 1986): "Ford Motor Company is a worldwide leader in automotive and automotive-related products and services as well as in newer industries such as aerospace, communications, and financial services. Our mission is to improve continually our

products and services to meet our customers' needs, allowing us to prosper as a business and to provide a reasonable return for our stockholders, the owners of our business.”

Vision

Leaders provide a vision, which is a clear definition of the result to be achieved. Organizations without a vision flounder. The vision establishes where the organization desires to move from its current state. It gives everyone a direction to work towards. Senior management should establish the vision, ensuring how it contributes to the business is clear. A vision is simple and concise, and it should be understood and supported by all.

Examples of visions include:

- From the Quality Assurance Institute: “Our vision is to produce competent and successful quality assurance analysts.”
- From the Eastman Kodak Company: "We see ourselves now and in the future as a company with a strong customer franchise, known for reliability, trust, and integrity in all relationships. Our business will be based on technologies that have evolved over our long history, and which will give us unique advantages over our competition. These technologies will span our core businesses, and will also go beyond boundaries we can see today.”
- From the Ford Motor Company: "A worldwide leader in automotive and automotive-related products and services as well as in newer industries such as aerospace, communications, and financial services."
- President Kennedy had a vision of putting a man on the moon before 1970.
- QA analysts should have a vision of improving quality, productivity, and customer satisfaction.

Goals

Goals explain how the vision will be achieved. For example, if an organization's vision is to produce defect-free software, a goal might be to have no more than one defect per thousand lines of code. Goals change as an organization moves closer to accomplishing the vision. Well-developed programs are necessary to achieve the goals.

Goals and objectives are often used interchangeably; however, goals tend to be more global and nonquantitative. Objectives come from goals, and tend to be more specific and quantitative.

Goals:

- Are consistent with the vision
- Are established by operational management (manager of systems and programming, manager of computer operations, etc.)
- Must have management commitment
- Clearly identify the role each individual plays in accomplishing the goal

- Should be linked to programs established to accomplish the goals

Strategic quality management goals must focus on both the producer and the customer. Short-term goals should:

- Reduce defects
- Reduce cycle time (i.e., shorter schedule and less resources)
- Provide return on investment from short-term programs

Long-term goals should be customer oriented. They involve improving customer satisfaction and greater matching of the products and services to the true customer needs. These goals could include, but should not be limited to:

- High customer satisfaction activities
- Management establishing a need for quality, thus creating an environment receptive to quality processes
- Understanding what must be done in order to deploy a new process
- Improving compliance to processes
- Sustaining quality effort, as a result of managing quality
- Involving all employees in quality processes
- Recognizing the need for the quality analyst
- Establishing a quality infrastructure with adequate resources to perform the assigned mission
- Having adequate resources to perform quality activities such as continuous process improvement
- A doable and measurable plan of action enabling the quality processes to demonstrate accomplishments based on approved objectives

Financial goal statements from the Eastman Kodak Company are:

- “To rank among the top 25 U.S.-based multinational companies in net earnings”
- “To approach a return on equity of 20 percent”
- “To increase worldwide productivity at double the U.S. average for manufacturing companies”

Values

Values or guiding principles tell how to conduct business. They help define an organization’s culture and personality by clarifying what behavior is expected in order for the organization to achieve its vision and mission. Values are established by senior management and respect the integrity of the individual. Examples of values are: customer-focused, quality management, innovative, employee empowerment, ethical, cooperative relationships, and risk-taking. Values should be consistent with Dr. Deming’s 14 quality principles (see Skill Category 1), and they

should be integrated into the organization's work program. If really believed, values help focus the organization on a shared behavioral model.

Examples of values include:

- From the Eastman Kodak Company:

“Quality – to strive for continuous improvement through personal contributions and teamwork.”

“Integrity – requiring honest relationships with colleagues, customers, shareholders, and suppliers.”

“Trust – characterized by treating everyone with respect and dignity.”

“Ethical behavior – so Kodak can earn and deserve a reputation that is beyond question.”

“Teamwork – through open communication that gives everyone a sense of personal involvement in the company's performance.”

“Job satisfaction – in an environment that encourages people to grow to their full potential.”

“Creativity – fostered by an atmosphere that challenges employees to seek new solutions and to take intelligent risks.”

“Flexibility – recognizing the need to anticipate and respond to changing economic, social, competitive, and market conditions.”

“Winning attitude – in knowing that through hard work, pride and confidence, Kodak people make up a ‘world-class’ team.”

- From the Ford Motor Company (listed in their Ford Q-101 Quality Systems Standard, January 1986) include:

“People – Our people are the source of our strength. They provide our corporate intelligence and determine our reputation and vitality. Involvement and teamwork are our core human values.”

“Products – Our products are the end result of our efforts and they should be the best in serving customers worldwide. As our products are viewed, so are we viewed.”

“Profits – Profits are the ultimate measure of how efficiently we provide customers with the best products for their needs. Profits are required to survive and grow.”

- From Arco Transportation Company Information Services relating to people: "Information services maintain a productive and challenging work environment to foster personal growth and career development."

Quality Policy

Executive management's commitment to quality should be expressed in writing to all employees in the form of a quality policy. Management should work as a team to develop the policy, which must be aimed at the employees and written so they can understand it. The policy should be concise and cover all aspects of quality. Eventually, every existing regulation, procedure, and policy letter should be reviewed to assure that it aligns with the new quality policy.

Examples of quality policies are:

- From Xerox: “Quality is the basic business principle of Xerox. Quality means providing our internal and external customers with innovative products and services that fully satisfy their requirements. Quality improvement is the job of every Xerox employee.”
- From Corning Glass Works: “It is the policy of Corning Glass Works to achieve total quality performance in meeting the requirements of external and internal customers. Total quality performance means understanding who the customer is, what the requirements are, and meeting those requirements without error, on time, every time.”
- From Baxter: “We will reach agreement on requirements with our customers and suppliers, inside and outside the company. We will conform to those requirements and perform defect-free work at all times.”
- Key components from IBM Corporation’s quality policy are:
 - “Quality is the cornerstone of the IBM Corporation business.”
 - “The objective of this policy is to provide products and services, which are defect free.”
 - “Everyone must learn to do his or her job right the first time (i.e., no rework due to defects).”
 - “Each stage of a job must be defect free.”
 - “Quality is everybody's responsibility.”

Monitoring Compliance to Organizational Policies and Procedures

Monitoring comprises those activities, which are undertaken to ensure the corporate governance is performed in the manner desired by the Board of Directors and executive management. Monitoring can be performed by the individual doing the job, by the organization responsible for the job, and by independent organizations that monitor the organization’s activities.

In monitoring, there are both preventive and detective controls. Preventive controls prevent an event from occurring; detective controls uncover an undesirable event once it has occurred. Corrective controls are part of the enforcement process. However, as will be discussed in the next section, both preventive and detective controls are ineffective unless corrective controls are in place and working.

Preventive monitoring focuses on the input or entrance criteria to a business process. For example, if a business pre-approved customers before they could buy products, then there would be a monitoring process in place to determine that only orders from authorized customers were processed.

Detective monitoring occurs during or after the business process. For example, during processing, the business process would determine that as customer orders are processed, that those orders are from authorized customers. Note that if the preventive controls are effective, detective controls can be reduced. However, sometimes, preventive controls cannot detect an unfavorable event. For example, an authorized customer may place an order, which during processing will be determined to exceed the customer's credit limit. Thus, the preventive monitoring would not be effective, but the detective monitoring would be effective.

Four Types of Monitoring

Monitoring is a very important component of corporate governance. It encompasses a wide range of activities. For example, monitoring in manufacturing is normally referred to as quality control. Independent monitoring is frequently called *auditing*. Monitoring by clerical and professional staff is sometimes called desk checking. Organizations rarely refer to monitoring as the combination of the totality of these activities. Rather, monitoring is a classification that can be used in corporate governance to encompass all of the activities executed by or under the direction of the organization to ensure that “the right job is performed correctly.”

For the purpose of this course, monitoring consists of these four activities:

- **Monitoring the tone at the top**
Assuring that the proper leadership is in place for effective corporate governance
- **Monitoring by individuals**
Assuring that individuals assigned to do work do the work in accordance with the procedures developed for doing that work
- **Ongoing monitoring**
Assurance by the organization responsible for work that the work is performed as specified
- **Independent monitoring**
The use of auditors, both internal and external, to provide an independent assessment of the adequacy of the system of internal controls and compliance to those controls.

Monitoring the Tone at the Top

The “tone at the top” implies that the appropriate message is sent to employees by senior management, and that senior management is, in fact, “walking the talk.” Walking the talk means that senior management is doing those things that they are asking the organization's employees to

do. For example, if there's a Code of Conduct, senior management is the role model for complying with that Code of Conduct. Employees frequently need a role model to determine how they should follow the governance rules of the organization.

Monitoring the tone at the top is segregated from the other three monitoring activities because experience has shown the other three will be ineffective without the appropriate tone at the top. For example, if the CEO wants employees involved in community activities, yet does not personally become involved in community activities, that sends the inappropriate message to the employees.

The employees of the organization all monitor the tone at the top. They continually look to how their management acts to determine the employee behavior that will be accepted by management. For example, if the Code of Conduct indicates that employees are not to accept gifts from suppliers, but management accepts gifts from suppliers, then all employees will assume that it is appropriate to accept gifts from suppliers.

Auditors should, but may not, monitor the tone at the top. Under auditing standards, both external and internal, there's no obligation to monitor the tone at the top. However, based on today's business climate, and the provisions of the Sarbanes-Oxley Act, auditors should evaluate this aspect of management behavior.

In monitoring the tone at the top, the following are reasonable control guidelines:

- Senior management effectively communicates the governance standards and guidelines to all employees of the organization.
- Senior management "walks the talk" regarding the communicated "tone" of corporate governance standards and guidelines from senior management.

Monitoring by Individuals

There are two components of every job. One component encompasses the procedures to do the job, and the second component involves procedures to check if the job was done correctly. Both aspects of every job should be specified. Each worker should be provided these two types of procedures.

Ideally, the check procedure is incorporated into the work activities so that it becomes a part of doing a job. For example, the bank teller returning cash to a customer might withdraw the cash from the drawer and count to see that it was the correct amount during the draw. Then to check that the amount is correct, the teller may deliver the cash to the customer and count it out loud at the same time. The withdrawal of the cash from the cash drawer would be the procedure to perform the withdrawal, while the counting in front of the customer would be the check procedure.

The COSO internal control standards include the following on individual monitoring. COSO defines individual monitoring as:

Extent to which personnel, in carrying out their regular activities, obtain evidence as to whether the system of internal control continues to function.

Ongoing Monitoring

Ongoing monitoring is the total activities of the organizational unit responsible for the work being monitored. It includes both individual monitoring and monitoring within the organization responsible for the work. Individual monitoring has been addressed separately because it is the exclusive responsibility of the individual doing the job. Ongoing monitoring will then include all monitoring activities except that performed by the individual.

In computerized organizations, monitoring may be performed automatically. In other words, the computer system may do work and then include procedures to check the work being performed. For example, there may be checks to ensure that a customer ordering a product who enters a customer number is then checked through the computer file of customers to verify that the customer name and the customer number match.

Monitoring within the organization may include such things as:

- Supervisor monitoring – a supervisor monitors the work performed by a subordinate.
- Peer monitoring – employees may monitor the work of other employees.
- A group assigned monitoring – responsibilities will monitor the work of employees. For example, someone solely responsible for monitoring may check all of the work performed by all of the employees or monitoring may be done on a random basis.

The COSO internal control standards provide monitoring control objectives. These control objectives are:

- Extent to which communication from external parties corroborate internally generated information, or indicate problems.
- Periodic comparison of amounts recorded by the accounting system with physical assets.

Independent Monitoring

Independent monitoring are those monitoring activities, which are performed by individuals not employed by the organizational unit responsible for the work. Independent monitoring can be performed by:

- An organizational unit established for the purpose of monitoring
For example, in health care units, an “HIPAA compliance group” might be established to monitor compliance to HIPAA requirements.
- Internal auditors
An audit activity within the organization who is independent from the organization they monitor or audit, and generally independent from the management over the activity being monitored.

- External auditors

Auditors engaged by the Board of Directors to evaluate the adequacy of the organization's system of internal control and other activities governed by auditing standards.

The extensiveness of independent monitoring is normally determined by the adequacy of the organization's system of internal controls. If those performing independent monitoring believe that the system of internal control is effective, the extent of their monitoring will decrease. On the other hand, if there's significant weakness in the system of internal control, monitoring will be increased.

The COSO internal control framework provides control objectives for independent monitoring. These control objectives are:

- Extent to which training seminars, planning sessions and other meetings provide feedback to management on whether controls operate effectively.
- Whether personnel are asked periodically to state whether they understand and comply with the entity's code of conduct and regularly perform critical control activities.
- Effectiveness of internal audit activities.
- Scope and frequency of separate evaluations of the internal control system.
- Appropriateness of the evaluation process.

Enforcement of Organizational Policies and Procedures

Enforcement, as discussed in this lesson, means the decision whether or not to enforce. This is done to separate the enforcement decision from the enforcement action. In some instances, the decision and action will be performed by the same individual; in other instances, it may be performed by other individuals or other organizational units.

Enforcement may be an *objective enforcement decision* or a *subjective enforcement decision*. An objective enforcement decision means that if there is a violation, the decision has been pre-determined. For example, if an employee is performing work for a competitor, the decision may always be to terminate that employee. If an employee has stolen money or property from the organization, the decision may always be to legally prosecute that individual and take action to be compensated for the loss.

A subjective enforcement decision is when someone will analyze the situation and then make a decision on enforcement. The decision may be among different types of enforcement, or whatever the individual believes is an effective enforcement for the organization. For example, enforcement may vary based on the magnitude of the variance from the standard, as well as whether it is a first time variance or multiple variances by the same employee. The enforcement decision should be based on the following:

- Are all the facts known?
- Has the defendant been given the opportunity to explain his/her position?
- Is the enforcement legal?
- Is the enforcement in the best interest of the organization?
- Is enforcement consistent with the “tone at the top” message from executive management?
- Is the enforcement consistent with previous variances of the same type and magnitude?

Types of Enforcements

There are three types of enforcement actions. These are: automated enforcement, self-enforcement, and supervisory enforcement. Each of those is discussed below.

Automated Enforcement

This is normally viewed as the most fair of all methods of enforcement. Automated enforcement is normally performed by a computer. The rules are included in the computer system, and when those rules are violated, enforcement is automatic.

Automated enforcement is a predetermined enforcement decision. The computer merely executes the action. For example, if you need a password to enter into a system and you do not enter a correct password, access is denied. If you are charging items on your credit card and you have exceeded your credit limit, the computer action is to disallow the purchase.

Self-Enforcement

The next best type of enforcement is self-enforcement. In the game of golf, professional golfers are required to self-enforce the rules of golf. In other words, if a golfer knows that he/she has violated a rule of golf, that golfer must self assess a penalty on himself/herself.

Organizations that can create an environment of self-enforcement become role models in corporate governance. Also, individuals who self-enforce, know the rules, believe the rules and are willing to accept the punishment associated with variance from the rules even if that variance is unintentional.

There are five steps in self-enforcement as follows:

1. Hear the compliance standards and guidelines.

An individual cannot self-enforce the rules if that individual has not heard the rules.

2. Understand the governance standards and guidelines.

It is not enough just to hear the rules. For example, reading published rules may not provide management’s intent. Understanding is usually associated with training. The

individual must be given enough instruction so that they understand both the letter of the rules and the intent of the rules.

3. Believe the governance standards and guidelines are good standards and guidelines.

If an individual does not believe that the rules are appropriate, the individual will not self-enforce. For example, if you were driving on an open road in the country with a 25 mph speed limit with no other vehicle or building in sight, you might not believe that 25 mph is a fair speed limit. Given this, the individual might be tempted to exceed the stated speed limit, thus violating the speed standards.

4. Remember the governance standards and guidelines.

If in a situation in which the governance standards and guidelines apply, the individual does not remember the standards and guidelines, that individual cannot self-enforce. Remembering may be done by providing checklists, pocket cards and so forth, which indicate what the rules are.

5. Do, in fact, execute self-enforcement procedures.

If the individual has heard the rules, understands the rules, believes the rules and remembers the rules, that individual will most likely enforce the rules.

Supervisory Enforcement

Supervisory enforcement is when someone other than the individual enforces compliance to a policy, procedure or standard. The individual can be a peer, a quality control person, the individual's supervisor, or member of management or their representative.

When a peer enforces a rule, it is normally part of a review process. For example, if a programmer is inspecting another programmers' work, and a coding standard has been violated, the inspector will not provide a clean inspection opinion until the rule has been corrected.

Quality control people are individuals assigned responsibility to enforce policies, procedures, or standards. For example, a security guard may not grant you access to a building unless you have the appropriate pass. The bitter librarian will not enter an item into the library until it meets the library standards. Software testers will not indicate a system is ready for operation if it has known defects. When management becomes aware of, or is informed of, a violation of policies, procedures, or standards, then they personally enforce compliance.

The enforcement decision is normally worthless unless enforcement action is taken on a regular basis. For example, warning messages by a compiler are an enforcement decision, but if no action is required, the enforcement decision can be ignored.

Enforcement is an important component of the quality environment. It is the means by which management communicates the way in which work must be done. If management fails to enforce the "rules," employees quickly recognize that the rules are unimportant.

Management communicates, in many ways, it is not necessary to follow the rules. For example, if management rewards project personnel on completing projects within schedule and budget

constraints, even though they failed to meet IT work standards, management communicates meeting the standards is unimportant.

Quality Baselines

Organizations need to establish baselines of performance for quality, productivity and customer satisfaction. These baselines are used to document current performance and document improvements by showing changes from a baseline. In order to establish a baseline, a model and/or goal must be established for use in measuring against, to determine the baseline.

<i>Quality Baseline Concepts</i>	<i>125</i>
<i>Methods Used for Establishing Baselines</i>	<i>130</i>
<i>Model and Assessment Fundamentals</i>	<i>136</i>
<i>Industry Quality Models</i>	<i>139</i>

Quality Baseline Concepts

Baselines Defined

Developing a baseline is performing an analysis/study to determine the current level of performance in a specific activity. Baselines normally are quantitative and describe multiple attributes of an activity/process. For example, if one were to baseline the software development process the baseline could include quantitative data on defect rates, resources expended by phase, productivity rates such as function points per person-month, and levels or amounts of documentation (number of words of documentation per line of code). The metrics used for baselining should be metrics, which, if improved, would help drive the management-defined results.

A baseline needs to be established for two reasons: first, to identify perceived quality problems; and second, to establish a baseline from which quality improvement objectives can be established and improvements in quality quantitatively measured. The practices used to improve the process are covered in Skill Category 6.

These quantitative quality baseline studies help establish the need for a quality initiative. Many IT managers do not know the severity of their quality problem. Once a need can be identified from analyzing a baseline, the need for a quality initiative to address the problem normally becomes obvious.

Types of Baselines

A baseline study is one designed to quantitatively show the current status of an activity, program, or attitude. It both shows “how you are doing” and provides a quantitative base from which change can be measured.

The studies included in this manual should be viewed as baseline studies. They are attempting to establish a “base” from which a quality improvement need can be identified and measured. Baseline studies can be conducted in one of the following two manners:

- Evaluate entire population

This means all of the parties or products involved will be surveyed. This method is normally most effective when the information to be analyzed is automated. For example, when looking at factors such as schedule and budget.

- Sample survey

Using this method, only a part of the population of people/products are surveyed. This approach is normally most effective when there is a large population to be surveyed and the data is not automated. While sampling should be done statistically, it is not essential in these studies for valid statistical samples to be drawn. The reason is that quality is attempting to eradicate all defects, and even if the defect is only perceived by a part of the population, it is still a defect and warrants attention.

Conducting Baseline Studies

The three logical groups to conduct baseline studies are:

- Quality assurance groups

If one has been established, they should conduct baseline studies in areas of quality concern.

- Quality task forces

A special task force or committee established to study quality/productivity in an information services function. In many instances these study groups precede the establishment of a quality assurance function. In fact, many of the individuals who chair the study group later become the quality assurance manager.

- IT management

Specific managers that have a concern over quality may wish to perform a baseline study. Note that in many instances the study is actually conducted by a subordinate to that manager.

Baseline studies need not be time-consuming or costly. The objective is to identify quantitatively potential problems. The quantitative data should be subject to analysis and interpretation. If the data appears unreliable, then an additional study might be undertaken to validate the reliability. It is generally good practice to get the data as quickly and cheaply as possible, because in most instances, the data is used to substantiate what people intuitively know.

The following are typical steps needed to perform a baseline study.

1. Identify products/services to be surveyed

This is the requirements phase of the baseline study. Studies should be directed at specific products or services. For example, computer programs as a product, and customer/user interaction as a service.

2. Define conformance and nonconformance

The individuals developing the survey instrument must have defined (at least on a preliminary basis) the expected conformance and nonconformance. Note that in many instances the survey will be used to help establish nonconformance, but the general areas of nonconformance will need to be identified in order to gather nonconformance data.

3. Identify survey population

The population of data/people to be surveyed needs to be identified. This is a critical step because the definition of nonconformance will vary significantly depending upon who is considered the population. For example, programming defects would look significantly different to the programmer than the end user of the program results. The programmer may only consider variance from specs a defect, but to the end user not meeting needs is a defect.

4. Identify size of population to be surveyed

This step is one that involves economics. It is always cheaper to look at fewer than more. The question that needs to be decided is how few can be examined to give valid results. Statistically, we rarely try to go below a sample size of twenty, but in surveying people we may be able to drop below this limit and still get valid results.

5. Develop survey instrument

A specific survey instrument must be developed to meet the survey objectives. Surveys need to be customized to the specific needs and vocabulary of the population being surveyed.

6. Conduct survey

The survey instruments should be completed by the surveyed population. Form a quality perspective; it is helpful to train the population on how to complete the survey questionnaire. This can be done through written instructions, but it is normally better to do it verbally. If the population group is small, they can be called together for a meeting, or have the survey

instruments hand delivered and explained. Generally, the validity of the results will increase when extra time is spent to explain the intent and purpose of the survey.

7. Follow up on incomplete surveys

The survey should have a time limit for response. Normally this should not exceed one week. If the surveys have not been returned by that time, then the surveying group should follow up and attempt to get as many surveys completed as possible. Note that it is generally not realistic to expect every survey to be returned.

8. Accumulate and present survey results

The survey information should be accumulated, consolidated, and put into presentation format. Suggestions on how to do this follow in a later part of this quality initiative.

9. Take action and notify participants of that action

All surveys should result in some specific action. Even if that action is to do nothing, a decision should be made based on the survey results. That decision should be given to the survey participants. Note that whenever a survey is conducted, the participants expect some action. Not to inform the participants of the action will reduce cooperation in future surveys.

Conducting Objective Baseline Studies

Objective baseline studies are ones which are viewed as factual and nonargumentative. There are very few objective studies that can be conducted within information services. Objective means performed by counting, and what can be counted must be considered nonargumentative. For example, if we wanted to know how many lines of executable code were in a program, and let us assume we can define what is meant by a line of executable code, then we could count those lines and have an objective baseline.

We need to note that what may appear as objective, may really be more subjective. For example, if we ask people to keep time records, and then record hours worked based on those times, we must make the assumption that the count is accurate. In most instances, the hours count is subjective because many will record their hours worked at the end of each month, and thus the hours count is a subjective measure and not an objective measure.

Objective measures are those, which can be accomplished by counting. Examples of objective baseline measures that can be used for baselines include:

- Project completed on schedule
- Lines of code
- Number of programs
- Number of people assigned to a project
- Number of abnormal terminations

Again, the exactness of the counting will actually determine whether the above measures are objective or subjective. It is important to recognize that there are very few objective measures, and thus we are forced to use subjective measures in measuring quality and productivity.

Conducting Subjective Baseline Studies

Subjective baseline studies will be the most commonly conducted studies in measuring quality and productivity. Subjective means that judgment is applied in making the measure. We noted in the discussion on objective measures that when the individual involved in recording time has the option of applying judgment, then the measure becomes subjective.

Baselines should be quantitative even if it is a subjective measure, but quantitatively subjective. For example, because quality conformance and nonconformance must be defined by people, we are looking for ways to put this information into a quantifiable format. This does not convey a lot of information, but it is indicative of a problem. However, if we develop a five-point scale for unresponsiveness, and ask your dissatisfied customer to complete that scale, we now have conveyed a lot more information. If our scale rates “1” as very poor service, and “5” as very good service, there is a great deal of difference between a “1” rating and a “3” rating for dissatisfaction.

Examples of products/services that can be measured subjectively for developing a baseline include:

- Customer satisfaction
- Effectiveness of standards/manuals
- Helpfulness of methodologies to solve problems
- Areas/activities causing the greatest impediments to quality/productivity
- Causes for missed schedules/over-budget conditions
- Understandability of training materials
- Value of tools
- Importance of activities/standards/methods/tools to individual activity

Baselines can be conducted for any one of the following three purposes:

Planning

To determine where detailed investigation/survey should be undertaken.

Internal analysis

To identify problems/areas for quality improvement. Once the problem/area has been identified, then no additional effort need be undertaken to formalize the results.

Benchmarking

Comparison against external organizations.

If you want to develop a baseline of customer satisfaction, you might ask your customer to rate the following factors using “very satisfied” to “very dissatisfied” as a scale:

- **Availability** – Accessible to the customer, as agreed (e.g., product is available when you need it).
- **Correctness** – Accurate and meets business and operational requirements (e.g., automated information is accurate).
- **Flexibility** – Easy to customize for specific needs (e.g., easy to enhance product with new features).
- **Usability** – Easy to learn, easy to use, and relevant (e.g., the way you interact with components of the product is consistent).
- **Caring** – Demonstrating empathy, courtesy, and commitment (e.g., project team is sensitive to your needs).
- **Competence** – Having and applying the appropriate knowledge and skills (e.g., project team provides correct answers to questions).
- **Dependability** – Meeting cost and schedule commitments (e.g., project team successfully completes work on schedule).
- **Responsiveness** – Providing prompt turnaround of customer inquiries and requests (e.g., project team responds quickly to your written requests).

Next, to develop a customer satisfaction score you would need to assign importance to each factor. For example, you might assign the percentage of importance to your end user/customer of these eight quality factors as follows: 10% availability, 10% correctness, 5% flexibility, 10% usability, 10% caring, 20% competence, 20% dependability, and 15% responsiveness.

Methods Used for Establishing Baselines

IT organizations have established many different baselines to evaluate current performance and to measure improvement. To help understand the type of baselines that are used in IT, QAI has categorized four most commonly used baselines as listed below. Each one of these will be discussed individually.

- Customer surveys
- Benchmarking
- Management established criteria
- Industry models

Customer Surveys

The customer needs to be defined as some group or individual receiving IT products and services. These can be internal customers, such as programmers receiving program specifications from

systems analysts, or external customers, such as the payroll department using IT products and services to produce payroll.

Skill Category 1 provides two definitions of quality. These were “meets requirements” and “fit for use.” Customer surveys use the “fit for use” definition of quality. Customer surveys are *subjective baselines*. Customer surveys measure attitude and satisfaction of customers with the products and services they receive. Because they are subjective it is important that customer surveys are properly constructed.

We can divide customer surveys into report cards and surveys. Report cards ask the customer what they think about something, for example, were they satisfied with the user manuals provided by IT. The report card will ask them on a scale of 1-to-5, one meaning very pleased and five meaning very displeased, on what they think of the user manual. Because the report card is not controlled, it is sometimes difficult to know how the user calculated a particular rating.

Surveys should be constructed around specific factors and attributes of a product or service. Questions are then constructed to support the assessment of that factor and attribute. The subjective response is defined in enough detail so that there is consistency among individuals regarding the response. The factors and attributes are then given rates so that a total customer-satisfaction score can be developed.

Benchmarking to Establish a Baseline Goal

A benchmarking baseline is one determined by another organization. For example, if you wanted to establish a baseline goal for up-time in your computer center, you might want to go to what you believe is a leading organization and use their up-time percentage as your baseline goal.

What is important in benchmarking is that you have a well established baseline measurement in your organization so that you can ensure the other organization’s baseline is compliable. Let us assume that you are baselining the number of defects per thousand lines of code created during development. Through your baseline of your organization you will have to carefully define what a defect is and what a line of code is. Once you have done that and you look at another organization’s defects per thousand lines of code for benchmarking purposes, you will know that they have the same definition for defects and the same definition for lines of code. If the definitions are different, then the benchmark that you get from that organization will be meaningless as the goal for your organization.

There are many different steps that organizations follow in benchmarking. However, most baselining processes have these four steps:

1. Develop a clearly defined baseline in your organization.

This means that all of the attributes involved in your baseline are defined. In our example of defects per lines of code, clearly defining what is meant by defect and a line of code would meet the objective of this step.

2. Identify the organizations you desire to baseline against.

Many factors come into this decision, such as do you want to benchmark within your industry, do you want to benchmark what you believe are leading organizations, do you want to benchmark an organization that uses the same tools that are used in your organization, and do you want to benchmark against organizations with a similar culture.

3. Compare baseline calculations.

Compare how your baseline is calculated versus the baseline calculation in the company you want to benchmark against. Benchmarking is only effective when you benchmark against an organization who has calculated their baseline using approximately the same approach that your organization used to calculate the baseline.

4. Identify the cause of baseline variance in the organization you benchmarked against.

When you find a variance between the baseline calculation in your company and the baseline calculation in the organization you are benchmarking against, you need to identify the cause of variance. For example, if your organization was producing 20 defects per thousand lines of code, and you benchmarked against an organization that only had 10 defects per thousand lines of code you would want to identify the cause of the difference. If you cannot identify the cause of difference, there is little value in benchmarking. Let us assume that the company you benchmarked against had a different process for requirement definition than your organization. For example, assume they use JAD (joint application development) and you did not. Learning this, you may choose to adopt JAD in your organization as a means for reducing your developmental defect rates.

A less formal method for benchmarking is to visit other organizations. This will provide the quality professionals with these benefits:

- The cost and effort to develop new and innovative quality approaches within IT is cost-prohibitive for most companies. Learn from others and don't "reinvent the wheel".
- Comparing quality programs to those in other companies can identify gaps in current processes and lead to obtaining more effective quality practices.
- Interfacing periodically with other quality individuals is good for professional development. Those colleagues will not exist internally unless the company is large.

Visiting another company is a five-step process, as follows:

1. Identify discussion areas

As it is important for the visit to be mutually advantageous, get management agreement on what oral and written information can be shared with the company being visited. Determine the objective of the visit.

- Identify a specific area such as conducting software reviews or independent testing.
- If the objective is to gather general information, convert it into a visit objective, such as identifying the three most effective quality practices used by the other company.

2. Identify target companies

Visits can be within divisions or subsidiaries of the corporation, or to other organizations or corporations. Identify constraints that need to be considered as part of the selection process. Consider items such as whether information can be exchanged with competitors, the availability of travel funds, whether the size of the target company should be similar, and whether the maturity of the company's quality function lends itself to the topics selected (starting a quality function requires a company that has gone through the process). Select three target companies, prioritizing them by the desirability, and obtain management's approval to schedule a visit with them.

3. Schedule the visit

Contact your peers at the targeted companies. Identify yourself and your company. State the purpose for the visit, what you can share, the information you would like to receive in exchange and how the visit would be mutually advantageous. Offer to provide a letter requesting the visit if your colleague needs it to get the visit approved. Set a date and time for the visit. One-half to two days is recommended.

4. Conduct the visit

Visits typically begin with introductions, a restatement of the objectives of the visit and a tour of the other company's facilities to put the size and purpose of the IT function in perspective. Agenda items are then discussed in detail. Written documentation is exchanged and agreement for usage (such as copying, reworking, distributing, etc.) is given. The visit concludes by thanking management of the other company, and, if possible, leaving some memorabilia or some of your company's products as a token of remembrance.

5. Put new practice into use

Select the one best idea from the visit for implementation. Demonstrating a positive outcome from these visits will increase the chance of management approving other visits. Do not try to implement too many things at once.

Assessments against Management Established Criteria

Management can develop a baseline for anything they feel needs to be measured and/or improved. These are baselines that are organizational dependent. There are no industry models or standards against which a baseline can be determined. For example, your management may want to develop a baseline on how effective an organization developed training program is. The organization may

have developed a training program for tool *x* and want to develop a baseline on how the students taking that course perceive the value of the course.

Generally, if management wants to know the efficiency or effectiveness of something in their organization, they should develop a baseline to evaluate performance. The baseline can also measure improvement due to changing that particular item.

The following is an example of a baseline for IT climate. *Climate* is a component of the environment relating to the individual worker's perception of the "climate" in which they perform their work activities. An example of calculating such a baseline follows below.

The organizational climate is the workers' attitude toward their organization. It is a composite of human behaviors, perception of events, responses of employees to one another, expectations, interpersonal conflicts, and the opportunities for growth in the organization. The climate is crucial in creating and maintaining an effective organization, and, therefore, should be periodically evaluated to discover whether the satisfaction level of the employees is positive or negative.

An evaluator can use the six steps below to assess the climate of a specific organization, group, committee, task force, etc.

1. Look at the organization (group, committee, task force, etc.)

Assess the mission and goals of the organization, what it is supposed to produce, and the overriding principles by which it operates.

2. Examine the jobs

Examine each job in the organization. Ask whether the job is necessary, whether it makes full use of the employee's capabilities, and whether it is important in accomplishing the mission and goals of the organization.

3. Assess employees' performance

Evaluate each employee's performance in relation to the organization's mission and goals. For each job being performed, ask if the employee is doing what should be done, is using his or her skills effectively, likes his or her job, and has enthusiasm and interest in performing the job.

4. Evaluate how employees feel about their manager or leader

Good organizational climate requires good leadership. Determine whether each employee within the group likes his or her manager, whether they follow or ignore the requests of their manager, and whether they attempt to protect their manager (i.e., make their manager look good).

5. Create a dialog with the members of the group

Interact with each employee asking a series of hypothetical questions to identify the employee's true feelings toward the organization. Questions such as, “do you feel the organization supports your suggestions”, can help draw out the true feelings of each employee.

6. Rate organizational climate

Based on the responses to steps 1-5, evaluate the climate on the following five-point scale:

- Ideal (5 points)
A fully cooperative environment in which managers and staff work as a team to accomplish the mission.
- Good (4 points)
Some concerns about the health of the climate, but overall it is cooperative and productive.
- Average (3 points)
The organizational climate is one of accomplishing the organization's mission and goals, but no more.
- Below average (2 points)
The individuals are more concerned about their individual performance, development, and promotion than accomplishing the organization's mission.
- Poor (1 point)
There is open hostility in the group and a non-cooperative attitude. As a result, the mission and goals are typically not met.

A negative climate of three points or less often results from individuals having too much responsibility without the authority to fulfill those responsibilities, or management's failure to recognize the abilities of the employees. Negative organizational climates can be improved with the following:

- Develop within the organization a shared vision of what needs to be changed. Get feedback from the employees, and through discussion and compromise agree upon the mission and goals for the organization.
- Change the organization's procedures, as the climate rarely improves without procedural changes.
- Develop a plan for accomplishing the organization's mission that is understood and acceptable to its members. This is normally accomplished if the members help develop the plan.

- If the workers' abilities are not being utilized effectively, reassign tasks to take advantage of their capabilities.
- Develop new methods; for example, try an incentive program or one tied to short-term rewards, such as a paid lunch, a day off, etc.

Assessments against Industry Models

An industry model can be used to measure your IT organization against that industry model. Your current level of performance against that industry model provides a baseline to use to measure improvement. The following section describes the more common industry models used by the IT industry.

Model and Assessment Fundamentals

Purpose of a Model

A model is an idealized concept to be accomplished. Models are usually developed under the auspice of national or international standards organizations and may be customized or 'tailored' to meet new or changing needs. Most industry models define the minimum that has to be accomplished for compliance, and allow compliance to be measured in "pass/fail" terms. Compliance assessments can be through first-party, second-party or third-party audits or other types of evaluation.

Organizations choose to adopt a model for any or all of the following reasons:

- Satisfy business goals and objectives
If the current infrastructure and processes are not meeting business goals and directives, adopting a model can refocus the IT organization to meet those goals and objectives.
- Requirements are imposed by customer(s)
In an effort to improve efficiency and effectiveness, key customers may direct the IT organization to adopt a model.
- For competitive reasons
External customers may only do business with an IT organization that is in compliance with a model.
- As a guide (roadmap) for continuous improvement
A model usually represents the combined learning of leading organizations, and, as such, provides a roadmap for improvement. This is particularly true of continuous models.

Types of Models (Staged and Continuous)

The two types of models that exist, staged and continuous, are discussed below.

Staged models

Staged models are composed of a number of distinct levels of maturity. Each level of maturity is further decomposed into a number of processes that are fixed to that level of maturity. The processes themselves are staged and serve as foundations for the next process. Likewise, each level of maturity is the foundation for the next maturity level.

The Software Engineering Institute's Capability Maturity Model Integration (CMMI^{®1}) is an example of a staged model, although it now has a continuous representation. See page 140 for more information.

Continuous models

In the continuous model, processes are individually improved along a capability scale independent of each other. For example, the project planning process could be at a much higher capability level than the quality assurance process.

ISO/IEC 15504 (SPICE) is an example of a continuous assessment model as is the continuous representation of The Software Engineering Institute's Capability Maturity Model Integration (CMMI[®]) a continuous process reference model. See page 156 for more information.

Model Selection Process

There are a number of software process improvement models publicly available. The most notable are the Capability Maturity Model Integration (CMMI[®]), Malcolm Baldrige National Quality Award (MBNQA), ISO 9001, the Deming Prize, and ISO/IEC 12207, used with ISO/IEC 15504 as the assessment model.

Quality models are designed for specific purposes. For example, the CMMI was developed to evaluate a software contractor's capability to deliver quality software. The MBNQA was designed to identify world-class quality organizations in the United States.

Having a model that is applicable, and that management is committed to achieve, can be a major step in improving the productivity and quality of an IT organization as it moves toward world-class status. A specific model may or may not fully apply to an IT organization, and selecting a model does not have to be an all or nothing decision.

An organization should consider these four criteria when selecting a model:

- Applicability of the model to the organization's goals and objectives.

[®] CMMI are registered in the U.S. Patent and Trademark Office.

A particular model may include criteria that are inconsistent with the organization's goals or objectives. For example, a model may propose the use of customer and employee surveys. The effort and resources required to do this may not be consistent with the organization's current work plan. In that case, the model may either be rejected or modified. It is important that each criterion in a model be directly related to accomplishing the organization's goals and objectives.

- Management commitment

The implementation of any model may require a significantly different management style than currently exists. The new management style may involve different programs or methods of doing work, independent audits to determine compliance to the model, and so forth. If management does not commit budget, schedule and resources to implement and sustain the model, the effort is doomed from the beginning.

- Need for baseline assessments

An organization may need to know how effective and efficient it is. Without measurement against specific criteria, any efficiency and effectiveness assessment is likely to be subjective. Measuring the object against a model improves the objectivity of a baseline assessment.

- Need for measurable goals and objectives

Organizations that desire to improve, need goals and objectives to measure that improvement. If a model is accepted as the means for continuous improvement, then measurable goals and objectives can be defined that describe movement toward meeting the model criteria.

Using Models for Assessment and Baselines

A baseline is a current level of performance. An assessment determines the baseline against the model (goal to be accomplished). In addition to the baseline and the model, the third criterion required for continuous improvement is a method for moving from the current baseline to the desired goal.

The assessment against the model characterizes the current practice within an organization in terms of the capability of the selected processes. The results may be used to drive process improvement activities or process capability determination by analyzing the results in the context of the organization's business needs and identifying strengths, weaknesses, and risks inherent in the processes. Sequencing of the improvements is determined by the organization, and then assessments are used again to show whether or not performance is being accomplished.

Many organizations put new programs into place without a model, and, as a result, have no means of determining whether performance has changed. The effective use of a model, including determining baselines and regular assessments, can help IT management continually improve effectiveness and efficiency.

Assessments versus Products

An assessment is a process used to measure the current level of performance or service. The objective is to establish a baseline of performance. You can measure against an internationally accepted model or an organization's own defined criteria.

An audit is an independent appraisal activity. While assessments may be done by the individuals involved in conducting the work, an audit is performed by someone independent of the activity under audit.

There are two general categories of auditors. These are auditors external to the organization (external auditors), and auditors internal to the organization (internal auditors). The external auditors are normally engaged by the organization's Board of Directors to perform an independent audit of the organization. Internal auditors are employees of the organization that they audit.

Both internal and external auditors are governed by the standards of their profession. In addition, in many countries, the external auditors (frequently referred to as Certified Public Accountants or Chartered Accountants) are covered by stock exchange and governmental regulations.

For the purpose of this study guide, we will define an audit as an independent evaluation of an IT related activity. For example, information systems may be audited after they are placed into operation to determine whether or not the objectives of those systems were accomplished.

Audits assess an activity against the objectives/requirements established for that activity. A report or opinion by the auditors relates to whether or not the activity under audit has or has not met the objectives/requirements for that activity. While internal and independent auditors are subject to their professional standards, auditors within an organization are not held to professional standards.

Many audits within the IT organization are conducted by quality assurance professionals. They frequently perform what are called "test implementation audits." Most of these audits look at a specific activity, such as building a single software system, as opposed to an overview of all the information systems built within a specified period by the organization.

Industry Quality Models

There are many industry models available against which your organization can establish a baseline. One of the most commonly used models for developing a performance baseline is CMMI model. Normally, IT management will establish achieving the criteria of a model as a goal, and then establish the baseline. For example, if an IT organization wanted to be at CMMI Level 3, they would determine their current performance baseline against the CMMI model.

Most baselines are determined by, and measured by, IT personnel. However, when industry models are adopted the organization has a choice to bring in outside consultants to establish the baseline. Although this can be costly, it does provide an independent view.

The following section describes the most commonly used models in the IT industry.

Software Engineering Institute Capability Maturity Model Integration (CMMI®)

In 1986, the Software Engineering Institute (SEI), a federally funded research and development center, began working with the Mitre Corporation to develop a process maturity framework that would help organizations improve their ability to build software solutions. In 1991, the Capability Maturity Model (CMMSM) version 1.0 was released and became the framework for software development and control. In 1992, CMM version 1.1 was released with improvements from the software community. On March 1, 2002, the integrated model for systems and software engineering, Integrated Product and Process Development, and supplier sourcing (CMMI®-SE/SW, version 1.1) was released. This has two different representations – staged and continuous.

Maturity Levels

The CMMI® influences an organization’s culture by instilling discipline and continuous process improvement into the workplace. It is a framework that enables an organization to “build the right products right.” Continuous improvement is based upon long-term objectives accomplished through the implementation of short-term evolutionary steps.

As shown in Figure 10, the CMMI® framework is a method for organizing these steps into five levels of maturity that lay successive foundations to support short and long-term process improvement initiatives.

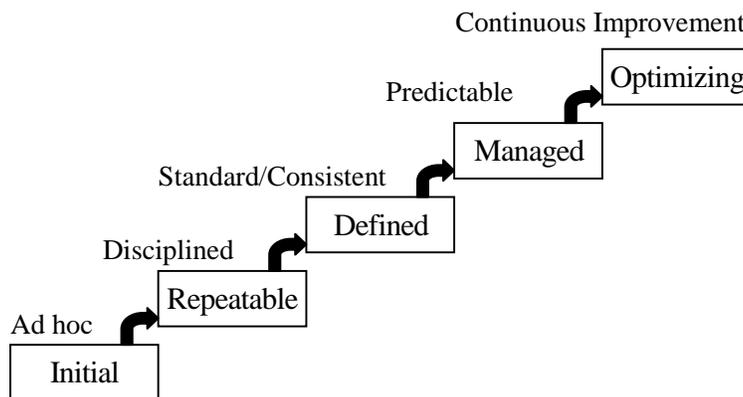


Figure 10. The SEI CMMI® Framework

The five maturity levels define an ordinal scale that enables an organization to determine its level of process capability. The framework is also an aid to quality planning as it affords organizations the opportunity to prioritize improvement efforts.

A maturity level is a well-defined evolutionary plateau for achieving a mature software process. Each level contains a set of goals that, when satisfied, stabilizes specific aspects of the software

development process. Achieving each level of the maturity model institutionalizes a different component, resulting in an overall increase in the process capability of the organization.

The five maturity levels are:

Level 1: Initial

At the initial level, an organization does not provide a stable environment for developing and maintaining software. When an organization lacks sound management practices, the benefits of good software engineering practices are undermined by reaction-driven commitments. In a crisis, projects typically abandon any planned procedures and revert to a code and fix methodology. Success depends on having exceptional people.

The process capability at Level 1 is considered ad hoc because the software development process constantly changes as the work progresses. Schedules, budgets, functionality, and product quality are generally unpredictable.

Level 2: Repeatable

Level 2 organizations have installed basic management controls. Policies for managing a software project and procedures to implement those policies are established. Planning and managing projects is based on experience with similar projects. Realistic project commitments are based upon the results observed on previous projects and on the requirements of the current project. Project managers track software costs, schedules, and functionality. Problems in meeting commitments are identified when they arise. Software requirements and the work products developed to satisfy them are baselined and their integrity is controlled.

The capability of Level 2 organizations is summarized as disciplined, because the ability to successfully repeat planning and tracking of earlier projects results in stability. To be certified at Level 2, organizations must document (define), practice, enforce, train, measure, and improve the following six key process areas:

- Requirements Management
- Software Project Planning
- Software Project Tracking
- Software Subcontract Management
- Software Quality Assurance
- Software Configuration Management

Level 3: Defined

The standard engineering and management processes for developing and maintaining software across an organization are documented, and these processes are integrated as a whole. There is a group responsible for the organization's software process activities, e.g., a standards development group. An organization-wide training program is implemented to ensure that the staff and the managers have the knowledge and skills required to fulfill their assigned roles.

The capability of Level 3 organizations is summarized as standard and consistent because engineering and management activities are stable and repeatable. Product lines, cost, schedule, and functionality are under control and quality is tracked. Process definition and deployment focus on the following key process areas:

- Organization Process Focus
- Organization Process Definition
- Training
- Integrated Software Management
- Software Product Engineering
- Inter-group Coordination
- Peer Reviews

Level 4: Managed

A Level 4 organization sets quantitative quality goals for both software products and processes. Productivity and quality are measured and included in an organization-wide database. Projects achieve control over their products and processes by narrowing the variation in their process performance to fall within acceptable quantitative boundaries.

The capability of Level 4 organizations is summarized as predictable because the process is measured and operates within measurable limits. Quantitative Process Management and Software Quality Management are the two key process areas of Level 4.

Level 5: Optimizing

At Level 5 the entire organization is focused on continuous process improvement. The organization has the means to identify weaknesses and strengthen the process proactively, with the goal of preventing the occurrence of defects. Software project teams analyze defects to determine their root causes, and lessons learned are disseminated to other projects.

The capability of Level 5 organizations is characterized as continuously improving, because projects strive to improve the process capability and process performance. Key process areas of Level 5 are:

- Defect Prevention
- Technology Change Management
- Process Change Management

Components of the Maturity Levels

With the exception of maturity Level 1, each maturity level is decomposed into constituent parts. The decomposition ranges from abstract summaries of each level to their operational definition in specific practices.

Maturity Levels 2-5 are each composed of several key process areas (as noted in the prior section), and each key process area is organized into five sections called common features. The common

features contain key practices that when collectively addressed, accomplish both the specific and generic goals of the process area. Figure 11 illustrates the maturity level decomposition.

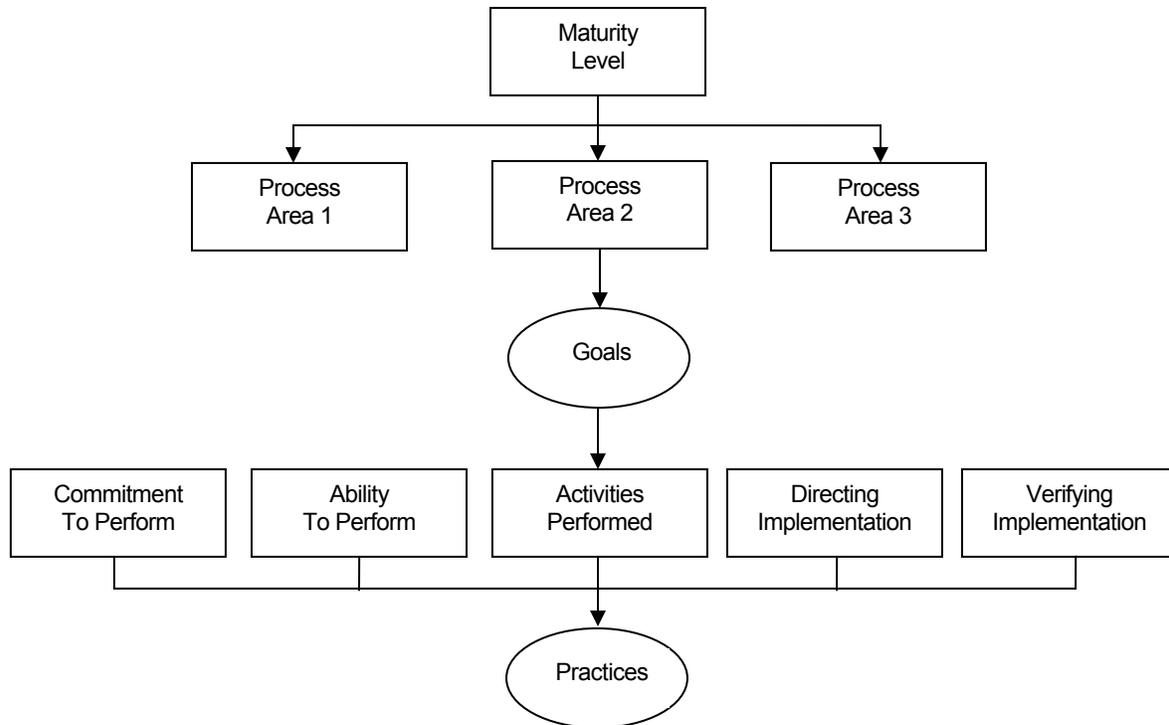


Figure 11. CMMI® Components

Skipping Maturity Levels

The staged representation of the CMMI identifies maturity levels through which the organization should evolve to establish a culture of engineering excellence. Because each maturity level forms a necessary foundation on which to build the next level, trying to skip maturity levels is almost always counterproductive.

Malcolm Baldrige National Quality Award (MBNQA)

The United States national quality award originated from the Malcolm Baldrige National Quality Improvement Act (Public Law 100-107), signed by President Ronald Reagan on August 20, 1987. That act, named after a former Secretary of Commerce, called for the creation of a national quality award and the development of guidelines and criteria that organizations could use to evaluate their quality improvement efforts. Awards are given in the five categories below, with no more than two awards being presented per category per year.

- Manufacturing
- Service

- Small Business
- Educational Organizations
- Health Care Organizations

National Institute of Standards and Technology

The U.S. Department of Commerce is responsible for the MBNQA and Program. The National Institute of Standards (NIST), an agency of the Department's Technology Administration manages the Baldrige Program, as follows:

Board of Overseers

The Board of Overseers is appointed by the Secretary of Commerce and consists of distinguished leaders from all sectors of the U.S. economy. The Board oversees and evaluates all aspects of the Program, including the adequacy of the criteria and processes for determining Award recipients, and advises the Department of Commerce on the Baldrige Program. An important aspect of the Board's responsibility is to assess how well the Program is serving the national interest.

Board of Examiners

The Board of Examiners evaluates Award applications and prepares feedback reports. The Panel of Judges, part of the Board of Examiners, makes Award recommendations to the Director of NIST.

Award Recipients

Award recipients are required to share information on their successful performance and their quality strategies with other U.S. organizations. However, recipients are not required to share proprietary information, even if such information was part of their Award application.

2005 Award Criteria

The award criteria are built upon a set of interrelated core values and concepts. These values and beliefs are embedded behaviors found in high-performing organizations. The core values and concepts are shown in Figure 12 and embodied in the following seven categories. See Skill Category 2 for a description of the core values and concepts.

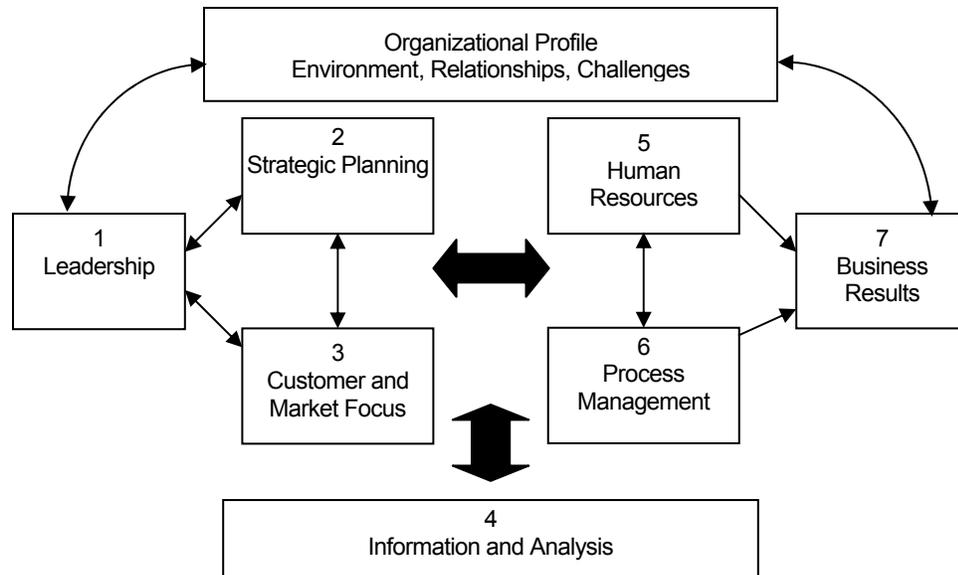


Figure 12. Malcolm Baldrige National Quality Program

1 – Leadership

These criteria address the approach that the organization’s senior leaders take towards values, directions, and performance expectations, as well as how they focus on customers and other stakeholders, empowerment, innovation, and learning. Also examined is how the organization addresses its responsibilities to the public and supports its communities.

2 – Strategic Planning

These criteria address how the organization develops strategic objectives and action plans. Also examined are how the objectives and plans are deployed and how progress is measured.

3 – Customer and Market Focus

These criteria address how the organization determines requirements, expectations, and preferences of customers and markets. Also examined is how the organization builds relationships with customers, and determines the key factors that lead to customer acquisition, satisfaction, and retention and to business expansion.

4 – Information and Analysis

These criteria address the organization’s information management and performance measurement systems and how the organization analyzes performance data and information.

5 – Human Resources

These criteria address how the organization motivates and enables its employees to develop and utilize their full potential in alignment with the organization’s overall objectives and action plans. Also examined are the organization’s efforts to build and maintain a work environment and an employee support climate conducive to performance excellence and to personal and organizational growth.

6 – Process Management

These criteria address the key aspects of the organization's process management, including customer-focused design, product and service delivery, primary business and support processes. It encompasses all key processes and all work units.

7 – Business Results

These criteria address the organization's performance and improvement in key business areas, customer satisfaction, product and service performance, human resource results, financial and marketplace performance, and operational performance. Performance levels relative to those of competitors are also examined.

2005 Award Scoring System

The scoring responses to Award Criteria and Award applicant feedback are based on two evaluation dimensions: Process and Results. Criteria users need to furnish information relating to these dimensions. Specific factors for these dimensions are described below. The award criteria of 1,000 points are distributed as shown in Figure 13.

Category	Points
Leadership	120
Strategic Planning	85
Customer & Market Focus	85
Information and Analysis	90
Human Resources Focus	85
Process Management	85
Business Results	450

Figure 13. Current Baldrige Award Criteria Distribution

Process

Process refers to the methods your organization uses and improves to address the Item requirements in Categories 1-6. The four factors used to evaluate process are Approach, Deployment, Learning, and Integration (A-D-L-I).

“Approach” refers to:

- The methods used to accomplish the process
- The appropriateness of the methods to the Item requirements
- The effectiveness of your use of the methods
- The degree to which the approach is repeatable and based on reliable data and information (i.e., systematic)

“Deployment” refers to the *extent* to which:

- Your approach is applied in addressing Item requirements relevant and important to your organization
- Your approach is applied consistently
- Your approach is used by all appropriate work units

“Learning” refers to:

- Refining your approach through cycles of evaluation and improvement
- Encouraging breakthrough change to your approach through innovation
- Sharing of refinements and innovation with other relevant work units and processes in your organization

“Integration” refers to the *extent* to which:

- Your approach is aligned with your organizational needs identified in other Criteria Item requirements
- Your measures, information, and improvement systems are complementary across processes and work units
- Your plans, processes, results, analyses, learning, and actions are harmonized across processes and work units to support organization-wide goals

“Results” refers to your organization’s *outputs* and *outcomes* in achieving the requirements in Category 7. The four factors used to evaluate results are:

- Your current level of performance
- Rate (i.e., slope of trend data) and breadth (i.e., how widely deployed and shared) of your performance improvements
- Your performance relative to appropriate comparisons and/or benchmarks
- Linkage of your results measures (often through segmentation) to important customer, product and service, market, process, and action plan performance requirements identified in your Organizational Profile and in Process Items.

The Application Review Process

Each applicant receives a feedback report at the conclusion of the review process. The feedback report is written by an evaluation team and contains an applicant-specific listing of strengths and opportunities for improvement based on the Award Criteria. The following four stages constitute the review process:

- An independent review by at least five members of the board.
- A consensus review for applications that score well in Stage 1.
- Site visits to applicants who score well in Stage 2.

- Judge's review and recommendations of award recipients.

Deming Prize and other National / Regional Awards

These use similar categories and variations of the scoring methods. The Deming Prize is awarded in Japan; there are also the European Quality Awards and the Australian Quality Award in their respective regions.

ISO 9001:2000

The ISO 9000 family of standards contains 3 standards and many supporting documents. Under ISO protocols, all standards must be reviewed at least every five years to determine whether they should be confirmed, revised, or withdrawn.

In 1997, ISO conducted a large global survey to better understand their needs. This survey revealed the need to substantially reengineer the model itself, with an emphasis on four needs:

- A common structure based upon a Process Model
- Promote use of continuous improvement and prevention of non-conformity
- Simplicity of use, ease of understanding, and use of clear language and terminology
- Consistency with the Plan-Do-Check-Act improvement cycle

In 2000, the ISO 9000 family of standards integrated the five 1994 Standards into four primary documents as follows:

- **ISO 9000** – Gives an introduction and a guide to using the other standards in the 9000 family.
- **ISO 9001** – Describes a quality system model, including quality system requirements applicable to organizations that design, develop, produce, install and service products.
- **ISO 9004** – Helps companies implement quality systems.

These are supported by the following Standard:

- **ISO 10011** – Includes guidelines for auditing quality systems.

For those interested in applying ISO 9001:2000 to Software the following will be useful;

- **ISO/IEC 90003:2004** Software Engineering – Guidelines for the application of ISO 9001:2000 to computer software

Model Overview

The ISO 9001:2000 standards are based on the following eight quality management principles.

1. Customer Focus

2. Leadership
3. Involvement of People
4. Process Approach
5. System Approach to Management
6. Continuous Improvement
7. Factual Approach to Decision-Making
8. Mutually Beneficial Supplier Relationships

Figure 14 shows the process model, which conceptually presents the quality management system requirements.

The model reflects, graphically, the integration of the four ISO 9001 clauses (discussed below). As it is a model of the complete quality system processes, it is capable of demonstrating the vertical and horizontal process integration in a closed loop manner.

The process model is not intended to reflect processes at the detailed level. However, quality management system requirements for achieving product or service conformity may be placed within the model. A domain model such as CMMI®, a process reference model such as defined in ISO/IEC 12207, or a process model derived from good practices such as in the ITIL® may be easily inserted into the Product Realization Clause of ISO 9001:2000.

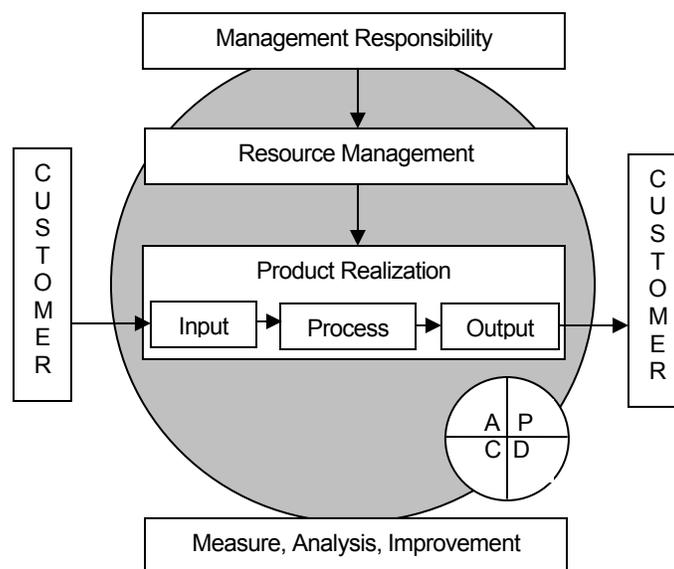


Figure 14. Quality Management Process Model

The following major clauses comprise the ISO 9001 standard. Subclauses are also noted in the description.

Management Responsibility

Management Responsibility includes these subclauses: Management Commitment, Customer Focus, Quality Policy, Planning, Administration, and Management Review. Customer Focus is an integral process within the quality management system.

Top management shall demonstrate that customer needs and expectations have been determined and translated into applicable customer requirements. Top management shall also demonstrate its commitment to meeting customer requirements for their product or service by:

- Creating an environment for awareness and fulfillment of customer requirements
- Establishing the quality policy and quality objectives
- Establishing a quality management system
- Performing management reviews
- Ensuring the availability of necessary resources

Resource Management

Resource Management includes these subclauses: Provision of Resources, Human Resources, Facilities, and Work Environment.

The organization shall determine and provide the resources needed to establish and improve the quality management system. This includes, but is not limited to:

- Human resources, such as assignment of personnel, training, qualification, and competence
- Other resources, such as information, infrastructure, and work environment

Product Realization

Product Realization includes these subclauses: Planning of Realization Process, Customer-related Processes, Design, Development, Purchasing, Production & Service Operations, and Control of Measuring & Monitoring Devices. It is this Clause into which the chosen domain model should be inserted.

The organization shall determine which processes are required to satisfy customer requirements. The sequence and interaction of these processes shall be determined, planned and controlled to ensure they operate efficiently.

The organization shall ensure that these processes are operated under controlled conditions and produce outputs that are consistent with the organization's policy and objectives. For all of these processes, the organization shall:

- Determine how each process influences the ability to meet product or service requirements

- Establish methods and practices relevant to process activities, to the extent necessary to achieve consistent operation of the process
- Verify processes can be operated to achieve product or service conformity
- Determine and implement the criteria and methods to control processes related to the achievement of product or service conformity
- Determine and implement arrangements for measurement, monitoring, and follow-up actions to ensure processes operate effectively
- Ensure the availability of process documentation that provides operating criteria and information to support the operation and monitoring of processes
- Provide the necessary resources for the effective operation of the processes

With regards to the customer-related processes, the organization shall establish a process for identifying customer requirements. The process shall consider:

- Identification of customer requirements
- Review of customer requirements
- Review of the ability to meet defined requirements
- Customer communication
- Customer property (physical property, intellectual property, etc.)

Measurement, Analysis and Improvement

Measurement, Analysis and Improvement includes these subclauses: Planning, Measuring & Monitoring, Control of Nonconformity, Analysis of Data, and Improvement.

The organization shall define and implement measurement, analysis, and improvement processes as a way of demonstrating that the product or service conforms to specified requirements. This includes:

- Measurement of system performance
- Measurement of customer satisfaction
- Internal auditing
- Measurement of processes and product conformity, internal improvements
- Measurement of product and/or service
- Control of measuring, inspection and test equipment
- Analysis of data to view operation trends, and to evaluate the effectiveness of the quality management system, customer satisfaction, and conformance to customer requirements
- Improvement by implementing corrective and preventive action

ISO/IEC 12207: Information Technology – Software Life Cycle Processes

Model Overview

ISO/IEC 12207, which was published in 1995, is the international standard that covers the software life cycle from concept through retirement. It contains a framework for managing, controlling, and improving the software life cycle activities. The standard describes the major processes of the software life cycle, how these processes interface with each other, and the high-level relations that govern their interactions. It has subsequently been amended twice; Amendment 1 defines a Software Engineering Process Reference Model for use with ISO/IEC 15504 Process Assessment.

For each process, the standard also describes the activities and tasks involved, defining specific responsibilities and identifying outputs of activities and tasks. Since it is a high-level standard, it does not detail how to perform the activities and tasks. The standard does not assume a specific life cycle model, nor does it specify names, format, or content of documentation. As a result, organizations applying ISO/IEC 12207 should use other standards and procedures to specify these types of details.

The 17 processes covered in this standard are grouped into three categories, as follows:

Primary Processes

- Acquisition Process
- Supply Process
- Development Process
- Operation Process
- Maintenance Process

Supporting Processes

- Documentation Process
- Configuration Management Process
- Quality Assurance Process
- Verification Process
- Validation Process
- Joint Review Process
- Audit Process
- Problem Resolution Process

Organization Processes

- Management Process
- Infrastructure Process
- Improvement Process
- Training Process

Following is the list of activities for the Development Process, which illustrates the types of activities included for a process:

1. System requirements analysis
2. System architectural design
3. Software requirements analysis (ends with successful reviews followed by a baseline for the software requirements)
4. Software architectural design
5. Software detailed design
6. Software coding and testing
7. Software integration
8. Software qualification testing (ends with successful audits and a baseline for software design and code)
9. System integration
10. System qualification testing (ends with successful audits and a baseline for the design and code of each software configuration item)
11. Software installation
12. Software acceptance test

Activities consist of tasks that are bundled together. Sample tasks include product evaluation, joint review, software configuration management, user documentation, and test. To use activity 7 as an example, tasks for software integration would include user documentation, product evaluation, and joint reviews.

ISO/IEC 12207 defines two categories of reviews: joint technical reviews, and joint management reviews. It states that the joint review process is “a process for evaluating the status and products of an activity of a project as appropriate. Joint reviews are at both the project management and technical levels and are held throughout the life of the contract.”

As the standard is a comprehensive set of processes, organizations can select an appropriate subset that applies to their purpose. It is also expected that an organization will tailor a process to the scope, size, complexity, and criticality of the software product and of the organization by deleting inapplicable activities. Compliance to ISO/IEC 12207 is defined as the performance of those processes, activities, and tasks selected by tailoring.

Target Audience

The target audience for ISO/IEC 12207 is:

- Organizations acquiring a system that contains software or a stand-alone software product
- Organizations that supply software products
- Organizations involved in software operation or software maintenance

The standard is especially applicable for acquisitions, as it recognizes the distinct roles of acquirer and supplier. Its intended use is for the two parties of an agreement or contract that defines the development, maintenance, or operation of software. It does not apply to the purchase of commercial off-the-shelf software products. It is also intended for use with trained, experienced developers, managers, and acquirers of software.

Views of Software Development

Given the target audience, ISO/IEC 12207 includes several different points of view regarding software development:

- Acquirers and suppliers see a contract view, which includes the acquisition and supply processes and begins with a contractual relationship to supply software. Depending on the contract, the supply process may use the development process to create new software, the operation process to provide software operation services, or the maintenance process to repair or improve the software.
- Operators see an operating view, which includes the operations process. The operations process may use the maintenance process.
- Developers and maintainers see the engineering view, which includes the maintenance and development processes. The maintenance process may also use the development process.
- The employer of supporting processes sees the supporting view, which includes the eight supporting processes.
- Managers see the corporate view, which includes the four organizational processes.

During execution of the primary processes, supporting processes are used. For example, reviews, QA, verification, validation, and audits occur during the development process. In addition, documentation is produced, and problem resolution is periodically needed. Organizational processes, the third category, occur in parallel. The organizational processes provide management of the process, and the infrastructure for employee development and training, and improvement of the software development process.

Relationship to Other Standards

ISO/IEC System and Software Standards

The following Standards and Technical Reports will be useful:

- ISO/IEC TR 15271:1998 Information technology – Guide for ISO/IEC 12207 (Software Life Cycle Processes)

The next standard and its guide extend the domain of processes discussed to system life cycles:

- ISO/IEC 15288:2002 Systems engineering – System life cycle processes
- ISO/IEC TR 19760:2003 Systems engineering – A guide for the application of ISO/IEC 15288 (System life cycle processes)

The next group shows just some of the many standards related to the individual processes of ISO/IEC 12207:

- ISO/IEC 15910:1999 Information technology – Software user documentation process
- ISO/IEC 15939:2002 Software engineering – Software measurement process
- ISO/IEC 16085:2004 Information technology – Software life cycle processes -- Risk management
- ISO/IEC TR 16326:1999 Software engineering – Guide for the application of ISO/IEC 12207 to project management

IEEE/EIA 12207

This standard is the United States implementation of the international standard, ISO/IEC 12207. Since ISO/IEC 12207 was intended to be adapted for specific types of applications, IEEE and EIA worked with the United States Department of Defense, and others, to develop software life cycle standards for use in the United States that should:

- Represent the best commercial practice
- Be suitable for application to the complex requirements of defense acquisition
- Be compatible with those of the emerging global marketplace for software

IEEE/EIA 12207 was released as the strategic standard to address the three needs. It also focuses on compliance at the organizational level rather than at an individual project level. This standard was released as a three-volume set that includes:

- IEEE/EIA 12207.0
ISO/IEC 12207 with a United States introduction and 6 additional appendices.
- IEEE/EIA 12207.1

Guidance on the documentation contents, with recommendations for the contents of each type of document and recommendations that expand on the data objectives of Part 0.

- IEEE/EIA 12207.2

A guidebook with additions, alternatives, and implementation approaches to many of the activities and tasks listed in ISO/IEC 12207, plus recommendations for implementing IEEE/EIA 12207 processes in the context of U.S. best practices.

ISO/IEC 15504: Process Assessment (Formerly Known as Software Improvement and Capability Determination (SPICE))

In June 1991, ISO approved a study period to investigate the needs and requirements for a standard for software process assessments. The resulting conclusion was that there was an international consensus on the needs and requirements for a process assessment standard.

ISO/IEC TR 15504:2004 Information Technology – Process Assessment, is a nine part Technical Report referred to as SPICE. Based on a series of trials that have been held around the world since 1994, this Technical Report has become an International Standard comprising of five parts.

Model Overview

ISO/IEC 15504: provides a framework for the assessment of software processes that is appropriate across all application domains and organization sizes. This framework can be used by organizations that plan, manage, monitor, control, and improve the acquisition, supply, development, operation, evolution, and support of software.

The SPICE standard provides a structured approach for assessing software processes by, or on behalf of, an organization with the objectives of:

- Understanding the state of the organization's processes for process improvement (establish process baselines)
- Determining the suitability of the organization's processes for a particular requirement or class of requirements
- Determining the suitability of another organization's processes for a particular contract or class of contracts

It is intended for use by acquirers, suppliers, and assessors as follows:

- Acquirers

Acquirers can reduce uncertainties in supplier selection by enabling risks to be identified before contract award, ensure controls are put in place for risk containment, and determine the current and potential capability of a supplier's software processes for quantitative comparison against competing suppliers.

- Suppliers

Suppliers can determine the current and potential capability of their own software processes, define areas and priorities for software process improvement, and have a framework that defines a road map for software process improvement.

- Assessors

Assessors can use it as a framework to define all aspects of conducting assessments.

One important feature of this model is that it produces a profile of all the assessed processes with a capability level rating for each, rather than simply a pass/fail rating. During the trial period these profiles are being used to determine target profiles, which apply to different sized organizations and domains. Suppliers can then use the target profiles for process improvement goals and acquirers can use the profiles to set acceptable standards for proposals.

Standard Content

The Standard includes the following parts:

1. ISO/IEC 15504-1:2004 Information Technology – Process Assessment – Part 1:
Concepts and Vocabulary

This part of ISO/IEC 15504:2004 provides overall information on the concepts of process assessment and its use in the two contexts of process improvement and process capability determination. It describes how the parts of the suite fit together, and provides guidance for their selection and use. It explains the requirements contained within ISO/IEC 15504, and their applicability to performing assessments.

2. ISO/IEC 15504-2:2003 Information Technology – Process Assessment – Part 2:
Performing an Assessment

This is the only normative part of the Standard, ISO/IEC 15504-2:2003 defines the requirements for performing process assessment as a basis for use in process improvement and capability determination. Process assessment is based on a two-dimensional model containing a process dimension and a capability dimension. The process dimension is provided by an external process reference model, which defines a set of processes characterized by statements of process purpose and process outcomes. The capability dimension consists of a measurement framework comprising six process capability levels and their associated process attributes.

The assessment output consists of a set of process attribute ratings for each process assessed, termed the process profile, and may also include the capability level achieved by that process.

ISO/IEC 15504-2:2003 identifies the measurement framework for process capability and the requirements for:

- Performing an assessment

- Process reference models
 - Process assessment models
 - Verifying conformity of process assessment
3. ISO/IEC 15504-3:2004 Information Technology – Process Assessment – Part 3: Guidance on Performing an Assessment

ISO/IEC 15504-3:2004 provides guidance on meeting the minimum set of requirements for performing an assessment contained in ISO/IEC 15504-2.

It provides an overview of process assessment and interprets the requirements through the provision of guidance on:

- Performing an assessment
 - The measurement framework for process capability
 - Process reference models and process assessment models
 - Selecting and using assessment tools
 - Competency of assessors
 - Verification of conformity
4. ISO/IEC 15504-4:2004 Information Technology – Process Assessment – Part 4: Guidance on Use for Process Improvement and Process Capability Determination

ISO/IEC 15504-4:2004 provides guidance on how to utilize a conformant process assessment within a process improvement program or for process capability determination.

Within a process improvement (PI) context, process assessment provides a means of characterizing an organizational unit in terms of the capability of selected processes. Analysis of the output of a conformant process assessment against an organizational unit's business goals identifies strengths, weaknesses and risks related to the processes. This, in turn, can help determine whether the processes are effective in achieving business goals, and provide the drivers for making improvements.

Process capability determination (PCD) is concerned with analyzing the output of one or more conformant process assessments to identify the strengths, weaknesses and risks involved in undertaking a specific project using the selected processes within a given organizational unit. A process capability determination can provide a fundamental input to supplier selection, in which case it is often termed a "supplier capability determination".

5. ISO/IEC 15504-5:2005 Information Technology – Process Assessment – Part 5: An Exemplar Process Assessment Model

This informative part of ISO/IEC 15504 defines an exemplar Process Assessment Model that meets the requirements of ISO/IEC 15504-2 and that supports the performance of an assessment by providing indicators for guidance on the

interpretation of the process purposes and outcomes as defined in ISO/IEC 12207 Amd 1 and Amd 2 and the process attributes as defined in ISO/IEC 15504-2. It also provides guidance, by example, on the definition, selection and use of assessment indicators.

Reference Models

ISO/IEC 12207:1995 with Amd 1:2002 defines a process reference model and ISO 15504-2 defines process capability, and lists the requirements for a conformant process assessment model, which forms the basis for any model to be used for process assessment. The reference model identifies critical attributes that a process should have to be considered complete and effective, without overly constraining the implementation of the process. Any model compatible with the reference model may be used for assessment, allowing the results of conformant assessments to be translated into a common base. Further guidance on implementing processes may be found in related software standards such as ISO/IEC 12207 or ISO 90003.

The reference model structure uses a process dimension and a process capability dimension.

Process Dimension

Each process description contains a statement of purpose and an outline of the unique functional objectives of the process for a particular environment. A process dimension defines the processes to be assessed, and is characterized by process purposes which are the essential measurable objectives of a process. Satisfying the purpose statements of a process represents the first step in building process capability.

The reference model classifies the processes normally used to develop, maintain, acquire, supply, and operate software into five categories, with each category containing a number of processes. The process categories and processes (listed below) are defined in ISO/IEC 12207 Amendment 1:2002.

Processes are grouped into these categories:

- Customer/Supplier

Processes that directly impact the customer, support development and transition of the software to the customer, and provide for its correct operation and use. The processes in this category are:

- CUS.1 Acquire software
- CUS.2 Manage customer needs
- CUS.3 Supply software
- CUS.4 Operate software
- CUS.5 Provide customer service

- Engineering

Processes that directly specify, implement, or maintain a system and software product and its user documentation. When the system is composed totally of software, the Engineering process deals only with the construction and maintenance of such software. The processes in this category are:

- ENG.1 Develop system requirements and design
- ENG.2 Develop software requirements
- ENG.3 Develop software design
- ENG.4 Implement software design
- ENG.5 Integrate and test software
- ENG.6 Integrate and test system
- ENG.7 Maintain system and software

- Support

Processes that may be employed by any of the other processes (including other supporting processes) at various points in the software life cycle. The processes belonging to the Support category are:

- SUP.1 Develop documentation
- SUP.2 Perform configuration management
- SUP.3 Perform quality assurance
- SUP.4 Perform work product verification
- SUP.5 Perform work product validation
- SUP.6 Perform joint reviews
- SUP.7 Perform audits
- SUP.8 Perform problem resolution

- Management

Processes that contain generic practices, which may be used by anyone managing any type of project or process within a software life cycle. Processes in this category are:

- MAN.1 Manage the project
- MAN.2 Manage quality
- MAN.3 Manage risks
- MAN.4 Manage subcontractors

- Organization

Processes that establish the business goals of the organization and develop process, product, and resource assets that, when used by the projects in the organization, help the organization achieve its business goals. Organizational processes generally have a broader scope than

software processes, but software processes are implemented in a business context thus, they require an appropriate organizational environment in order to be effective. Organizational processes build infrastructure, and make the best of what is available in any one part of the organization (effective processes, advanced skills, quality code, and good support tools) available to all. Organizational processes are:

- ORG.1 Engineer the business
- ORG.2 Define the process
- ORG.3 Improve the process
- ORG.4 Provide skilled human resources

Process Capability Dimension

Process capability dimension describes the scale for measurement of capability. Evolving process capability is expressed in terms of process attributes grouped into capability levels. Process attributes apply to all processes - they are features that can be evaluated on a scale of achievement to provide a measure of the capability of the process. Each process attribute describes a facet of the overall capability of managing and improving the effectiveness of a process in achieving its purpose and contributing to the business goals of the organization.

Each capability level represents an incremental evolution in the management and control of the processes, so that the assessment models provide a road map for increasing capability. There are six capability levels in the reference model:

- Level 0 – Incomplete
There is general failure to attain the purpose of the process. There are no easily identifiable work products or outputs of the process.
- Level 1 – Performed
The purpose of the process is generally achieved, but the achievement may not be rigorously planned and tracked. Individuals within the organization recognize that an action should be performed, and there is general agreement that this action is performed as and when required. There are identifiable work products for the process, and these testify to the achievement of the purpose.
- Level 2 – Managed
The process delivers work products of acceptable quality within defined timeframes. Work products conform to specified standards and requirements. Performance according to specified procedures is planned and tracked. The primary distinction from the Performed Level is that the performance of the process is planned and managed and progressing towards a defined process.
- Level 3 – Established
The process is performed and managed using a defined process based upon good software engineering principles. Individual implementations of the process use approved, tailored versions of documented standard processes. The resources needed to define the process are in

place. The primary distinction from the Managed Level is that the process is planned and managed using a standard process.

- **Level 4 – Predictable**

The defined process is performed consistently within defined control limits, to achieve its goals. Detailed measures of performance are collected and analyzed, leading to a quantitative understanding of process capability and an improved ability to predict performance. Performance is objectively managed. The quality of work products is quantitatively known. The primary distinction from the Established Level is that the defined process is quantitatively understood and controlled.

- **Level 5 – Optimizing**

Performance of the process is optimized to meet current and future business needs, and the process repeatedly meets its defined business goals. Based on the business goals of the organization, quantitative process effectiveness and efficiency goals (targets) for performance are established. Quantitative feedback allows continuous process monitoring against these goals, and improvement is achieved by analyzing the results. Optimizing a process involves piloting innovative ideas and technologies and changing non-effective processes to meet defined goals or objectives. The primary distinction from the Predictable Level is that the defined and standard processes undergo continuous refinement and improvement, based on a quantitative understanding of the impact of changes to these processes.

The Assessment Process

The reference model alone cannot be used as the basis for conducting reliable and consistent process capability assessments since the level of detail is not sufficient. The process purpose and capability attributes in the reference model need to be supported with a comprehensive set of indicators of process performance and capability.

The assessment process begins by evaluating the input, which includes the purpose for the assessment, its scope, and any constraints, responsibilities, or additional information. The assessment process then involves selecting a number of processes from Part 2 (Reference Model), assessing them using indicators (from Part 5 or another conformant assessment model) in accordance with the requirements from Part 3, and producing the process ratings (capability level) for each process, and any other information that might have been requested (in the input) by the assessment sponsor. Ratings and other information are documented in an assessment record.

The rating scale defined below describes the levels of achievement of the defined capability of the process attributes.

- **N – Not achieved**
There is no evidence of achievement of the defined attribute
- **P – Partially achieved**
There is some achievement of the defined attribute

- **L** – Largely achieved
There is significant achievement of the defined attribute
- **F** – Fully achieved
There is full achievement of the defined attribute

One of the main reasons for conducting an assessment is to ensure comparability with other assessment outputs. This is possible by implementing the requirements for rating processes and calculating results within the measurement framework, and reporting them in a way that makes the results of the calculation obvious.

In process capability determination, the proposed capability of selected processes is analyzed against a target process capability profile to identify the risks involved in undertaking a project using the selected processes. The proposed capability may be based on the results of relevant, previous process assessments, or may be based on an assessment carried out for the purpose of establishing the proposed capability. Results from this may lead to process improvements.

Process assessments also provide a way to characterize the current practice within an organization in terms of the capability of the selected processes. Analysis of the results in light of the organization's business needs identifies strengths, weaknesses, and risks inherent in the processes. From the analysis it can be determined whether the processes are effective in achieving their goals, and if there are significant causes of poor quality or overruns in time or cost. These results provide the drivers for prioritizing improvements to processes.

Figure 15 shows the relationships between process assessment, capability determination, and process improvement.

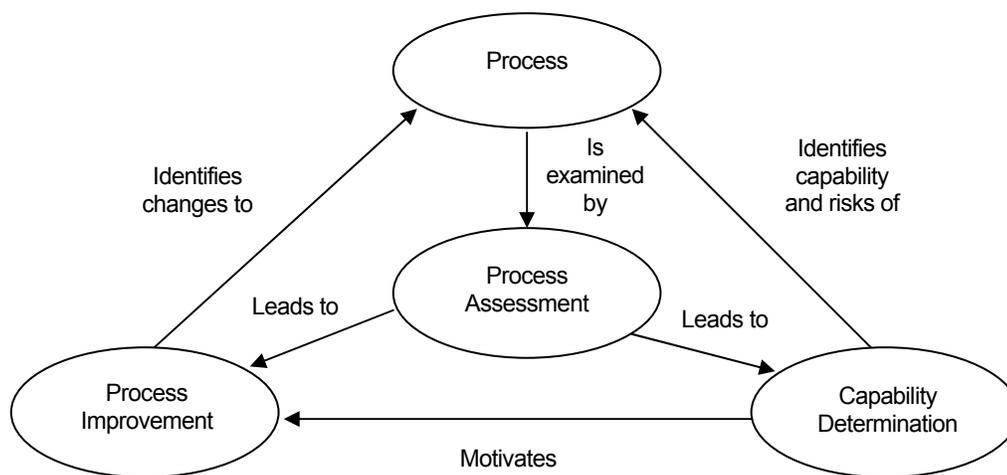


Figure 15. Uses of Process Assessment

Relationship to other International Standards

This Standard is complementary to several other International Standards (including ISO 9000 and ISO/IEC 12207 and 15288) and other models for evaluating the capability and effectiveness of organizations and processes. The continuous representation of the CMMI[®] is an ISO/IEC 15504-2 conformant process reference model, and the SCAMPI[®] assessment method provides a conformant process assessment model.

Post-Implementation Audits

A post-implementation audit is conducted to determine any, or all, of the following:

- The system objectives were met
- The system specifications were implemented as specified
- The developmental standards were followed
- The IT quality objectives (e.g., maintainability) were achieved

Post-implementation audits are a quality assurance activity. The post-implementation audit is used to assess the ability of the IT organization to perform effective and efficient work. The results of the audit will be used to both improve the software system *and* to improve the process that builds software systems.

Quality Assurance

Quality Assurance is a professional competency whose focus is directed at critical processes used to build products and services. The profession is charged with the responsibility for tactical process improvement initiatives that are strategically aligned to the goals of the organization. This category describes the management processes used to establish the foundation of a quality-managed environment:

<i>Establishing a Function to Promote and Manage Quality</i>	165
<i>Quality Tools</i>	185
<i>Process Deployment</i>	212
<i>Internal Auditing and Quality Assurance</i>	224

Establishing a Function to Promote and Manage Quality

Inadequate attention to quality in IT normally results in high systems maintenance costs and customer dissatisfaction. Through the promotion of effective quality practices, the quality function can reduce the cost of systems development, operation, and maintenance, and can improve customer satisfaction.

The quality function has been implemented in hundreds of companies. Quality functions have demonstrated that quality can be defined and measured. Experience has shown that effective quality does increase productivity, and pays for itself by actually reducing costs.

The key concept to establishing a quality function is establishing the need for quality. Until management believes there is a need for quality improvement, the real impediment, management, cannot be dealt with.

A quality function exists when a specific individual/group is assigned the responsibility to assist in improving quality. While individual workers have responsibility for the quality of their products and services, it is management's responsibility to ensure that the environment is one in which quality can flourish. The ultimate responsibility for quality rests with senior management.

Many people argue that because everyone has some quality responsibility, a staff function for quality is unnecessary. That argument is theoretically correct, but in practice unless there is a group charged with responsibility for ensuring quality, the pressures of other priorities such as meeting schedules and budgets frequently takes precedence over quality. The following example shows how this happens in a typical systems development project.

The four project variables are scope, schedule, resources, and quality. The management challenge in completing the project can be illustrated as a dashboard of four system attribute dials, which get set according to the project criteria as illustrated in Figure 16. The four dials are interconnected, so movement of one dial affects one or more of the other dials.

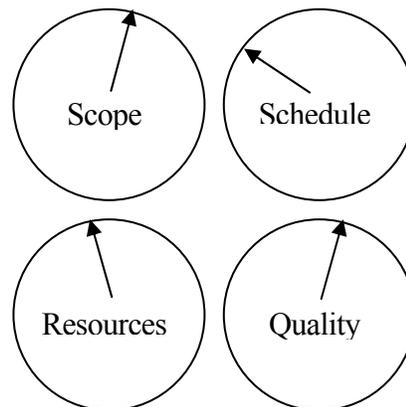


Figure 16. The Four Interconnected Variables of IT Development

All goes well until one of the variables needs to be changed. For example, the project team agrees to install a new requirement, but if management holds schedule and resources firm while moving the scope, something must give. Since the dials are interconnected, what must give is quality. Reduced quality occurs as less testing, less documentation, or fewer controls. One role of the quality function is to hold the quality dial firm, so that scope changes cause the resources or schedule dials to move, and not cause a reduction in quality.

"Of all men's miseries, the bitterest is this: to know so much and have control over nothing."

Herodotus 484-424

The Challenges of Implementing a Quality Function

Under the assumption that it is a good idea for an organization to establish a quality function, it is helpful to analyze the challenge of gaining management's acceptance of the establishment and operation of the activity. During this analysis, two basic challenges emerge, and either or both may be present in any particular organization. One challenge is that the organization is not ready to establish a quality function, and the other is that the salesmanship for getting the process sold to executive management is not available even though the organization is ready. This observation is shown in Table 4.

Table 4. Organization Readiness Matrix

	Poor Salesman	Good Salesman
Organization Not Ready	Don't Bother	Challenge 1
Organization Ready	Challenge 2	No Problem

If an organization is not ready, and also lacks the sales skills to convince the management group, the probability of success is so low that it is best not to bother with implementation at this time. A ready organization with good salesmanship can take care of itself. The other scenarios present different challenges. Organizations that are not ready, but have the salesmanship, are referred to as "Challenge 1." Organizations that are ready, but lack the salesmanship, are referred to as "Challenge 2." Each challenge is analyzed separately.

▪ Challenge 1

The solution is to get ready. To get ready, an organization needs the following:

- The presence of a competent, trained staff that is capable of performing quality work when building and maintaining systems. Additionally, there must be backup for these individuals. In a moderate-size organization, quality initiatives might be assigned to one person. With the substantial effort to get it going, it can be traumatic if that individual becomes promoted or should leave.
- A good IT management team that is capable of setting up the quality function, monitoring it, and making it work.
- The existence of good management tools, especially project management and systems development methodologies that spell out what to do when, and are specific about where the quality review functions belong in the development cycle and in maintenance.
- A limit on the number of internal improvement projects to be done at once.

- Good credibility. If there has been a history of not doing things that were promised it may be premature to initiate a quality function until a track record of satisfying promises has been accomplished.
- To be mentally prepared for the length of time (measured in months and years, not weeks) to get a new and important function like this into operation and stabilized.
- **Challenge 2**

The process of selling a new quality function is similar to the process of selling anything associated with internally improving an organization. Some items to consider include the following:

- To be an issue of real value to the total organization, the quality function must be heartfelt to the salesperson. Without a very strong belief in quality, no one can sell it.
- When the proposition is taken to the management group, it is essential for the salesperson to speak the language of the listener.
- Management values their time, which places a burden on the presenter to think the proposition through, present the case, ask for a clear decision, and then leave the room when receiving an affirmative answer. Most executive managers will buy into any well thought-out plan that has the necessary controls built in and where they can conceptually see the logic of how the pieces fit together.
- It is essential to present a realistic picture. Extreme benefits should not be promised, nor should the salesperson claim that problems could be solved in a very short period of time. Most executive managers are astute enough to see through the unrealistic claims rapidly, weakening the total case.
- There is an important principle of getting some things accepted, which is referred to as the Mafia Rule. This is an offer that management can't refuse. Offer that the management group can make all of the decisions on quality issues, while the proponents of the quality function will handle the administrative details.
- Ideas are typically accepted only after the person presenting the ideas is accepted. If the salesperson's personal relationship with management is weak, some "fence-mending" should be done before asking management for a major decision.
- Using outsiders can sometimes help in getting the right decision. The simple process of paying for advice somehow makes that advice seem more important.

How the Quality Function Matures Over Time

The MBNQA program has stated that it takes five to seven years to mature a quality management system. The SEI of Carnegie Mellon University states that it takes at least three years to move from SEI capability maturity Level 1 to Level 3. It is during this maturation period that the role of the QA analyst significantly changes from quality control to quality assurance to quality consulting. Other individuals performing these roles also change as the organization's quality management philosophy matures.

Three Phases of Quality Function Maturation

The maturation of the quality management system can be divided into three phases:

Initial Phase

The initiation point of the quality management system can be considered the formalization of quality control activities. An organization in this phase is results-driven, focusing on defining and controlling product quality. It normally takes at least two years in this phase of maturity for management and staff to recognize that quality cannot be tested into a product; it must happen through process maturity. In this phase:

- A quality control department performs quality control activities. This department may be called independent testing, software QA, or other names that identify an activity focused on product control. The quality practitioners performing the work are viewed as inspectors or testers, regardless of their job title. In many IT groups, quality control is an ongoing activity performed through maintenance, rather than controlling the initial development of products and services.
- A standards committee or standards manager normally performs QA activities focused on process definition. However, these processes are rarely defined fully, and rarely followed in detail. Also, the standards committee tends to define tasks, such as assigning job numbers, rather than defining processes. Once defined, the tasks are usually not improved.
- The role of the quality consultant may not be performed at all. If performed, it tends to be by an individual member of management or initiated by the organization's auditing function. It is more often a single advocate for quality, rather than an organizational responsibility.

Intermediate Phase

In this phase an organization's objectives move from control to assurance. The emphasis is on defining, stabilizing, measuring, and improving work processes. While process improvement occurs before and after this phase, it is during this phase that resources are allocated to make process maturity happen. It takes between two and four years for significant process maturity. At this level products and services achieve consistency in product quality (consistency being the prerequisite to improved quality and productivity). In this phase:

- Quality control is a shared responsibility of the customer or user, worker, and the QA analyst. For example, the customer may be involved in acceptance testing and requirement reviews; the worker may perform checklists and do some independent verification and validation activities or do it in conjunction with peers or the QA analyst.
- Process definition and improvement are emphasized, and are performed by the quality function, QA analyst, and consultants. Under the direction of the QA analyst, teams may be formed and facilitators obtained. The QA analyst often performs the facilitator role, the team leader role or the reviewing role.
- The quality function or QA analyst also acts as quality consultant. That role may be fulfilled on an ad hoc, part-time basis, or as requested. Quality Councils are also normally formed (see Skill Category 2), and may act as consultants to their staffs.

Final Phase

During this phase, objectives such as consulting, motivating, and benchmarking move the organization toward optimization. The MBNQA program defines such an organization as a “world-class” organization. World-class begins when work and management processes are well defined, are being continually improved, and are producing high-quality results that yield high customer satisfaction. These processes are integrated to obtain maximum customer satisfaction at minimum cost. Here:

- The worker has processes to build and check work (i.e., quality control processes); and, as a result, quality control and the quality of the work become the responsibility of the worker. The quality control or check process may be performed by the individual doing the work, or by a group of workers or peers.
- Workers (normally teams) are assigned responsibilities for process definition, measurement, and improvement. These teams may require facilitators, or a team member may be trained in facilitating skills.
- The primary role of QA analysts is to perform quality consulting to management and employees in promoting and implementing quality initiatives. For example, instead of just interpreting requirements from customers, they now work with customers on strategic plans. They may participate in the development of a product instead of simply reviewing it. As a consultant, the QA analyst is knowledgeable in quality principles and practices, is an advocate for quality, and accepts the responsibility for motivating the organization’s employees to improve quality and customer satisfaction.

While the QA analyst can impact and facilitate the changing role, there are two major drivers that cause that role to change. The QA analysts’ impact on maturing their role is more effective when it is directed at the drivers of change, rather than at the change itself.

Drivers that Change the Role of the QA Analyst

The two major drivers that change the role of the QA analyst are the management philosophy used in the IT group, and the personal belief system of managers. The relationship between the management philosophy, the management belief system, and the QA analyst’s role is illustrated in Figure 17.

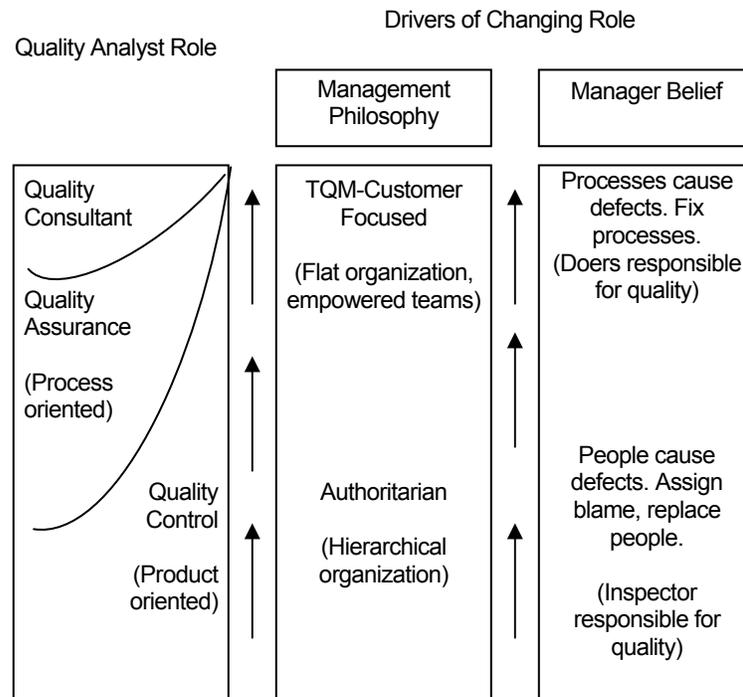


Figure 17. Changing Role of the QA Analyst

Management Philosophy

As management moves from an authoritarian philosophy in the initial phase toward the quality management philosophy, the need from their quality staff changes. Quality management as a process-oriented philosophy leads management to begin defining, measuring and improving processes. This creates the need for a quality function directed at process definition and improvement.

As the quality management philosophy matures, the organization moves from a hierarchical structure, to teams that are organized and empowered to define, measure and improve their own processes. Thus, some activities initially performed by QA analysts are transferred to the empowered teams. The QA analyst then consults those individuals with quality responsibilities.

Personal Belief System of Managers

The management philosophy and managers' belief systems may not be in synchronization. A corporation may dictate a quality management philosophy, but the IT manager's approach is more authoritarian. These managers may talk quality management, but walk authoritarian approaches.

With authoritarian management, the belief is that defects are caused by people, and are only correctable by people doing better work. If people are unable to perform their work correctly, the solution is to organize independent groups to validate the correctness of work. This philosophy may lead to independent groups checking on independent groups.

As the managerial belief system moves toward accepting that processes cause defects, and that management is responsible for the processes, it understands workers can no longer be blamed for defects and that management must achieve the solution. With this belief system, management needs the support of the QA analyst, first as a quality group and then as a consultant to define and drive the quality initiatives to achieve high customer satisfaction, high productivity, and high levels of quality.

Quality Function Recommendations

Quality management is only effective when a professional support staff trained in quality principles and practices supports the quality management implementation. Conceptually, a support staff is not needed, but in practice it is one of the important drivers of implementing the quality management process. The quality support staff should oversee the control of quality as the financial support staff oversees financial control.

Dr. Deming says that a leader of statistical methodology is needed to drive the quality management process. He states in his book *Out of the Crisis*, “There will be a leader of statistical methodology responsible to top management. He must be a man of unquestioned ability. He will assume leadership in statistical methodology throughout the company. He will have authority from top management to be a participant in any activity that in his judgment is worth his pursuit. He will be a regular participant in any major meeting of the president and staff. He has the right and obligation to ask questions about any activity, and he is entitled to responsible answers. The choice of application for him to pursue must be left to his judgment, not to the judgment of others, though he will, of course, try to be helpful to anyone that asks for advice. The nonstatistician cannot always recognize a statistical problem when he sees one.”

Deming says that there must be other statistical methods people in various divisions with a dotted-line relationship to the leader of statistical methods. He also recommends that there be a support team. “This support team may be as small as one person who performs all the functions (facilitator, trainer, and statistician) or it can be quite large in large organizations with people dedicated to each of the three functions. It is important that the organization recognizes this requirement, staffs it, and trains those assigned this responsibility.”

Support in Corporate Quality Management Environment

The benefits of quality management are rarely achieved without the assistance of QA analysts. The development and deployment of a corporate quality management program can be enhanced, both corporately and within an IT group, by assimilating and leveraging an existing IT quality function into the implementation of the corporate program. Both the IT QA analyst and IT management can support the corporate quality management program.

Support of Corporate Quality Management with Repository of Quality Information

The IT quality function can help define and implement the scope, validity, use, and management of the data and information that underlie the corporation’s corporate quality management system. Also involved is assuring the adequacy of the data, information, and analysis to support a prevention-based approach to quality and customer satisfaction built upon “management by fact.”

Specifically, the responsibility of the IT quality function involves:

- Developing the company's base of data and information used for planning and day-to-day management. Evaluation of how quality and data and information reliability, timeliness, and access are assured.
- Developing the company's approach to selecting quality-related competitive comparisons and world-class benchmarks to support quality planning, evaluation, and improvement.
- Determining how data and information are analyzed to support the company's overall quality objectives.

IT Management Responsibilities in Corporate Quality Management Environment

IT management involved in a corporate quality management program should undertake the following four actions in an effort to rethink and assign their quality responsibilities:

1. Determine what new, changed, and continued quality responsibilities will be assigned to the IT group as a result of the corporate quality program.
2. Evaluate the capability of the IT group to deploy quality initiatives within and outside the group. Define the skills, commitment, and quality approaches currently available to line management.
3. Determine the value added by utilizing IT QA analysts to support corporate quality management initiatives.
4. Develop a plan to assimilate and leverage the current function into the quality support activity needed to support both the corporate quality management program and IT quality responsibilities.

Implementing an IT Quality Function

The seven steps for implementing a quality function are listed below, and then detailed in the following sections.

1. Develop a charter.

Determine the responsibilities and activities of the quality function.

2. Identify the quality manager.

Select a quality "fanatic" to head the quality function.

3. Locate organizationally the IT quality function.

Determine to whom the quality leader reports.

4. Build support for quality.

Initiate programs that will encourage both management and staff to support quality processes.

5. Staff and train the quality function.

Determine what is needed to make quality happen, identify and select the people to do it, and provide them with the training they need.

6. Build and deploy the quality toolbox.

Select effective quality tools and implement them throughout the organization.

7. Drive the implementation of the quality management environment.

The quality manager will need to spend a significant amount of time maturing the quality management environment.

These steps are listed sequentially, but may be implemented concurrently. In addition, some steps are only performed periodically, such as selecting a quality manager, while other steps will be performed continually, such as driving the implementation of the quality environment.

Step 1: Develop a Charter

IT management must plan for quality and develop an IT policy that documents the objectives for the quality function. An important component of the quality function is to facilitate the analysis, improvement, and integration of work processes in order to meet organizational strategic and tactical goals more effectively and efficiently. With the quality manager (when appointed), IT management must also create a quality plan to document a visible strategy for implementing the policy. The quality plan should be referenced from, or included within, the business plan.

The charter of a quality function is management's authorization to make quality happen. It is a job description for the quality function, explicitly explaining to all affected parties the function's scope and responsibilities. Management may select a manager for the quality function and let that person run with the ball. This takes minimal effort on their part, but also limits the probability of success. The more desirable approach is to draft a Quality Charter, and then staff the function in a manner that will drive the accomplishment of that charter. With the second approach, the IT manager and quality manager can share the task of completing the Charter and Quality Plan.

Developing a Quality Charter is necessary to:

- Determine the caliber of people needed to fulfill the quality responsibilities, and ensure there are sufficient resources to perform them.
- Determine where to put the function in the organization so that the needed authority and management support to successfully accomplish the assigned tasks can be obtained.

- Limit the scope of the quality function to a group of achievable tasks.
- Notify the affected parties of the responsibility and authority of the quality group.
- Place responsibility for product quality with the line functions/project teams, instead of making the quality function responsible for the quality of the application systems.
- Provide a tool for measuring the effectiveness of the quality group.

The Quality Charter should contain the following:

- **Scope**
The scope should involve activities supporting the implementation of the quality policy, and closing the producer and customer gaps (see Skill Category 1).
- **Objectives**
The objectives clarify the role of the quality function, and normally move through three phases as quality processes mature. Product quality is the initial focus, followed by the definition and improvement of work processes. Finally, objectives are oriented towards optimization. The next section covers the maturation of the quality function.
- **Responsibilities**
Responsibilities assigned to the quality function should be important, meaningful, accomplishable and measurable. They should be defined in sufficient detail so that the accomplishment or non-accomplishment of the tasks can be determined. The quality function is not responsible for product quality, but will have responsibilities supporting quality initiatives. These responsibilities should include maintaining current knowledge of quality principles and practices; identifying companies and sources that can provide best processes and practices; training, coaching, and facilitating; and maintaining and analyzing databases of information that will help identify process and product weaknesses as candidates for improvement. Additional responsibilities may include participating on committees; providing process management expertise; serving as a centralized resource for measurement analysis and reporting; acting as custodian for processes by formatting, editing, publishing, distributing, and controlling access to changes, etc. Auditing for process deployment and compliance may also be a responsibility, although this is more desirable separated from QA.
- **Authority**
The quality function needs the authority to perform its responsibilities. This authority should give the quality staff access to all documentation and data that is necessary to analyze and identify problems. In many organizations the quality function receives data that is not available to line management. For example, from product inspections it may know the type and quantity of defects made by individuals in building products. Since inspections are designed to assist the worker in improving quality, the usefulness may be compromised if the worker believes management will use them

negatively in a performance appraisal. On the other hand, the quality staff needs that information to identify defect-prone products and processes.

- **Ongoing Quality Programs**

The charter should specify programs that will be performed regularly by the quality function. These might include building and analyzing defect databases, assisting in the development of work processes; summarizing and analyzing quantitative data; training the IT staff in quality principles, practices, and processes; and assisting in the performance of tasks such as software reviews and inspections. These programs may change over time as the quality program matures. For example, initially the quality function may chair reviews and inspections, but when that function matures, the review and inspection responsibilities go to the developers of the products.

Step 2: Identify the Quality Manager

Led by the quality manager, the quality function is in the business of selling quality, and must compete with the other concerns and principles of management, such as schedules and budgets. Thus, the quality manager is a significant factor in determining the ongoing success of the quality activities. The wrong individual may waste resources monitoring compliance to policies regardless of their merit and value, or may only focus on quality control and ignore the assurance function.

The quality manager should be the quality conscience and a spokesperson for quality in the department – able to lobby for quality IT objectives and recommend how to achieve them. The quality manager will be passionate and enthusiastic about the function as a vehicle for achieving quality. This position requires self-motivation, excellent oral and written communication skills, and the ability to work well with, and through, people. They should be objective and open-minded about different approaches to IT methods and application design. As a leader, the quality manager must be willing to let others accept credit for their recommendations.

QAI surveyed quality groups to determine what skills they believed were necessary to be successful in quality. Table 5 shows their responses in order of priority.

Table 5. Ranking of Desired Skills for Success in Quality

Rank	Skill	Explanation
1	Verbal communication	Able to communicate orally with management, users, and systems analysts
2	Written communication	Able to communicate through letters and reports with management, users, and systems analysts
3	Systems knowledge	Understands how systems are designed and constructed
4	Knowledge of operations	Understands how computer applications are operated on the computer hardware and software
5	Computer systems knowledge	Understands how computer systems are designed and constructed
6	Business system design	Understands how to solve a business problem independently of the methods by which that solution will operate
7	Project management	Has experience in managing a systems project
8	Programming	Can design, program and debug a program in one or more languages

"Success requires enthusiasm and excitement about what you're doing. If you're not happy every morning when you get up, leave for work, or start to work at home – if you're not enthusiastic about doing that, you're not going to be successful. The other thing you need in, probably, an equal proportion is a lot of luck, because there are a lot of people with the same ability, who worked very hard, who haven't made it and who deserved to make it."

Donald M. Kendall

Former Chairman and Chief Executive Officer of PepsiCo, Inc.

Step 3: Locate Organizationally the IT Quality Function

The quality function establishes the policies, procedures, standards, and processes or work instructions used in the IT function. To be effective, the quality function must not report to a line manager responsible for any part of the day-to-day production work. Quality functions can be located differently within IT organizations. Based on research from QAI, the percent of quality groups in each location is noted.

- Reports to senior IT manager – 50%

This is the best positioning because it gives the quality manager immediate access to the IT manager to discuss and promote quality issues. When the quality manager reports elsewhere, quality issues may not be raised to the appropriate level or receive the necessary action.

- Reports to manager of systems programming – 25%

The quality function is normally the weakest with this reporting structure, because most quality reviews involve applications for which this manager is responsible. Normally, it is more desirable to solve problems within a group; thus, it is easy to squelch many quality concepts.

- Reports to manager of computer operations – 15%

When using applications developed in-house, the quality function can be strong here because it has the ability to stop poor quality projects from being run on the computer. On the negative side, placement here rarely allows the opportunity to influence applications during the development process, as finished products are judged. Changes made at this point in time may be very costly. With this reporting structure, the quality function also lacks the ability to change work processes in other areas of IT.

- Reports outside of the IT function – 10%

This reporting structure provides the quality function with independence not possible when the group reports within IT. This organizational arrangement is more common when the applications will either be sold or used in locations other than where the application was developed. In other words, if the organization bases its reputation on quality (i.e., the product is sold), or maintenance personnel are not readily available (i.e., the application is run in remote areas), the stamp of approval by an independent quality function is frequently warranted. In this capacity the quality function is often an independent test group.

Step 4: Build Support for Quality

Nobody is against quality. Members of an organization will not state that they produce poor-quality products and services. On the other hand, quality is only one of many programs vying for the time and attention of people. There is merit to the old theory that the "squeaky wheel gets the grease."

Obtaining and sustaining support for quality is an ongoing activity. Many quality managers say that support is cyclical. There are times when quality is very important, and then it diminishes in importance. Thus, the level of difficulty in building support for quality changes as the level of support increases and decreases.

The following tasks are helpful in building support for quality:

- Estimate the Cost of Quality (COQ)

In times when people are asked to do more with less, identifying nonproductive costs (COQ) provides opportunities to use wasted resources for other purposes. Skill Category 1 explains COQ.

- Teach a quality vocabulary

One of the impediments to quality in many organizations is the lack of a quality vocabulary. People do not understand the terminology, and thus do not understand the

objectives and methods of quality improvement. The inability to understand terms like quality control, quality assurance, quality improvement program, or standards significantly reduces communication, and, thus, reduces progress in promoting quality. Appendix A provides a glossary of quality terms.

- Issue a quality newsletter

Issue a short, periodically produced newsletter that promotes good practices and explains better ways to achieve quality. The newsletter should describe problems that have occurred and solutions to those problems so that others can take the necessary steps to avoid the same problems in their areas.

- Include quality in job descriptions

The job descriptions of all IT group members should state that part of their responsibility is to perform quality work.

- Meet daily on quality

Some IT functions meet for a few minutes each morning to discuss problems that occurred the previous day, how to resolve them, and how to prevent them from occurring in the future. The objective is to concentrate management's attention for a few minutes each morning on improving quality.

- Reward quality work

Give rewards to those people who follow processes and produce defect-free products.

- Organize a quality improvement team

Organize a small team of IT personnel to solve specific problems for the department. Use a process improvement process such as that defined in Skill Category 6.

- Differentiate quality control from quality assurance responsibilities

Quality control is a line management responsibility and must not be practiced by quality assurance. If quality assurance practices quality control, they will be directly involved in the line management operation of a project, and will be resented by both project management and the staff of that project.

- Establish ongoing quality improvement programs

A quality improvement program is designed to reduce the defect rate in the products or in the processes that produce them. Ongoing quality improvement programs are essential to the quality management environment, and should be initiated by the quality function (refer to the Act Process discussed in Skill Category 6). For example, if job control language syntax errors are found to be a high-defect product, start a quality improvement program to drive down the defects in the job control language.

There are four principles important to the success of a quality improvement program:

1. As management is responsible for the process, management must recognize the problem and initiate a quality improvement program to address that problem.
2. All functions, activities, and individuals involved in the defective product or process must be involved or represented in the defect solution. The final solution should be “owned” by the users of that solution, so these users must be involved in the solution process.
3. Managers and the quality improvement program participants should investigate the cause of all defects in the product or process under investigation, and attempt to eradicate that cause.
4. Managers and the quality improvement program participants must strive to price the cost of poor quality in order to demonstrate the value of the changes proposed by the quality improvement program.

Step 5: Staff and Train the Quality Function

The size of the quality function will vary with the size of the organization that it is supporting. Ideally QA analyst positions will be full-time staff positions. For this step, consider the following:

- Staffing

Over 90% of all personnel in IT quality functions have IT experience. IT skills are important, but other skills are equally important. The quality function should be staffed with the best possible people that have the experience and ability to achieve the responsibilities listed in the quality charter.

Larger quality functions are usually organized into areas of responsibility under a quality manager. For example:

- Quality Improvement Programs – area that develops recommendations to reduce the existing level of defects (usually a single defect at a time).
- Process (Standards) Development – creates and maintains the IT processes, and develops quality control procedures to help users of the processes with self-enforcement.
- Production Control – monitors the progress of applications during operations to ensure that they are being delivered on schedule. It may also control the program libraries.
- Training – responsible for the quality education of management, systems analysts, programmers, computer operators, and other technical staff.
- Error Tracking – area that analyzes operational problems to pinpoint correction responsibility and to categorize and quantify problems as a means of developing departmental-wide solutions for common problems.

The quality manager should view every member of the IT organization as a member of the quality staff. Many organizations encourage IT members to spend a few hours per week on quality activities, such as process definition and quality improvement programs. In addition, they will spend larger periods of time on quality control activities such as reviews, inspections, and testing.

The greater the involvement of the entire IT staff, the less the need for a large quality staff. As the quality function matures, the roles and responsibilities of the quality staff also change. The quality staff tends to be larger in a results-driven organization than it is in a manage-by-process organization.

- **Training QA Analysts**

Like all individuals in the organization, QA analysts should have a training plan with short and long-term objectives. To remain effective, QA analysts need to understand the technology and environment that they support as well as keep current on quality methodologies, techniques, and tools. Training methods include:

- On-the-job training, where the quality manager instructs quality personnel in how to perform their function.
- Attending quality assurance training courses and seminars, which are available from a number of sources, such as the Quality Assurance Institute, local quality chapters, on-line distance learning programs, etc.
- Interfacing with systems development methodologies, many of which explain how to perform quality control tasks.
- Studying related disciplines. Many disciplines are helpful in performing the quality function, such as business system design (emphasizes fact-finding), auditing (emphasizes fact-finding), control design (emphasizes detection and prevention of problems), individual engineering (emphasizes statistical methods), and statistical quality control.
- Preparing for a quality certification exam, which is a good review of basic quality principles and current quality-related topics. Obtaining the certification is also a good personal objective.

Step 6: Build and Deploy the Quality Toolbox

Much of the development of quality practices resulted from the Hawthorne Experiments involving Dr. Joseph Juran and Dr. W. Edwards Deming at Western Electric in the late 1920s and early 1930s. The practices evolving from that effort were the early tools used in industrial engineering. Many of these tools became associated with the practices of statistical process control.

Experience has shown that individuals need tools to properly evaluate their processes. For example, tools such as control charts help individuals determine whether a process is in control or out of control.

One of the most commonly used statistical tools is the checklist. This is a simple list of questions that enables people to follow or check processes, by indicating on the checklist the presence or absence of the occurrence of a particular event.

A quality toolbox is a set of tools, which assists in defining, controlling, and improving quality. The starter toolbox for a QA analyst contains the seven tools listed below. These tools and others are discussed later:

- Brainstorming
- Flowcharts
- Cause-and-effect diagrams
- Histograms
- Pareto charts
- Control charts
- Scatter diagrams

Step 7: Drive the Implementation of the Quality Management Environment

Implementing a quality management environment is a continuous journey, not a destination. David Kearns, past CEO of Xerox Corporation, stated after Xerox had won the MBNQA that the majority of the effort to create a quality management environment had not yet been done. His statements were echoed by CEOs of other winning companies who concurred with the very long-range aspects of quality.

A major reason to appoint a quality manager is to ensure that there will be a continuous focus on quality in the organization. To fulfill this objective, the manager must continually drive to improve the quality environment. This is a never-ending challenge that will consume a large portion of the quality manager's time.

To assist in driving and continually maturing the quality environment, it is recommended that the quality manager build the following into his/her annual plan:

- Attend at least one quality conference annually to stay abreast of the new quality practices
- Study in detail the criteria for winning the MBNQA (and other country or local quality award programs), particularly how the program changes from year to year
- Visit one to three other IT functions annually to informally benchmark against the organizations being visited
- Subscribe to quality literature to keep familiar with new quality theories and practices
- Obtain and maintain quality-related certifications to demonstrate proficiency in quality

IT Quality Plan

Quality planning is discussed in Skill Category 5. The IT Strategic Business Plan should include the following:

- The mission, giving a detailed description of what business IT is in
- Long-term IT goals giving direction for IT in the next five years
- Short-term objectives for the next business year
- How the objectives will be accomplished, including IT projects for the next business year
- Organizational renewal programs that will assure the long-range success of the organization
- The resources necessary to accomplish the short-term objectives and the organizational renewal activities that will enable the long-term goals to be achieved

An IT Quality Plan has two objectives:

- Supporting the organizational quality policy
- Ensuring the high quality achievement of the IT Strategic Business Plan.

The IT Quality Plan should include the following:

- A reference to the organization's quality policy
- An assessment of where the organization currently stands in relation to accomplishing the quality policy
- Long-term quality goals - these are discussed below
- Short-term quality objectives (i.e., programs) - these are discussed below
- The means of implementing the quality objectives
- Resources required in order to accomplish the short-term objectives and long term-goals

A major reason for the failure of quality initiatives is a lack of action. Many organizations have introduced quality principles through generalized education and other departmental-wide awareness and motivational sessions. However, at the end of these sessions nothing has changed. In fact, these sessions often demoralize the staff because they recognize the benefits that could be achieved if action was taken.

Quality is a long-term strategy, and any successful quality program must balance the long-term strategy of building a quality management environment with the short-term need for quick payback.

Long-Term Actions

The quality function should have a long-term plan of action to become a champion for moving the IT function to a world-class organization. While the short-term plan focuses on specific work tasks, the long-term plan is more complex and should incorporate these three objectives:

- Building a quality management environment
- Supporting the implementation of the IT function's quality policy
- Assisting management and staff in closing the two quality gaps (see Skill Category 1)

One of the best case studies in long-term planning is the Ford Motor Company. In a short period, from the late 1970s to the mid-1980s, Ford Motor Company went from losing hundreds of millions of dollars per year to becoming the most profitable corporation in the world. This happened under the guidance of Dr. W. Edwards Deming. His plan required Ford Motor Company to develop a mission, vision, guiding principles, and corporate values, and then live by them. Dr. Deming placed William Scherkenbach in the Ford Motor Company as the quality manager. His duties were to make sure that the management and staff of the Ford Motor Company were doing those things necessary to accomplish Ford's vision.

Short-Term Actions

Each quality manager must assess his/her own organization to identify quick-payback opportunities. The following short-term actions have proven beneficial in demonstrating that positive results can be achieved from quality initiatives.

Involve Management in Quality

The first of this two-part action is to help management get beyond quality as an abstract concept by educating them in the "how-to" aspects of quality management. The second part gets management involved in performing quality tasks, large or small. For example, management can post a paper on a bulletin board asking IT staff what prevents them from doing their job effectively, and then select tasks to perform from this list.

Redefine a Problem Process

Process definition should not take longer than just a few days to accomplish. In some organizations a small process can be defined in a day or two. Redefining a process should not usually take longer than five days.

It is best to select a process that is performed in a variety of different ways, none of which appears to be completely successful, and to select a team that consists of individuals with a vested interest in the process. The process definition team chooses the best pieces of all of the processes, and puts them together to create the best of the best practices within the newly defined process. The team reviews and evaluates the process based on the following criteria:

- Its value to the users (criticality of process)

- How current the process is
- Usability of the process
- Measurability of the process
- Attainability of adherence to the process in the existing environment

Find and Fix a Problem

As noted in Skill Category 1, the cost of quality in most IT functions is about 50%. This means half of the budget is already devoted to finding and fixing problems. Unfortunately, the fix is usually to the work product and not the process.

Measurement (see Skill Category 8) is essential to determine what problems need to be fixed, and is a prerequisite to improvement. It requires identifying a product, and then counting the number of defects, providing an accurate record of the types and frequencies of product defects. Analyzing this record identifies recurring defects and their root causes. Eliminating the root causes eliminates the defects, and results in process improvement.

Improve quality control

An important step in process definition is to identify the most logical points in the process to add a control. Skill Category 6 discusses this further.

"Overriding all other values is our dedication to quality. We are a market-driven institution, committed to our customers in everything we do. We constantly seek improvement and we encourage the unusual, even the iconoclastic."

Louis V. Gerstner, Jr., CEO, IBM Corporation

Quality Tools

A tool is defined as a vehicle that assists in performing a task. Some tasks that a quality management organization will be performing where quality tools can be used are:

- Defining a mission, vision, goals, and objectives
- Defining Do and Check processes
- Defining measures
- Collecting data
- Problem-solving
- Designing solutions
- Improving processes
- Measuring results

Quality tools can be categorized many different ways. For this presentation the following three groups have been selected. The tools described in each of these categories are a subset of existing tools. They have been included because they are more common and experience has demonstrated their effectiveness.

Management Tools

These tools are based on logic rather than mathematics, to address idea generation and organization, decision-making and implementation.

Statistical Tools

These tools have a mathematical focus, usually related to data collection, organization, or interpretation. They may also be separated into tools used for counting and tools used with measures.

Presentation Tools

These tools are used during presentations to summarize or graphically illustrate data. These tools may be used in the development of written materials, such as proposals or reports, or to accompany oral presentations.

The three steps needed to select and use a quality tool are:

1. Select the Tool

The general process for selecting a tool is to first define the objective (what is needed to perform the work task more effectively and efficiently). Next, study the tool description to determine whether the need objectives match the tool objectives. Finally, assure that the user of the tool believes it meets the objectives.

2. Learn the Tool

If applicable, the person using the tool must receive some training. Reading through the tool's documentation is the minimum. If possible, get classroom training or take a self-study course. Many tools are not only valuable in quality improvement, but can help individuals in the performance of their day-to-day work. Dr. W. Edwards Deming frequently stated that individuals knowledgeable in quality tools tend to be better on-the-job performers.

3. Use the Tool in Performing the Work Practice

The tool should be utilized in the manner in which it is taught. The user should ensure that there is an approach for deploying and using the tool, and that the results meet the objectives.

Management Tools

Tools in this category are used to help generate ideas and information, to organize the ideas or information, to facilitate making decisions about the information, and to aid in the

implementation. These tools are broad in nature. While they are not based on statistics, some, such as cause-and-effect diagrams and benchmarking may be used in conjunction with statistical tools. The tools to generate or organize ideas and information are:

- Brainstorming
- Affinity Diagram
- Nominal Group Technique
- Cause-and-Effect Diagram
- Force Field Analysis
- Flowchart and Process Map
- Benchmarking
- Matrix
- Quality Function Deployment
- Playscript

Brainstorming

Brainstorming is a technique used to quickly generate a quantity of creative or original ideas on or about a process, problem, product, or service. Brainstorming can be used to:

- Develop a vision
- Review inputs, outputs, and flows of existing processes
- Create a list of products or services
- Eliminate wasteful and redundant work activities
- Reengineer a process, product, or service
- Design a new or improved process
- Establish standards, guidelines, or measures
- Identify the internal and external customers served
- Improve the work environment
- Gather data for use with other tools

A brainstorming session begins with a facilitator establishing basic ground rules and a code of conduct. Typical brainstorming rules state that all members have an equal opportunity to participate, there is no criticism or pulling rank, people should think creatively, no idea will be treated as insignificant, and there should be only one conversation at a time. Members need to be active participants, willing to share their ideas, opinions, concerns, issues, and experiences.

Next the team agrees on the topic to be brainstormed and whether to give ideas verbally or written on individual index cards, or any other easily manipulated medium. Either a structured (round table) or unstructured (free-flowing) approach is selected. Ideas should be generated quickly (5-15

minutes) and are recorded clearly on a flipchart or board. The process stops when ideas become redundant or infrequent. Recorded ideas are reviewed for duplication and clarification, and eliminated when necessary. Remaining ideas are then evaluated with an open mind and may be used with the affinity diagram, nominal group technique, or cause-and-effect diagram.

Affinity Diagram

The affinity diagram is an orderly extension of a structured brainstorming session. Teams use this tool to help create order out of chaos, by categorizing large numbers of ideas. Rather than having teams react logically to a group of ideas, this technique helps to identify more creative solutions or to structure ideas for a cause-and-effect diagram.

Possible topics or problem statements where affinity diagrams could help are:

- Why policies don't exist
- Why standards are not adhered to
- Why QA failed
- Why objective measures aren't used
- Understanding the leadership role in quality management
- Why employees are not involved or lack empowerment
- Why quality doesn't work
- Improving teamwork in the workplace
- Understanding the issues to automation and use of CASE tools

To generate affinity diagrams, continue with these steps after a brainstorming session:

1. Write each idea on a separate index card.
2. Randomly place each index card on a flat surface, wallboard or flipchart.
3. In silence, team members move the cards into meaningful groups until consensus has been achieved (the group stops moving the cards).
4. As a team, discuss and then label each category with a title.
5. As a team, discuss each category, using cause-and-effect diagrams, if needed.

Nominal Group Technique

The nominal group technique is a structured, facilitated technique where all team members participate by individually ranking ideas, issues, concerns, and solutions, and then achieve consensus by combining the individual rankings. Sample ideas that could be ranked with the nominal group technique are:

- Which defect is the greatest?

- Who are our customers?
- What are our impediments to quality improvement?
- What new standards are needed?
- What are our key indicators?
- What tool is not being used effectively and how can we increase usage?
- How do we get a quality tool used?

Nominal grouping uses a round table (verbal) or index card (written) method for equal participation of teams or groups. It is a good technique to gather large amounts of information. The steps for the nominal group technique are:

1. Generate a list of ideas, issues, concerns, or solutions to prioritize. Brainstorming can be used if the list is not readily available.
2. If the list contains more than about 35 items, it may be desirable to shorten it using Pareto analysis to make it more manageable.
3. As shown in Table 6, record the remaining listed items in a location visible to the team, prefacing each item with a letter of the alphabet, such as:

A – list item 1

B – list item 2

C – list item 3

Table 6. Results from Nominal Grouping

List Items	Member 1	Member 2	Member 3	Total
A – item 1	2	3	1	6
B – item 2	1	1	2	4
C – item 3	3	2	3	8

4. Team members individually rank the list by assigning a number to each line item. One represents the lowest (least important) ranking, and higher numbers signify increasing importance.
5. Total the rankings of all team members. In this example, item “C” is the most important.

Cause-and-Effect Diagram

Teams use cause-and-effect diagrams to visualize, clarify, link, identify, and classify possible causes of problems in processes, products, and services. They are also referred to as "fishbone

diagrams," "why-why diagrams," or "Ishikawa diagrams" (after Kaoru Ishikawa, a quality expert from Japan).

Through understanding problems within the work processes, teams can identify root causes of a problem. A diagnostic approach for complex problems, this technique begins breaking down root causes into manageable pieces of a process. A cause-and-effect diagram visualizes results of brainstorming and affinity grouping through major causes of a process problem. Through a series of "why-why" questions on causes, this process can uncover the lowest-level root cause. Figure 18 displays a sample cause-and-effect diagram.

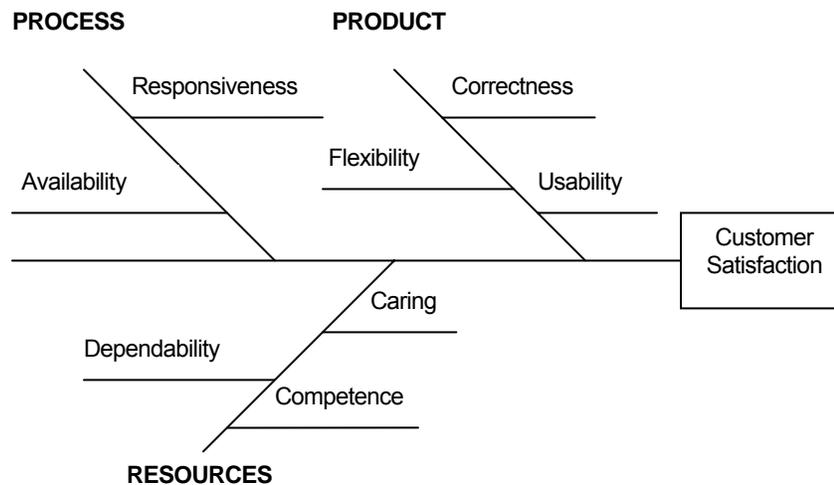


Figure 18. Cause-and-Effect Diagram

Cause-and-effect diagrams are applicable for:

- Analyzing problems
- Identifying potential process improvements
- Identifying sources of defect causes
- Analyzing improper use of test routines/testing problems
- Scheduling problems/cycle times

Developing a cause-and-effect diagram requires this series of steps:

1. Identify a problem (effect) with a list of potential causes. This may result from a brainstorming session.
2. Write the effect at the right side of the paper.
3. Identify major causes of the problem, which become "big branches". Six categories of causes are often used: Measurement, Methods, Materials, Machines, People, and Environment, but the categories vary with the effect selected.

4. Fill in the “small branches” with subcauses of each major cause until the lowest-level subcause is identified.
5. Review the completed cause-and-effect diagram with the work process to verify that these causes (factors) do affect the problem being resolved.
6. Work on the most important causes first. Teams may opt to use the nominal group technique or Pareto analysis to reach consensus.
7. Verify the root causes by collecting appropriate data (sampling) to validate a relationship to the problem.
8. Continue this process to identify all validated causes, and, ultimately the root cause.

Force Field Analysis

Force field analysis is a visual team tool used to determine and understand the forces that drive and restrain a change. Driving forces promote the change from the existing state to the desired goal. Opposing forces prevent or fight the change. Understanding the positive and negative barriers helps teams reach consensus faster. Following are sample processes that could benefit by a force field analysis:

- Implementing a quality function
- Implementing quality management in IT
- Developing education and training programs
- Establishing a measurement program/process
- Selecting a new technique or tool
- Implementing anything new
- Establishing meaningful meetings
- Empowering the work force

The steps below show how a team uses force field analysis:

1. Establish a desired situation or goal statement.
2. Brainstorm and list all possible driving forces.
3. Brainstorm and list all possible restraining forces.
4. Determine the relative importance of reaching consensus on each force.
5. Draw a force field diagram that consists of two columns, driving forces on one side and restraining forces on the other.

6. Select the most significant forces that need to be acted upon using the nominal group technique to prioritize and reduce the number, if there are too many.
7. Proceed to a plan of action on the forces selected in the previous step.

Flowchart and Process Map

A flowchart is a diagram displaying the sequential steps of an event, process, or workflow. Flowcharts may be a simple high-level process flow, a detailed task flow, or anywhere in between. They are standard tools for any IT organization. Flowcharts are most useful when applied by a team to obtain knowledge of a process for improvement. The technique is used to develop a common vision of what a process should do or look like. Once the process is documented, inefficiencies and redundancies can be identified and reduced.

A process map is a more detailed flowchart that depicts processes, their relationships, and their owners. The display of relationships and owners helps identify wasted steps, redundant tasks, and events with no trigger activities. Figure 19 shows a sample process map.

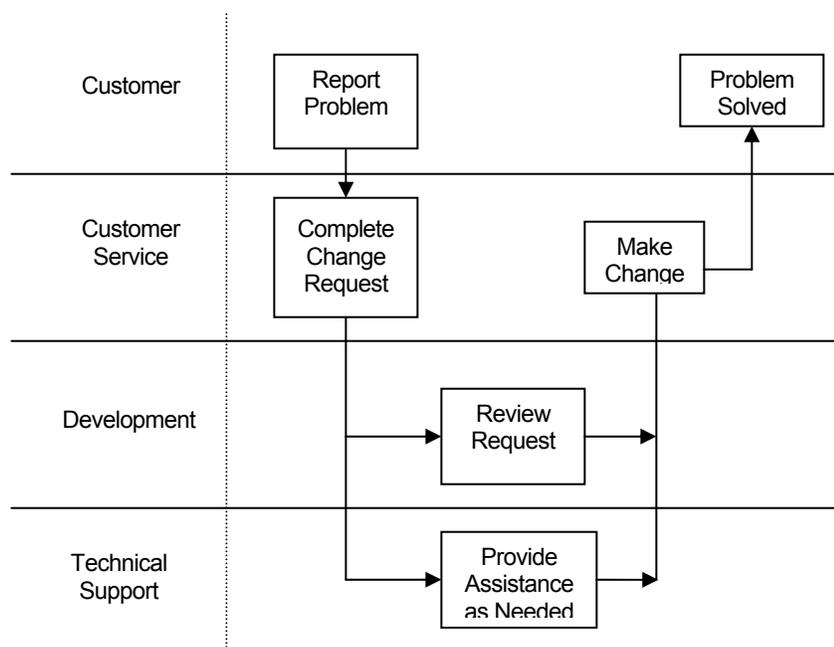


Figure 19. High-Level Process Map for Project Management

Sample processes where flowcharts are useful include:

- Life cycle activities, such as internal or external design, review processes, testing processes, change management, configuration management
- Customer surveys or interviews
- Supplier agreements or contracts
- Service level agreements

A flowchart is constructed using the following steps:

1. Identify the major function or activity being performed or needing to be performed.
2. Identify the tasks to support the function or activity.
3. Determine the steps needed to do the tasks.
4. Sequence the tasks and steps for the function or activity.
5. Connect the tasks and steps for the function or activity.
6. Create a flowchart of the tasks and steps for the function or activity using directional arrows or connections.
7. Reach team consensus on the process depicted in the flowchart.

Flowcharts should reference the following information:

- Process owners
- Suppliers
- Customers
- Key deliverables
- Decision points
- Interfaces
- Tasks and task sequence
- Policies
- Standards
- Procedures
- Tools used

Benchmarking

Benchmarking is the process of determining how well a company's products, services, and practices measure up against others. Benchmarking partners can be internal (against other company units), competitive (against competitors within the same industry), or functional (against "best in class" or "best in breed" within any industry). It is the highest level of performance that fully meets customer requirements.

Benchmarking enables a company to identify the performance gap between themselves and the benchmarking partner, and to determine a realistic improvement goal (some set higher goals) based on industry practices. It helps achieve process improvement, measurement, motivation and a management process for improvement. The use of benchmarking is not normally associated with

cost cutting or a quick fix. Benchmarking should be integrated with an organization's process improvement process.

Benchmarking has been used to:

- Evaluate and upgrade the customer requirements process.
- Design a professional career ladder for information technology professionals.
- Identify and install measurements for quality and productivity.

The three types of benchmarking are identified below. The first two types account for about 80% of all benchmarking that is done.

Process Benchmarking

This benchmark is used to plan for business process improvement, and is documented as part of business plans and quality improvement projects.

Product Benchmarking

This benchmark is used to help in product planning and development, using product documentation that includes the product performance goals and design practices identified through benchmarking.

Performance Benchmarking

This benchmark is used to set and validate objectives to measure performance, and to project improvements required to close the benchmark "gap."

Benchmarking is a ten-step process, involving four phases, as described below. Steps 2 through 5 are iterative.

Planning Phase

Step 1: Identify Benchmarking Subject and Teams

These internal or external candidates come from personal knowledge; interaction with industry groups; studying industry reports; and interviewing consultants, professional groups, etc.

Step 2: Identify and Select Benchmarking Partners

Determine viable candidates for benchmarking; obtain their agreement to participate; and confirm the visit time, agenda, and attendees with the benchmarking partner.

Step 3: Collect Data

Document the organization's process to be benchmarked. Develop questions and meet with the process owners to collect and record the data.

Analysis Phase**Step 4: Determine Current Competitive Gap**

Identify the difference between the attributes of the organization's process, product, or performance and those of the benchmarking partner.

Step 5: Project Future Performance Levels

Based on the competitive gap, make a managerial decision regarding performance goals for the organization.

Integration Phase**Step 6: Communicate Findings and Gain Acceptance**

Describe the benchmarking results to the process owners and involved parties, and communicate the potential future performance levels to gain acceptance to move forward.

Step 7: Establish Functional Improvement Objectives

In conjunction with the process owners, establish specific improvement objectives to be achieved. These are generally short-term objectives not to exceed one year.

Action Phase**Step 8: Develop an Action Plan**

Plan improvement using the organization's process improvement process.

Step 9: Implement Plans and Monitor Progress

Perform the plan, measure progress, and make adjustments as necessary.

Step 10: Recalibrate and Reset Benchmark Performance Levels

Based on the analysis, set new objectives, benchmark again to find better ways of executing the process, and continue the improvement cycle.

Lessons learned from the benchmarking leaders include the following:

- Focus on a specific objective and process, and facilitate the benchmarking session to keep it on track.
- Prepare objectives, agenda, data, attendees, meeting and benchmarking protocol, and process documentation requirements in advance.
- It is not easy to identify the IT “best of the best” because good performance data is not readily available, and research is required to evaluate opinion versus fact.
- It always takes longer than you think.

Matrix

A matrix is a structured, problem-solving technique used to show the relationship between groupings. It is also known as a matrix check sheet or a matrix diagram.

The matrix data analysis is frequently used to identify whether customer needs are being met, not being met, different, or no longer exist. For IT organizational teams, this approach could support the determination of system requirements, such as when teams need to understand and analyze customer preferences to drive out requirements or improvements on a product or service. This tool helps view a problem as a whole for clarity, especially in conjunction with the JAD process.

For multi-dimensional problems, this approach focuses on the essential factors in problem areas. It helps teams sort language information for discussion and consensus, focus on what is important, and clarify the strength of the relationships. The matrix data analysis allows a team to test, evaluate, and develop strategies of multi-dimensional factors in solving problems.

Matrix diagrams can be used to:

- Research or survey customer preferences
- Compare skill levels versus experiences in job
- Evaluate tools available versus usage
- Correlate defect rates, cycle times, effort, and skills
- Understand tasks in a process versus goals and resources

The two common types of matrices are the L-type matrix and the T-type matrix. An L-type matrix compares two sets of items to each other or compares a single set to itself, such as two characteristics of a process or product. A T-type matrix is used to compare two sets of items to a common third set, such as observed attributes between causes and results.

Table 7 is an L-type matrix, showing attributes of an improvement objective. The relationship between the attributes and objectives helps clarify how to prioritize the objectives.

Table 7. L-type Matrix

Improvement Objective	Contribution (1-5)	Readiness (1-5)	Capability (1-5)	Cost/Benefit (1-5)	Score
Implement Unit Testing	5	5	1	3	14
Define Coding Standards	3	1	4	2	10
Implement Design Reviews	5	5	5	3	18
Build Usability Lab	2	1	1	4	8

To produce an L-type matrix, use the following steps:

1. Determine the (usually) two lists of items to be compared.

2. Create a matrix (tabular) structure with enough rows and columns to accommodate the two lists. Put one list across the top of the matrix and one down the left side.
3. Determine how the comparison will be symbolized. Typically this is shown with numbers or with relationship symbols such as shaded circles, circles and triangles (indicating strong, probable, none).
4. Complete the matrix by putting the relevant symbols in the corresponding boxes.
5. Total the scores if applicable.

With the exception of the format, a T-type matrix is generated the same as an L-type matrix. For the T-type matrix, the common set of items is listed in a row across the middle of the matrix. Listed along the top half of the left side is one set of items (such as causes) and listed along the bottom half of the left side is the other set of items (such as results). The resulting matrix is in the shape of a “T”.

Quality Function Deployment

A quality system is an organized approach to quality with tools, techniques and a set of methods. Dr. Yoji Akao, principal developer of Quality Function Deployment (QFD), defines QFD as a quality system with many components as shown in Figure 20. Comprehensive quality deployment includes quality deployment, technology deployment, cost/schedule deployment, and reliability deployment. It can also address other special concerns with a corresponding deployment, such as usability, reuse, security, etc. QFD provides forward and backward traceability of value in the software development life cycle.

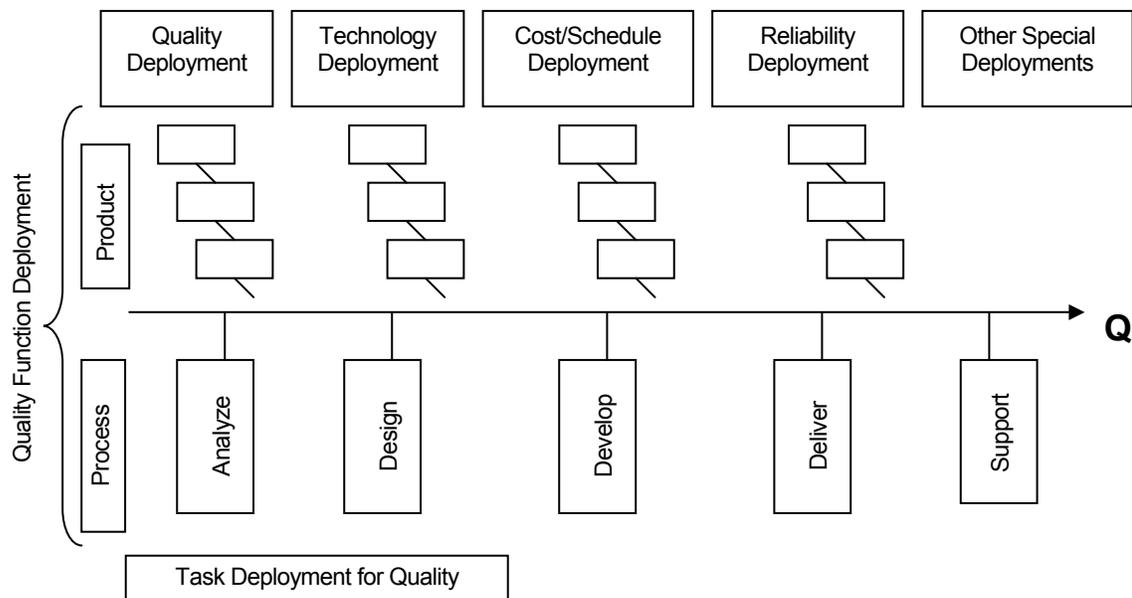


Figure 20. QFD Defined

A special challenge exists for complex products or services involving a combination of hardware, software, and service; or those where nontrivial design decisions must be made at the system, subsystem, and component levels. In QFD there are a series of matrices that comprise a visible means to address a particular concern during development, such as reliability. These deployments are a structured way of dealing with a special concern at a detailed level, and provide a basis for linking the concerns into an integrated framework. The result is a very sophisticated (and efficient) product development process as shown in Figure 21.

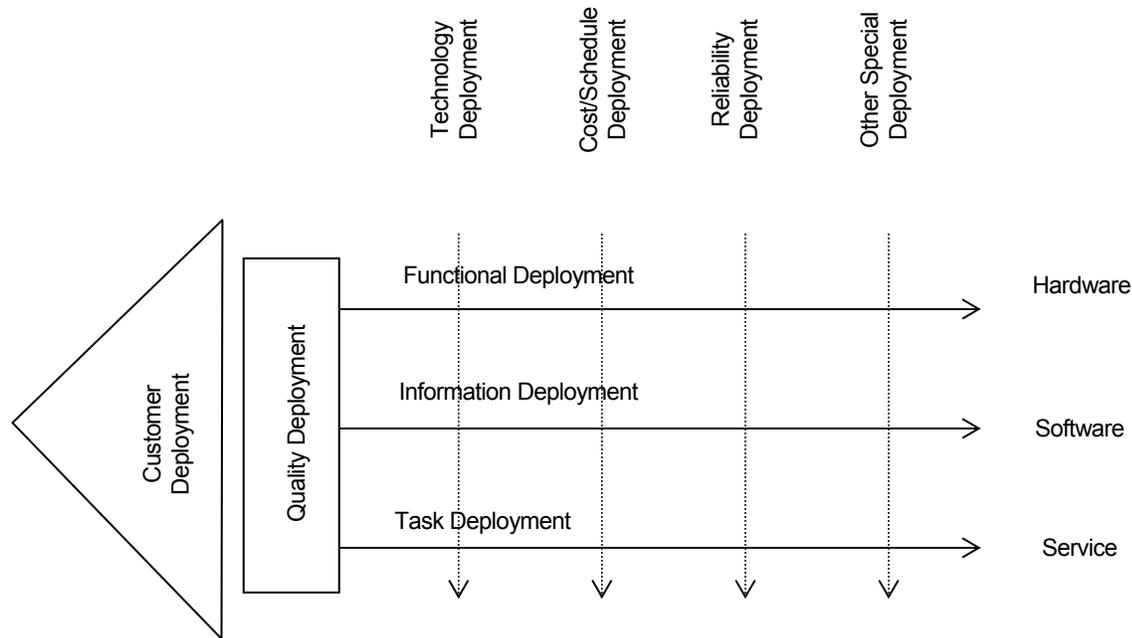


Figure 21. QFD Deployments

QFD covers all special concerns of the software organization and its customers with fundamental horizontal and vertical deployments:

Fundamental Deployments

The starting point in new product development is the decision of what to build for whom. This requires determining customers, finding out what they want, and identifying what capabilities can be provided to them. In QFD, the fundamental deployments of customers and quality address this.

- **Customer Deployment** involves determining which types of customers for which the organization is trying to provide a product or service. It precedes the determination of requirements in quality deployment. Customers must be identified first in order to know to which voices to listen. The customer deployment component of QFD is particularly important for software, as a single software product must often satisfy many types of customers.
- **Quality Deployment** has tools and techniques for the exploration and specification of high-value customer requirements (or "demanded quality"). Once captured, the customer

requirements are translated and deployed into technical requirements (or “quality characteristics”) in the House of Quality matrix. (The House of Quality is the first matrix that a product development team uses to initiate a Quality Function Deployment (QFD) process.) This can be done at various levels of sophistication, ranging from four matrices to a dozen.

These fundamental deployments are the foundation for downstream design-and-development decisions about “how” the product will work (the horizontal deployments) and 'how well' it will be designed and developed (the vertical deployments). Both horizontal and vertical deployments are required to appropriately handle this complexity.

Horizontal Deployments

There are three basic horizontal deployments that follow from the customer and technical requirements.

- **Functional Deployment** drives the design of the hardware aspects of the product or service. Often this involves determining the functions, mechanisms, components, and materials for a hardware product, such as medical equipment. This is where value analysis and value engineering enter the QFD process. For hardware products, the technical capabilities must be deployed into functions. The functions are then deployed into mechanisms, components, and materials.
- **Information Deployment** handles the information (or data and processing) aspects of the product or service. These aspects are not necessarily automated, but can be automated with software, such as patient medical records. For software-intensive products or services, the technical capabilities must be deployed into data and processing, which is all that software does. In more modern software engineering languages, these are objects with encapsulated data and methods.
- **Task Deployment** addresses the detailed activities required to satisfy the customers, such as professional patient care. The task deployment component of QFD includes value analysis techniques (such as the quality systems diagram and the quality activities table) to identify and make visible the key activities of a service or development process. This is fundamental to QFD for services.

A combined example would be a training class. The training room environment and physical course materials would be developed via functional deployment. The course content would be developed via information deployment, and the delivery approach would be developed with task deployment. Each deployment utilizes a particular set of QFD tools and techniques.

Vertical Deployments

There are often organizational concerns that apply to the hardware, software, or service aspects of the total product, or to overriding concerns of the customers that can only be effectively addressed throughout design and development. To deal with these types of concerns, a series of specialized vertical deployments exists to visibly deploy defined concerns across the horizontal deployments. There are three common, vertical deployments to address technology, cost and schedule, and reliability. An additional category of “other deployments” is constructed to address special concerns. Vertical deployments should be tackled after the project team has mastered the

fundamental quality deployment component of QFD, and the appropriate horizontal deployment(s).

- **Technology Deployment** seeks to systematically and rapidly deploy new technologies into the design and development of new products or services. High-tech organizations are keenly interested in leveraging their research and development strengths into new, technically advanced products and services. The technology deployment component of QFD provides a systematic way to accomplish these aims.
- **Cost and Schedule Deployment** sets customer-derived cost and schedule targets and seeks the necessary adjustments during product development to meet those targets. Cost reduction is often applied to hardware, where the costs of materials and manufacturing dominate. For software and services, costs derive primarily from the person-hours expended, so schedule reduction is more common. This deployment component of QFD provides a systematic framework for improving the speed of the software development process so that it is (eventually) capable of meeting customer-set schedules.
- **Reliability Deployment** is a systematic way of looking to failure modes and faults to prevent or ameliorate the effects of failures, and design in reliability. Standard reliability engineering tools and techniques are used throughout design and development.
- **Other Special Deployments** are used to address additional concerns of customers, stakeholders, or the development organization. For example, in some aerospace projects, weight or mass deployment is used to deal with payload constraints. In some embedded software projects with tight memory constraints, memory deployment is used.

Any and all of these deployments may be required for a complex product. Together, the fundamental deployments of customer and quality, the horizontal deployments, and the vertical deployments provide an integrated framework for thorough attention to all customer-critical aspects of software development.

Dr. Deming said development must be viewed as a system. How an organization satisfies its customers must be examined. Software QFD is one quality system that aims to deliver high-quality, software-intensive products and services to multiple types of customers. This approach has been applied by a number of leading software firms in Europe, Japan, and North America. Results to date are very promising, and further refinement is still occurring.

Playscript

A playscript is a document that defines the complete requirements or procedure for executing a selected segment of work, and for coordinating between work segments. As its name implies, this tool is written following the format of a play. The action steps are written in the present tense, in a logical time sequence. Each action step notes who does what, and who receives what. Actions are written concisely, avoiding excess verbiage, details, opinions, etc.

Playscript is easy to write. Using it is advantageous because it helps the author think in terms of a logical work sequence, resulting in procedures that are easy to read. It reveals any duplicated actions or unnecessary steps. The format ensures individual or department roles are clear, as are

relationships and connecting work sequences between departments. The simplicity and directness of the format helps people focus on the work (not the words), and facilitates making changes.

Statistical Tools

The tools covered in this section are used to collect, view, and analyze numbers. They are:

- Check Sheet
- Histogram
- Pareto Chart
- Run Chart
- Control Chart
- Scatter Plot

Check Sheet

A check sheet (also called a checklist or tally sheet) of events or occurrences is a form used to gather and record data in an organized manner. This tool records the number of occurrences over a specified interval of time to determine the frequency of an event. The data is recorded to support or objectively validate the significance of the event. It may follow a Pareto analysis or cause-and-effect diagram to validate or verify a problem, or it may be used to build Pareto charts or histograms. Figure 22 shows a sample check sheet.

(Daily System) Failures	Week of dd/mm/yy					
	Day 1	Day 2	Day 3	Day 4	Day 5	Total

Figure 22. Check Sheet

Check sheets can be used to record the following types of information:

- Project review results, such as defect occurrences, location, or type
- Documentation defects by type or frequency
- Cycle times, such as requirements to design or design to implementation
- Conformance to standards
- End user complaints of all types
- End user surveys
- Late deliveries

To use a check sheet:

1. Clarify what must be collected objectively.
2. Establish a format for the data collection that is easily understood by the collector.
3. Ensure those involved understand the objectives so the collection process is accurate.
4. Establish the sample size and time frame of data collection.
5. Instruct or train data collectors for consistency.
6. Observe, record, and collect data.
7. Tally the results.
8. Depending on the purpose, build a Pareto chart or histogram or evaluate the results to determine whether the original analysis is supported.

Advantages of check sheets are that they pre-define areas to discuss, limit the scope, and provide a consistent, organized, and documented approach. Disadvantages might be their applicability or limiting of other questions.

Questions on check sheets should be organized by topic and tested prior to use. A response of “I don’t know” should be allowed for, and bias should be avoided. The person using the check sheet should understand the reason for the questions and be able to anticipate a response.

Histogram

A histogram (or frequency distribution chart) is a bar graph that groups data by predetermined intervals to show the frequency of the data set. It provides a way to measure and analyze data collected about a process or problem, and may provide a basis for what to work on first. Histograms are also useful for displaying information such as defects by type or source, delivery rates or times, experience or skill levels, cycle times, or end user survey responses. Figure 23 shows a simple histogram.

Interval	Tabulation	Frequency	Cumulative Frequency
0-3	11	2	2
3-6	111111	6	8
6-9	111	3	11

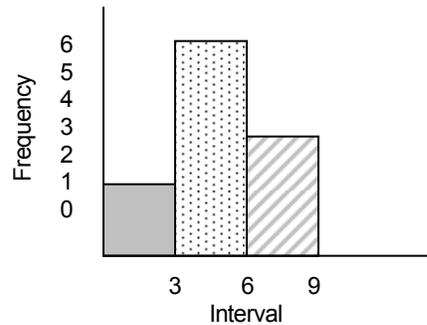


Figure 23. Histogram

A histogram requires some understanding of the data set being measured to consolidate or condense it into a meaningful display. To create a histogram, perform the following steps:

1. Gather data and organize it from lowest to highest values.
2. Calculate the range, which is the largest value – smallest value.
3. Based on the number of observations, determine the number of cells, which is normally between 7 and 13.
4. Calculate the interval or width of the cells, which is the range divided by number of cells.
5. Sort the data or observations into their respective cells.
6. Count the data points of each cell (frequency) to determine the height of the interval, and create a frequency table.
7. Plot the results.

The distribution is normally a bell-shaped curve. Other shapes such as double peak, isolated island, cliff, cogwheel, and skewed can provide insight on the variability of the process.

One variation on the histogram is to create a graph by drawing a line from the midpoints of the bars. Then add the range of acceptable values (e.g., within plus or minus 5 of budget) to show whether the actual values lie within the acceptable range.

Pareto Chart

The Pareto chart is a special type of histogram, used to view causes of a problem in order of severity from largest to smallest. It is a simple statistical tool that graphically shows the 20-80 rule where 20% of the sources cause 80% of the problems. Joseph Juran refers to this Pareto principle as the separation of the “vital few” from the “trivial many”.

A Pareto chart is typically used early in the continuous improvement process when there is a need to order or rank problems or causes by frequency. The vital few problems and their respective root causes can then be focused on. This technique provides the ability to:

- Categorize items, usually by content (type of defect, position, process, time, etc.) or cause (materials, operating methods, manpower, measurement, etc.) factors
- Identify the causes or characteristics that contribute most to a problem
- Decide which basic causes of a problem to work on first
- Understand the effectiveness of the improvement by doing pre- and post-improvement charts

Sample problems for Pareto analysis include:

- Problem-solving for the vital few causes or characteristics
- Defect analysis
- Cycle or delivery time reductions
- Failures found in production
- Employee satisfaction/dissatisfaction

The process for using Pareto charts is described in the following steps:

1. Use brainstorming, affinity diagrams, or cause-and-effect diagrams to define the problem clearly.
2. Collect a sufficient sample size (at least 30 occurrences) of data over the specified time, or use historical data, if available.
3. Sort the data in descending order by occurrence or frequency of causes characteristics.
4. Construct the Pareto Chart and draw bars to correspond to the sorted data in descending order, where the “x” axis is the problem category and the “y” axis is frequency.
5. Determine the vital few causes to focus improvement efforts.
6. Compare and select major causes, repeating the process until the problem’s root causes are reached sufficiently to resolve the problem.

Run Chart

A run chart as shown in Figure 24 is a graph of data, in chronological order that displays changes and trends in the central tendency (average). The data represents measures, counts, or percentages of outputs (products or services) from a process. Run charts are often used to monitor and quantify process outputs before a control chart is developed. Run charts can be used as input for establishing control charts.

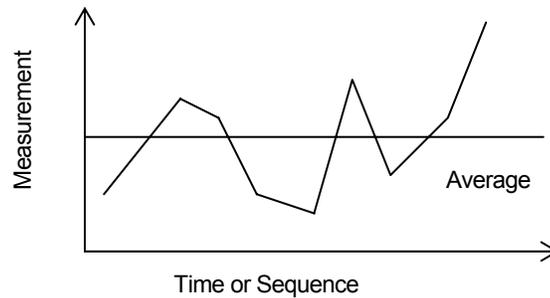


Figure 24. Run Chart

Run charts can track events such as:

- Total failures
- Complaint levels
- End user satisfaction levels
- Suggestion levels
- Training efforts
- Production yields
- Number of invoices
- Number of system errors
- Down time (minutes, percent)

The steps for developing a run chart are as follows:

1. Decide which output of a process to measure.
2. Label the chart both vertically (quantity) and horizontally (time).
3. Plot the individual measurements over time (once per time interval or as they become available), tracking the data chronologically in time.
4. Connect data points for easy use and interpretation.

5. Monitor the data points for any obvious trend.

Control Chart

Control charts provide a way of objectively defining a process and variation. They establish measures on a process, improve process analysis and allow process improvements to be based on facts. Note: variation is briefly described here to put control charts in perspective. Skill Category 8 provides additional details on the topic of variation.

The intent of a control chart is to determine if a process is statistically stable and then to monitor the variation of stable process where activities are repetitive. There are two types of variation:

- Common or random causes of variation

These are inherent in every system over time, and are a part of the natural operation of the system. Resolving common cause problems requires a process change.
- Special causes of variation

These are not part of the system all the time. They result from some special circumstance and require changes outside the process for resolution.

Common causes of variation are typically due to many small random sources of variation. The sum of these sources of variation determines the magnitude of the processes inherent variation due to common causes. From the sum, the process control limits and current process capability can be determined. Accepted practice uses a width of three standard deviations around the population mean ($\mu \pm 3\delta$) to establish the control limits. A process containing only common causes of variation is considered stable, which implies that the variation is predictable within the statistically established control limits.

Processes containing special as well as common causes of variation are referred to as unstable processes. Figure 25 and Figure 26 show control charts for stable and unstable processes. Note: the special cause falls outside of the control limits of the unstable process.

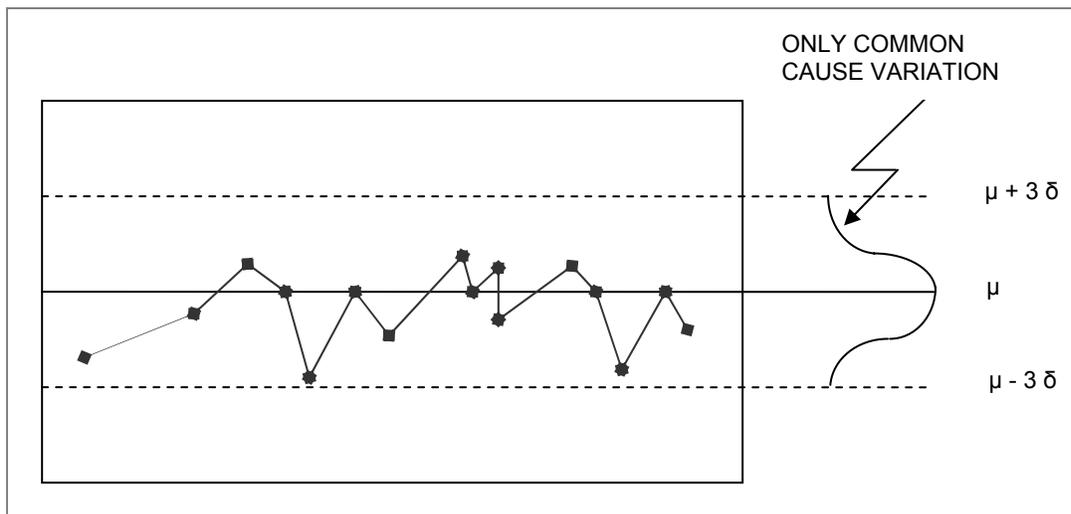


Figure 25. Control Chart of a Stable (In Control) Process

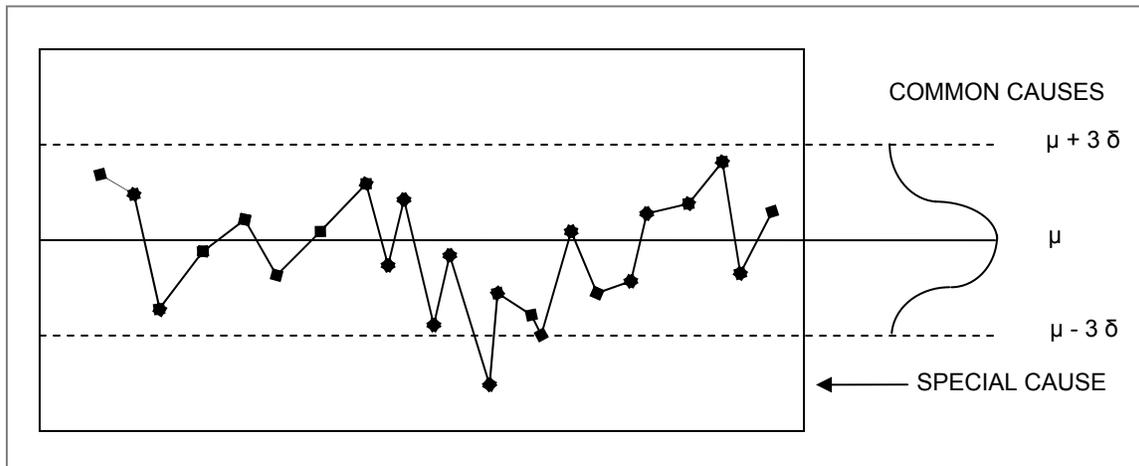


Figure 26. Control Chart of an Unstable (Out of Control) Process

Control charts are suitable for tracking items such as:

- Production failures
- Defects by life cycle phase
- Complaint/failures by application/software
- Response time to change request
- Cycle times/delivery times
- Mean time to failure

When there is reason to believe a process is no longer stable, it is typically evaluated first by brainstorming, Pareto analysis, and cause-and-effect diagrams. Use of control charts follows:

1. From the initial evaluation, identify the characteristics of the process to monitor, such as defects, cycle times, failures, cost, or maintenance.
2. Select the appropriate type of control chart based on the characteristic to monitor. Data that is variable (measured and plotted on a continuous scale such as time, cost, figures, etc.) may require different charts.
3. Determine the methods for sampling, such as how many or over what time frame.
4. Collect sample data. Check sheets can be used to gather data.
5. Analyze and calculate the sample statistics: average, standard deviation, upper control limit, and lower control limit.
6. Construct the control chart based on the statistics.

7. Monitor the process for common and special causes of variation. The process is in control when observations fall within the control limits.

Five rules are used to determine the existence of special causes. If observed, the process needs to be evaluated and analyzed for causes related to the situation. The five rules constituting a special cause are:

- Any point outside the upper or lower control limit.
- Any run of eight or more data points above or below the average value (centerline), indicating the average has changed.
- Six or more consecutive data points, which are increasing (trend up) or decreasing (trend down).
- Two out of three consecutive points in the outer one-third control limit.
- Fifteen consecutive points between the centerline and inner one-third of the chart.

Scatter Plot

A scatter plot is used for problem solving and understanding cause-and-effect relationships. It shows whether a relationship exists between two variables, by testing how one variable influences the response (other variable). Scatter plots are also called scatter diagrams or correlation diagrams.

Scatter plots may be used to look for relationships, such as:

- Defect Level versus Complexity
- Defects versus Skill Levels (Training)
- Failures versus Time
- Cost versus Time
- Change Response versus People Availability
- Defect Cost versus Where Found (Life Cycle)
- Preventive Cost versus Failure Cost

The steps for creating scatter plots are:

1. Select the variable and response relationship to be examined.
2. Gather data on the variable and response; determine the sample size of the paired data.
3. Plot the results; determine the appropriate scale to plot the relationship.
4. Circle repeated data points as many times as they occur.

5. The pattern of the plots will suggest correlation: positive, negative, random, linear, curvilinear, cluster, etc.

Figure 27 shows a few typical patterns. Be careful when interpreting results – a frequent error in interpreting results is to assume that no relationship exists between a variable and a response because a relationship isn't immediately apparent. It may be necessary to take additional samples, or use specialized axes such as logarithmic scales.

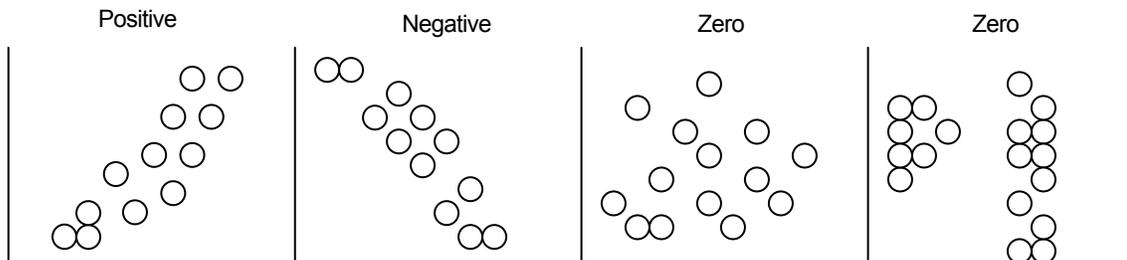


Figure 27. Types of Scatter Plots

Presentation Tools

Presentations are an integral part of the change process. The involved parties, sometimes called stakeholders, need to be convinced that a proposed change is beneficial, or want to see reports during and after implementation. Stakeholders include management, the individuals that will use the changed process, and the individuals impacted by the changed process.

The five tools below are the more common methods for graphical presentation:

- Table
- Line Chart
- Bar Chart
- Pie Chart
- Stem-and-Leaf Chart

Table

Quality reports often use tables as worksheets presented to management. The information is presented in row and column format. Spreadsheet software can prepare these types of graphical presentations.

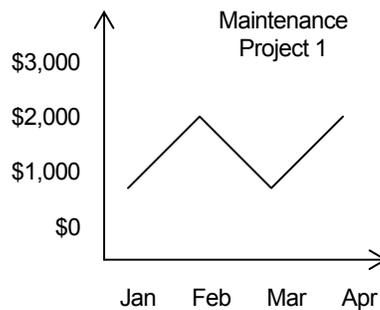
Table 8 shows a sample table, which depicts the dollars spent on maintenance for three different projects of about equal complexity over a four-month period.

Table 8. Sample Table

Month	Project 1	Project 2	Project 3
January	\$1,000	\$2,000	\$2,000
February	\$2,000	\$1,000	\$3,000
March	\$1,000	\$2,000	\$1,000
April	\$2,000	\$3,000	\$3,000
Total	\$6,000	\$8,000	\$9,000

Line Chart

A line chart is used to show direction of events. For example, Figure 28 shows how maintenance costs for Project 1 fluctuate over time. Line charts can also be used to compare:

*Figure 28. Line Chart*

- Like Units – There could be a line for each of the three projects.
- Related or Fixed Variables - The total or average maintenance could be shown as another line on the chart.
- Like Periods - Maintenance for Project 1, for the first four months of this year could be compared to the same time period last year.

Bar Chart

A bar chart is normally a two-dimensional chart using bars to represent values or items. In Figure 29, project 3 maintenance costs are illustrated. Note that the same type of information can be presented in a tabular format, a line chart, or a bar chart.

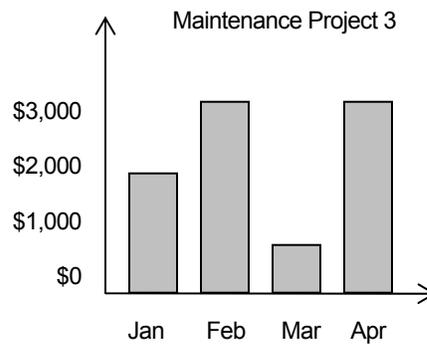


Figure 29. Bar Chart

A bar chart is particularly advantageous when:

- A large graphic is desired.
- Specific time frame periods are to be emphasized.
- Two or more items are to be included, representing different values or items so that the bar permits easy distinction between the two items.

Pie Chart

A pie chart graphically presents the components of a total population. The pie chart illustrated in Figure 30 uses percentages to show how four categories of information are spread among the population. The size of each pie segment should reflect the portion of the total represented by that piece. It is not necessary to be highly precise if the numbers are placed within the pie.

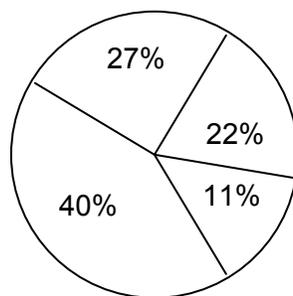


Figure 30. Pie Chart

Pie charts can be used to view:

- Segments visually differentiated from one another.
- Segments showing the percent of the whole (uses 100% of the total pie).

- Dollar volumes, where each pie piece indicates how many dollars of the total dollars are included.
- Items, where each piece shows the number of items, such as claims processed by the processing district.

Stem-and-Leaf Chart

The stem-and-leaf chart is a variation of the bar chart using the actual distributed values shown by category. Since the actual values are used, the stem-and-leaf chart is only practical when the absolute number of values is low (usually 100 or less).

Figure 31 shows a stem-and-leaf analysis of the cost per pound to acquire product X. The vertical line shows the categories on the left and the absolute values on the right.

Cost per Pound of Product X	
\$.00-\$.99	95
\$1.00-\$1.99	07, 17, 23, 49, 69, 80, 81, 82, 83, 86, 90
\$2.00-\$2.99	01, 01, 01, 02, 03, 04, 06, 09, 71
\$3.00-\$3.99	02, 07
\$4.00-\$4.99	43

Figure 31. Stem and Leaf Chart

In the values shown, the high-order digit of the cost per pound is eliminated. For example, in the zero category the 95 represents \$.95 per pound, while in the “1” category the value 07 represents \$1.07 per pound. Using this graphic, the actual values as well as the shape of the population represented by the string or bars of absolute value can be visualized.

Process Deployment

One of the most difficult tasks facing any IT function is changing the way that function operates. In most organizations, stability is the norm and change is abnormal. That cycle needs to be reversed, if quality and productivity are to be constantly improved.

People resist change for the following reasons:

- It is significantly more difficult to implement change than to develop the approach that will cause the change.
- People do not like change imposed on them. If they are not actively involved in making the change, there is a natural resistance because the change is not his or her idea. (This is closely associated with the “not-invented-here” syndrome.)

- Workers know they can be reasonably successful using the current process, but not whether they can be successful using the changed process. Change brings risk, and there is a higher probability that they will make errors using it.
- When people spend time and effort to learn a new process, and then make routine mistakes while learning, it discourages them from wanting to try the changed process.
- The person(s) who developed the current process may resent having that process changed. This resentment sometimes leads to overt action to stop the change.
- Management may be more committed to meeting schedules and budgets than to implementing change.

Getting Buy-In for Change through Marketing

The marketing process is one of integrating solutions into an environment. As a result, marketing is an important component of the approval and the deployment processes. Marketing tactics begin before preparation of a formal proposal and continue through acceptance. They are also a major part of the acceptance step, which is discussed in Step 2 of the “Deployment Phase 3: Tactical” on page 219.

Figure 32 illustrates the relationship between the PDCA cycle and a five-step marketing process.

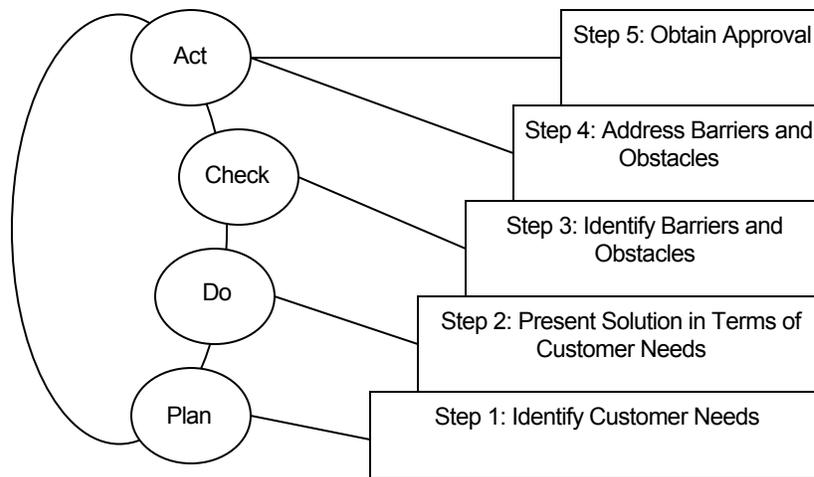


Figure 32. Marketing Tactics --A Five-Step Process

The five-step marketing process is as follows:

Step 1: Identify Customer Needs

In this step of information gathering, the customer’s needs are differentiated from wants, and the needs are validated as true needs. Needs are what the customer requires to do his/her job

effectively. Wants are the requirements defined by the customer. Ideally, they will be the same, but it cannot be assumed that what the customer says is wanted is what is really wanted.

Customer needs are typically determined by any combination of customer surveys, user-developed statement of requirements, customer process decomposition, data flow diagrams, JAD sessions, and prototyping. They then are confirmed with the customer.

While confirming the requirements, effort is undertaken to win as many supporters as possible, so that when the formal proposal is made, it has already been approved and the ceremony is a formality rather than a selling event.

Step 2: Present Solution in Terms of Customer Needs

People buy benefits, not products (i.e., deliverables). Thus, in this step the deliverables or solutions are defined in terms of showing benefit to the customer in a manner understandable to the customer. For example, end users do not need client-server technology; they need the capabilities provided by client-server technology. It must be demonstrated how these benefits are aligned with organizational goals. They must be packaged to match the organization's and decision-maker's style and then marketed to the customer (emphasizing the value of the proposal and importance to them).

Step 3: Identify Barriers and Obstacles

Customer objections such as lack of resources, schedule length, difficulty to build, or not meeting management approval criteria are a normal part of the marketing process. They should be viewed positively, because objections usually show interest. Objections fall into three general categories:

- **Formal Objections**

These objections are those voiced by the customer or potential customer about the solution to a need. The challenge is to ensure that the objections are the correct ones, not substitutes. For example, a decision-maker who does not want to implement empowerment might not voice personal objections, but, substitute objections like lack of time, resources, or priorities. Once an objection has been validated as real, it can be addressed through compromise or alternative strategies.

- **Organizational Barriers and Obstacles**

These objections are internal, such as procedures that inhibit change within an organization, or conflicting policies. Plans are often prepared without identifying organizational barriers and obstacles. Root causes should be determined to understand why these barriers exist. Then the most effective way of addressing them should be incorporated into tactical plan tasks.

- **People Barriers and Obstacles**

These objections come from stakeholders. Marketers should identify stakeholders, their stake in success or failure, and the reasons that support this stake so that selling efforts can focus on their known or unknown objections.

Step 4: Address Barriers and Obstacles

Once an objection occurs, the key to the sale is overcoming the objection. For example, if management says they cannot buy a tool because there is no money available, finding a way to transfer money from an end user budget to the information budget is all that is needed to gain approval to acquire the tool. It is generally poor business practice to request approval for a project to which objections have been raised but not addressed. If this happens, the objections normally occur again during the approval process or during implementation of the project. People disliking the proposal may give lukewarm support to the proposal during the early stages, but will make their objections known if they detect a weakness. Objections can be expressed formally, or can occur through intentionally performing the new activities ineffectively.

It is more cost-effective to address objectives before implementation, and the probability of a successful project is significantly higher. After recognizing that there is an objection, assess its magnitude and root cause (such as ineffective, personal preference). Address the objection quickly and forcibly. Too many objections may indicate that the proposal is probably not aligned with the customer or organization.

Step 5: Obtain Approval

In marketing, this is called “closing the sale,” and is generally done as early in the process as possible. For example, if approval can be obtained when defining the need, do not continue with the remaining steps. Management may be ready to give approval, but staff members who have prepared a detailed proposal and marketing strategy may want to present their material. Once management begins to understand barriers and obstacles they may change their mind.

If the first four steps have been adequately performed, approval should be an automatic process. If approval is not obtained, then the execution of steps one through four should be carefully examined to determine the root cause of disapproval. While it may not be possible to reverse this decision, lessons learned could improve the marketing process to ensure approval on the next proposal.

The Formula for Effective Behavior Change

Industrial psychologists state that the formula for behavior change is:

$$\text{Behavior} = \text{Individual} + \text{Environment}$$

This formula indicates that if neither the individual nor the environment changes, there is no change in behavior. It is significantly easier to change the environment in which an individual operates than to change an individual’s attitudes and beliefs.

The Deployment Process

Initiating change is only effective when change is implemented through a process. This change process is called deployment. Deployment is the process of implementing a new or improved approach to completing a work task.

Deployment is normally only effective in an environment that uses processes. If there are no processes, there is no way of effectively implementing a change. In a software engineering environment, compliance to process is enforced, thus deployment is a critical component of making the software engineering environment work.

Dr. Curt Reimann, past director of the U.S. National Quality Award Program, stated that less than one percent of U.S. Corporations have an effective quality program. Most quality experts believe the cause of ineffective quality programs is attributable to poorly designed or nonexistent deployment efforts, coupled with the lack of measurement processes to assess the results of quality programs. Starting a quality management program forces management to rethink its quality responsibilities.

There are three deployment phases - assessment, strategic, and tactical. The assessment and strategic deployment phases represent the Planning component of the PDCA cycle. The tactical deployment phase represents the Do, Check and Act components of the PDCA cycle.

Deployment Phase 1: Assessment

The first step in the deployment process is to establish the current level of performance for both the environment (via general assessments) and the goal to be accomplished (via specific assessments). This phase answers the question “Where am I?”

- **General Assessments**

Give an overview of both the management and the work processes. The objective of a general assessment is to evaluate the environment into which change will be made. Experience has shown that if the environment is not conducive to change, it will be difficult to introduce change into it. For example, if IT professionals do not believe their end users know what their requirements should be, it is difficult to build approaches that are driven by customer satisfaction surveys. Skill Category 2 contains a discussion on assessing the organizational climate.

- **Specific Assessments**

Relate to the activity or process for which change is directed. The two assessments below are recommended:

- Use a control chart(s) to determine the variability of the process in which change will be introduced.
- Use a dashboard of key indicators to evaluate the process subject to change. These same key indicators will be used for goal setting. Establishing the key indicators is a consensus process performed by the owners of the process.

Skill Category 3 also includes a discussion of assessment models and baselining.

Deployment Phase 2: Strategic

The deployment strategy establishes the framework in which deployment will occur. Without strategic planning, deployment rarely works. This phase results in a goal, which is normally a step towards accomplishing the vision, and a definition of the process or approach to accomplish that goal. The questions “Where do I want to be (goal or vision)?” and “How am I going to get there (process)?” are answered in this phase.

The recommended deployment strategy for quality initiatives is a four-step process:

1. Set the Goal

Establishing a vision or a goal starts the strategic phase of deployment. If there is no desire to improve, deployment tactics will not work. The gap between the current level of performance (from the assessment phase), and the goal to be achieved defines the deployment challenge. Both must be expressed quantitatively.

Goal setting is a management responsibility, but it may also be done by self-managed teams or empowered individuals. Goals are the results desired from the process change. Failure to meet a goal should not be viewed negatively since the overall objective should be continuous improvement. A goal may not be met because it was too optimistic or the process to meet it was inadequate.

Some guidelines to follow when setting goals are:

- Use measurements that were established to identify the baseline for the goal.
- The goal must be realistic (this does not preclude setting stretch goals if there is a reasonable expectation they can be achieved).
- Those responsible for achieving the goal must agree to it.
- Those responsible for meeting the goal should follow the process, and not circumvent the process to meet the goal.
- Use tools that can help with goal setting, such as consensus or benchmarking.

2. Identify Possible Solutions

Since a problem can have hundreds of solutions, it is important to identify the better ones. Good identification practice states to first identify all viable options, and then select the ones that appear to be most effective. Searching for, and identifying, effective solutions varies based on the goal. Avoid having a solution looking for a problem, or the search leading to “analysis paralysis”. Determine whether spending excessive time seeking out the optimal solution is a better choice than quickly implementing a good, effective solution that also works.

Developing the strategic plan for how to reach a goal involves building the roadmap to get from the current baseline to the desired point. For example, the strategic plan might establish moving the structure from a waterfall development process to an IT engineering environment.

Some guidelines for identifying solutions are:

- Assure that all involved parties have an opportunity to offer solutions.
- Determine how other organizations have solved the same or similar problems.
- Realize that the goal must be the driver, and there must be reasonable agreement that the solution will move the organization toward achieving it.
- Assure that it is possible to implement the solution within the organization's level of competency.
- Assure that it is possible to implement the solution within the time frame associated with the goal.
- Consider different methods of identifying possible solutions, such as brainstorming, benchmarking, affinity diagrams, a quality improvement process, request for proposals, or engaging consultants.

3. Select and Sequence or Prioritize Change Approaches

Simple goals can have simple, easy to implement solutions. If the solution for reducing job control language (JCL) errors is to acquire a software tool to edit the JCL statements, the sequencing and implementing of the solution is readily apparent.

The implementation of complex goals and solutions can have complex sequencing. For example, a goal to improve customer satisfaction may involve a series of approaches that can be implemented in a variety of sequences. Assume the approach selected was to improve the score from the MBNQA criteria. This is realistic because the goal of the Baldrige model is to improve customer satisfaction. The dilemma is that the Baldrige model has approximately 100 items to address in implementation. The question becomes: "Which of the 100 should be emphasized first, second, third, and so forth?" The objective of this step is to answer that question.

From the possible solutions in Step 2, select the most appropriate ones. Then prioritize and sequence the changes for implementation. For example, decide that IT standards should be developed before implementing the engineering tools. Identify any prerequisites and constraints associated with the needed approach.

Some guidelines for selecting and sequencing change approaches are:

- Select approaches that are most acceptable to the culture (poor approaches that are acceptable may be more effective than optimal approaches that are resisted).
- Select approaches that have the highest probability of success, first.
- Select approaches that will provide both short-term and long-term results.

- Select an implementation strategy that will use the best people and the best projects, as opposed to trying to improve the poorest projects with the least effective staff.
- Implement the prerequisites to the approach before implementing the approach.
- Use tools such as risk ranking, analysis, or return on investment.

4. Develop, Acquire and Customize the Approach

After selecting the approach, it must be implemented. If conducting a survey was the approach selected, the survey would be developed in this step. The options for implementing the approach are:

- Develop the approach; for example, select someone to write a customer survey.
- Acquire the approach; for example, purchase a survey, or engage a market research firm to conduct the survey.
- Customize the approach, starting with a generic approach and then customizing it for the organization; for example, purchase a customer survey, and then modify, delete, or add to the questions to meet the specific needs of the organization.

Guidelines to assuring that an effective approach is developed are:

- Ensure that the owners/users of the approach help develop the approach.
- Select simplicity of execution over complexity.
- Develop the quality control tools in conjunction with the approach tools (build the do and check procedures at the same time.)
- Ensure that objectives for the approach are well defined and understood (The goal should be expressed quantitatively and might be a policy for the approach.)
- Ensure that the approach is consistent with, and fits with, other interrelated activities.
- Use tools for developing, acquiring, and customizing, such as a system development methodology, a process management process, or a quality improvement process.

Deployment Phase 3: Tactical

As previously stated, effectively performing the strategic deployment activities helps ensure the success of the deployment tactics. It takes three to ten times more resources to deploy an approach than to develop it. If the deployment resources are inadequate, there is a high probability that the approach will fail, or that its use will be minimal. For example, if a capture/playback testing tool is purchased for \$25,000, installation costs should be between \$50,000 and \$250,000. If those funds are not expended, that tool will normally fall into disuse.

The tactical phase answers three questions:

When the process is initially implemented, compliance is attempted (see Step 5 below), answering the implementation question “How do I get people to follow the process?”

Measurement is performed (see Step 6 below) to answer the question “Does the process work (is it under control) and is the organization moving in the right direction to support the vision or goal?”

Based on the measurement, the question “Does the process need improvement, and if so, how?” is answered in Step 7 below.

There are seven recommended steps in the tactical phase of deployment. Each of these is listed individually; however, during execution steps may be combined into a single deployment task. For example, the hearing step may require some training in the approach (i.e., the learning step) to provide the user with enough information to fully understand its objectives and benefits. The hearing step may also incorporate some of the “selling” that is needed to get a potential user of the approach to accept it as the preferred way to perform work.

1. Hear the Approach

Those responsible for deployment must assure that those who need to deploy “hear” what they are responsible for complying with. This step assures the doers have enough information to know what they are supposed to do – not how to do the approach. Hearing does not imply that they accept the approach as doable or realistic. Tools that can help with hearing the approach are awareness training, newsletters, meeting and policy statements.

2. Accept the Approach

This buy-in step is critical in deployment because this is when people become convinced that following the approach benefits them. Users of the approach personally decide that they will use the approach, which, in turn, requires a behavioral change on their part. The behavioral change formula stated that change occurs when either the individual or the individual’s environment makes a change. Thus, acceptance should address both the individual’s belief system (e.g., the “What’s in it for them” concept) and the environment in which the individual works.

The challenge in this step is the time span required for user buy-in. The champion, the developers, and the deployers have bought-in and accepted the approach over the previous weeks, months, or even years. During the development effort, the leaders have become excited about what it can do for the organization. Unfortunately, those that have to use the approach in their day-to-day work may not yet share that excitement. The challenge is to gain their acceptance in a short time span.

Activities that help encourage acceptance include in-depth discussions of why leaders believe the approach is good, testimonials from others who have used the approach, presentations of evidential matter substantiating that the approach works, one-on-one selling, and management accepting the consequences if the approach fails. Tools that help acceptance of the process include “What’s in It For Me” (WIIFM rewards), peer pressure, leadership, commitment analysis, and a champion.

3. Learn the Approach

Once people hear and accept the approach, they are ready to learn it through education and training. Education (covered in Steps 1 and 2) teaches what the approach will do. This step covers training, that is, how to use the approach.

Some training guidelines that result in effective learning are:

- The military approach – tell them what you are going to tell them; tell them; then tell them what you told them.
- Multi-media – use as many of the learner’s senses as possible to produce effective learning.
- Practice is an important part of learning.
- Learning through teams.
- Peers helping and reinforcing each other in learning.
- Regularly repeat learning – do not assume that once someone has learned something that they will have learned the approach forever.
- Reward learning – at least, compliment the learner on a job well done
- Learning tools – such as the classroom, self-study, videos, and user manuals.

4. Remember the Approach

Hearing, accepting, and learning are not enough. If the time comes to use the approach, the individual must remember at that time to use it. As an approach becomes lengthy and complex, it is harder to remember all of it.

Some guidelines that help individuals remember when to use an approach are:

- Make it hard not to follow the approach. For example, if a step is forgotten and an activity can be stopped until that step is performed, individuals will be forced to remember to perform that step.
- Make remembering, a positive reinforcement. Negative responses discourage acceptance. For example, putting a note in someone’s performance appraisal file that he or she failed to comply with a specific standard does not motivate individuals to want to follow standards.
- Make remembering tools highly visible. For example, a notation on a screen regarding the task to follow next.
- Use tools, such as help screens, hot lines, edits, playscripts, and forms, or screens to help remember. The tools will be as varied as the approaches.

5. Do the Approach

Determine the series of steps needed to achieve compliance to the process. Useful tools that help accomplish this Do step are default options, forms or screens, QC tools, teams and peer checking, procedures, and expert systems.

This deployment process is proposed in order to get people to follow the process; however, a command or edict to follow a process doesn't work. People rarely make a conscious decision to ignore management's desires. Lack of compliance is usually attributable to conflicting messages from management; for example, follow the process and meet the schedule. Which does management rate as more important? People also fail to comply with processes because they are unaware of them or do not know how to comply with them.

6. Check the Approach

Determine that the process is accomplishing the vision or goal. This requires two methods for measuring results: statistical process control to determine that the process is being followed in a consistent manner, and management dashboards (see Skill Category 8) to determine that the process is moving the organization toward the desired vision or goal.

The probability of an individual executing the approach is high if it has been heard, accepted, learned, and remembered. The question here is, "will it be done correctly?"

To check an approach, consider the following guidelines for developing effective quality controls:

- Develop quality control to verify literal compliance (doing those things that are required to be done) and to verify intent compliance (doing those things in a manner that is consistent with the goal or objective to be accomplished).
- Place the control as close to the Do step as possible.
- Execute the Check step in a manner different from the way the Do step was performed, for example, verify division with multiplication; do not repeat the division.
- Rather than use independent checkers, let the check tool be an aid to the doer, making the doer accountable and responsible for performing the Do activity correctly, and, at the same time, providing a mechanism to assure the doer that it has been performed correctly.

7. Act to Improve the Approach

It is unrealistic to expect that an approach will be deployed effectively the first time. Either the approach is not fully effective, or the individuals using it lack the necessary skills and motivation to use it effectively. Adjust the process to bring it under control and to assure it is accomplishing the vision or goal. This is the traditional quality improvement process, and the last component of the tactical deployment process.

The improvement process is the means of assuring effective change. It recognizes and accepts deficiencies in the tactical deployment and approach, and makes the necessary adjustments and calibrations to ensure that the approach does accomplish its goal. This success formula for effective change is consistent with the PDCA cycle.

Critical Success Factors for Deployment

Deployment is much harder than defining an approach. Approach is an intellectual exercise; deployment is a people-intensive process. There are five intangible attributes called critical success factors that help make deployment work.

These five critical success factors for an effective deployment process are:

- Deployment is a series of integrated tasks, which together enable approaches to be effectively implemented.

These integrated tasks are a deployment process that should be customized as needed for each organization and each approach.

- Deployment champion(s) is in place.

Someone must take the lead for making the identified approach happen. While the champion can be a highly respected staff member, it is always advantageous for the champion to be a senior manager.

- Deployment is a team effort.

A single individual can develop an effective approach, but can rarely deploy that approach. A team of people including instructors, technicians, colleagues and management must implement the deployment process.

- There is buy-in by the affected parties.

Tasks that transfer ownership of the approach to the users of the approach involve a buy-in. In this activity an individual accepts the approach as the way business will be done. The individual does not have to like the approach, but does have to wholeheartedly support its use in performing the effective work tasks.

- Deployment responsibilities are effectively passed between individuals and between teams.

Deployment is a continuous process that begins prior to developing the approach, and goes on until the approach is discontinued. During that time, the level of enthusiasm for the approach will vary. People involved in ensuring that the approach is followed (i.e., deployed) likely will change over time. It is essential that new people involved in the work tasks have the same enthusiasm and desire that existed in the initial deployment effort.

Internal Auditing and Quality Assurance

Both internal auditing and QA are professions. It is generally recognized that a profession has the following criteria:

- Code of ethics
- Common body of knowledge
- Statement of responsibilities
- Certification program (including continuing education)

The differences between the auditing and QA professions are in the common body of knowledge and the statement of responsibilities.

Types of Internal Audits

Internal auditing is a management control directed at measuring and evaluating an activity to determine if it is performed in accordance with the policies and procedures of an organization (i.e., meets the intent of management). It is an independent appraisal activity. The specific types of auditing are:

- **Financial Auditing**
Financial auditing is performed in accordance with generally accepted accounting procedures and other applicable laws and regulations to determine that the accounting records are reasonable.
- **Operational Auditing**
Operational auditing is performed to determine that operations are performed in an efficient, effective and economical manner.
- **Program Auditing**
Program auditing is performed to determine that the objectives of specific business activities are being properly fulfilled.

There are three important characteristics in the performance of an internal audit:

1. The work of the internal auditor needs to be detached from the regular day-to-day operations of the company. A good practical test is that if the internal auditing activity were temporarily discontinued, the regular company operations would go on in a normal manner for the time being.
2. Internal auditing cannot get involved in developing procedures, standards, or usurp the roles and responsibilities of other employees.
3. The internal auditor is to evaluate the interaction of all company groups with regards to meeting objectives.

Differences in Responsibilities

The main role of auditing is to identify and report problems, while the role of QA is to find and implement solutions for those problems. QA should be a leadership position, emphasizing the strong interpersonal activities involved in making improvement occur. While QA performs many appraisals, it strives to be independent of the activities being appraised. Auditing, by nature, has a negative role; QA, by practice, should have a positive role. Confusion between the two roles frequently leads to a negative image of QA.

Some of the skills and activities that an internal auditor has are not applicable to QA analysts.

- Internal auditors must be knowledgeable of the Standards for the Professional Practice of Internal Auditing and are required to comply with those standards in the performance of their work.
- Internal auditors review the means of safeguarding assets and verify the existence of assets.
- Internal auditors verify compliance to corporate policies, plans, procedures, and applicable laws and regulations.
- Internal auditors normally coordinate their activities and work in conjunction with the organization's firm of external auditors.
- Internal auditors have direct lines of communication to senior corporate officers and frequently to the organization's board of directors.

Some key activities performed by QA analysts that are not normally performed by internal auditors are:

- Developing policies, procedures, and standards
- Acquiring and implementing tools and methodologies
- Marketing or creating awareness of quality programs and concepts
- Measuring quality
- Defining, recording, summarizing, and presenting analyses
- Performing process analysis (i.e., statistical process control)

See Skill Category 3 for a discussion of quality audits.

Quality Planning

Executive management establishes the vision and strategic goals. Planning is the process that describes how those strategic goals will be accomplished. Quality planning should be integrated into the IT plan so that they become a single plan. In simplistic terms, the IT plan represents the producer, and the quality plan represents the customer.

<i>Planning Concepts</i>	227
<i>Integrating Business and Quality Planning</i>	231
<i>Prerequisites to Quality Planning</i>	233
<i>The Planning Process</i>	234
<i>Planning to Mature IT Work Processes</i>	241

Planning Concepts

Planning is the totality of activities that determine, for an individual or organization, what will be done and how it will be done. *Quality planning* is a component of overall business planning. Quality planning focuses on the policies, processes and procedures which assure that the defined requirements are implemented, and the implemented requirements meet the customer's needs. The following two concepts epitomize the importance of planning.

- If you do not know where you are going, all roads lead there. This means that without a plan, any action is acceptable.
- If you fail to plan – plan to fail. This means that without a good plan which defines the expectations of work, activities may be performed which provide no benefit and lead to customer dissatisfaction.

Two important components of quality planning are the management cycle and the planning cycle. The management cycle, frequently referred to as the Plan-Do-Check-Act Cycle, is repeated here to emphasize the importance of planning as a management activity.

The Management Cycle

Note that the material in this section is from Skill Category 1. It is repeated here as an introduction to planning.

The management cycle comprises the four steps as illustrated in Figure 33. The four-step procedure shown is often referred to as PDCA. Repeatedly going through this management cycle is known as “going around the PDCA circle.”

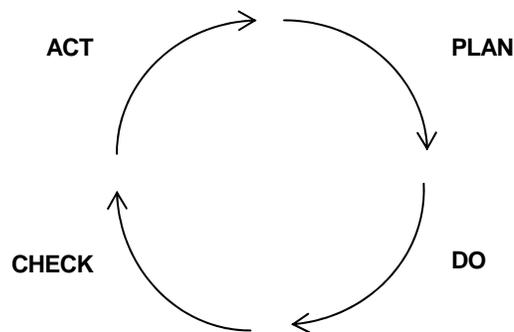


Figure 33. The PDCA Cycle

- Plan (P): devise a plan
Define your objective and determine the conditions and methods required to achieve your objective. Describe clearly the goals and policies needed to achieve the objective at this stage. Express a specific objective numerically. Determine the procedures and conditions for the means and methods you will use to achieve the objective.
- Do (D): execute the plan
Create the conditions and perform the necessary teaching and training to execute the plan. Make sure everyone thoroughly understands the objectives and the plan. Teach workers the procedures and skills they need to fulfill the plan and thoroughly understand the job. Then perform the work according to these procedures.
- Check (C): check the results
Check to determine whether work is progressing according to the plan and whether the expected results are obtained. Check for performance of the set procedures, changes in conditions, or abnormalities that may appear. As often as possible, compare the results of the work with the objectives.

- Action (A): take the necessary action

If your checkup reveals that the work is not being performed according to plan or that results are not what were anticipated, devise measures for appropriate action.

If a check detects an abnormality – that is, if the actual value differs from the target value – then search for the cause of the abnormality to prevent its recurrence. Sometimes you may need to retrain workers and revise procedures. Make sure these changes are reflected and more fully developed in the next plan.

These procedures not only ensure that the quality of the manufactured goods meets expectations, but they also ensure that the anticipated price and delivery date are fulfilled. Sometimes our preoccupation with current concerns makes us unable to achieve optimal results. By going around the PDCA circle, we can improve your working methods and obtain the desired results. Repeated use of PDCA makes it possible to improve the quality of the work, the work methods, and the results. This concept can be seen in the ascending spiral of Figure 34.

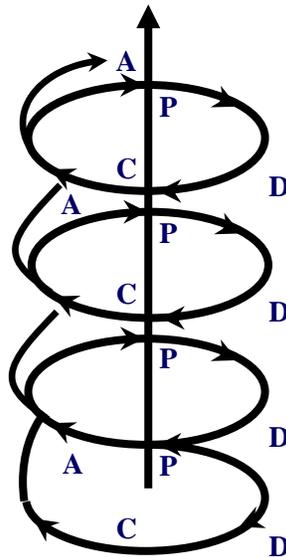


Figure 34. Ascending Spiral

The Planning Cycle

Planning is a management responsibility. The responsibility commences when management establishes a vision for the IT organization, and works through the development of a tactical plan which defines the detailed work activities to be performed.

The planning cycle is a decomposition of the IT vision into work activities which will help accomplish that vision. Table 9 shows that decomposition. It also shows the do, check, and act activities that follow when planning is completed.

The planning cycle must be integrated with the do, check, and act activities because planning is a continuous activity. While the PDCA cycle implies that you plan, then do, then check, and then act, that concept is misleading. While the plan should be complete before work activities commence, business requirements may change, and problems or opportunities may be encountered. These events which affect work activities should be incorporated into a new version of the plan.

These changes to the work activities can have any or all of the following impacts on the plan:

- Change the schedule
- Change the budget
- Change the number of resources allocated
- Change how one implemented component of software will affect other components of the software
- Change in work priorities
- Addition or deletion of work activities to accommodate the needed changed work activities

Table 9. Planning Cycle Example to Show Decomposition from Vision to Rework

Planning Activity	PDCA Phase	Example of Planning Activity
Establish IT Vision	P	IT deliverables and service exceed customer satisfaction.
Define Mission	P	We will work with our customer to assure satisfaction.
Set Goals	P	On a scale of five to one -- from very satisfied, satisfied, neither satisfied nor unsatisfied, dissatisfied, very dissatisfied -- our goal is 90% of our customers very satisfied or satisfied.
Strategic Planning	P	Involve users in the software development process.
Tactical Planning	P	Conduct reviews at the end of each development phase with users as part of the review team.
Execution	D	For project "X" conduct a requirements phase review on November 18, 20xx.
Monitoring	C	Did the requirements phase produce testable requirements?
Rework	A	Make non-testable requirements testable.

The planning cycle is illustrated with a customer satisfaction example showing the decomposition from the first listed planning activity to the last planning activity.

- Establish IT vision
 - A vision is broad in nature, probably unachievable but it is the ultimate goal.

- **Define Mission**
The responsibility of an organization unit related to achieving the vision.
- **Set Goals**
A target established by management to be achieved by the staff of the implementing organization.
- **Strategic Planning**
A description of what must be done in order to meet the goals set by management.
- **Tactical Planning**
The detailed “how-to” work activities that need to be undertaken to accomplish the strategic planning objectives.
- **Execution**
The execution of the tactical plan as written.
- **Monitoring**
An ongoing assessment to assure that the plan is followed, and problems encountered in following the plan, are appropriately addressed.
- **Rework**
Actions approved by management based on problems uncovered through the monitoring process.

Integrating Business and Quality Planning

Quality planning should focus on two major activities: process management and quality control. Process management is discussed in Skill Category 6 and quality control is described in Skill Category 7. The quality professional will do other activities, most quality activities that require planning are related to these two quality activities. Business planning should focus on accomplishing business objectives.

Let’s look at these two planning processes. The IT organization develops a business plan. The purpose of the business plan is to define the work and work activities that will be conducted during the planning period. These work activities are designed to accomplish the business objectives. The quality professionals will develop a quality plan focusing on quality activities that will help assure the outputs from the business plan meet the defined output specifications and meet the needs of the users of those deliverables.

The Fallacy of Having Two Separate Planning Processes

Many IT organizations develop both a business plan and a quality plan. However, they do not integrate these plans. Results can be expected, but are not desirable. For example, the quality plan

may call for system development reviews to occur prior to the end of a software development phase. However, if the business plan does not allot time and resources to participate in that development review, the review may never occur. In many organizations, the quality professionals who organize the reviews are not informed when a software development phase is concluding.

Planning Should be a Single IT Activity

Both the business staff and the quality staff should be involved in IT planning. Involvement is in both strategic and tactical planning.

The objective of this single planning cycle is to ensure that adequate resources and time are available to perform the quality activities. The net result is that the individuals executing the business plan cannot differentiate the quality planning from the business planning. For example, if business planning calls for a quality review prior to the end of each phase of software development, the business staff will assume, that is a logical part of the software development process. If it is not integrated, the review is owned by the quality professional and may not be adequately supported by the IT business staff, such as, system designers and programmers.

Figure 35 illustrates the integration of quality planning into IT business planning.

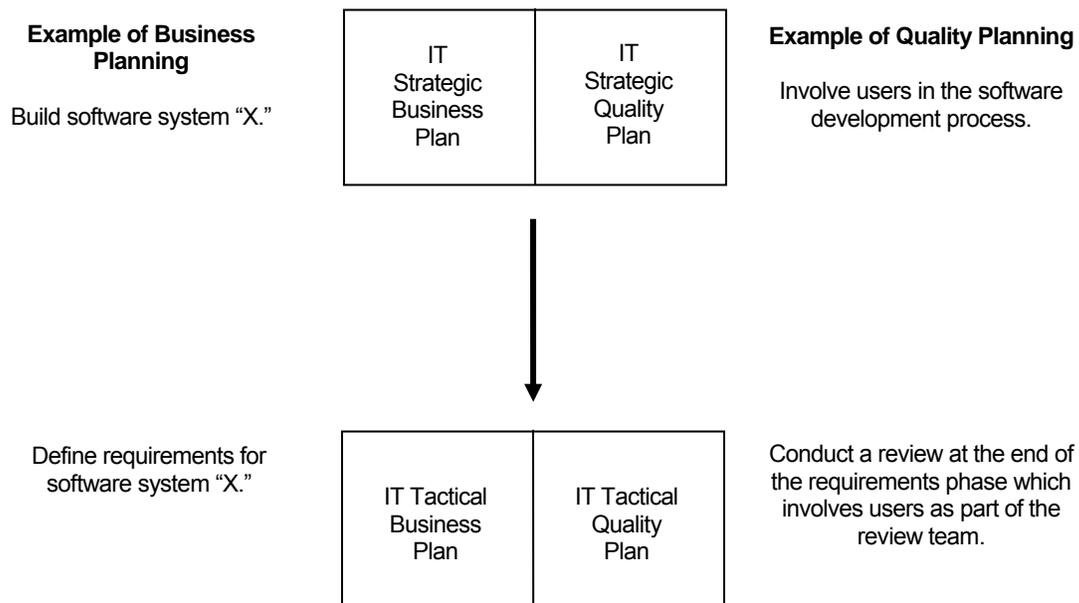


Figure 35. Integrating Quality Planning into IT Business Planning

The figure gives an example of business planning and an example of quality planning. While the blocks show that strategic and tactical business planning and strategic and tactical quality planning occur, the end result is a business plan that incorporates quality activities.

Prerequisites to Quality Planning

Quality planning is a process. Quality planning should be a defined process indicating who is involved in planning and the specific work procedures and deliverables included within the planning process. Individual IT staff members should not create their own planning process.

Before effective quality planning can occur these prerequisites should be met:

- IT vision, mission and goals documented

Those planning need to know specifically what the plan is to accomplish. The planning process begins by knowing the desired output from the plan. If those performing planning understand the IT vision, the IT mission and the specific goals related to that vision and mission, they can develop a plan that hopefully will meet those goals.

- Defined planning process

The IT organization needs a planning policy, planning standards, and planning procedures. To better understand the attributes of an effective process, refer to Skill Category 6.

- Management support for planning

Effective planning will only occur when management requires the plans be developed using the IT organization's planning process. Management support means that the plan must be completed and approved before resources will be allocated to accomplish the work defined by the plan.

- Planners competent in the planning process

Those who will use the planning process to plan need to be competent in using the planning process. The competency can be achieved by training, or working under a mentor to teach effective planning. Normally both should occur.

- Compliance to the plan

If a planning process exists, management should require compliance to that process.

- Maintenance of the planning process

Planning processes should continually be improved. If the process is not working or not effective the process should be changed.

- Reliable information required

The plan will be no better than the information used to create the plan. If those performing quality planning cannot rely on the information provided them, they should take whatever steps necessary to assure them that they're working with valid and reliable information.

The Planning Process

The planning process is the same for business planning and quality planning. There are literally hundreds of different books on planning. Discussed below are the planning activities that are most frequently identified as important to effective planning, as well as, these three areas:

- Planning process overview
- Basic planning questions
- Common planning activities

Planning Process Overview

While there is no standard off-the-shelf plan for planning that can be universally applied to every situation, the systematic planning process described in this section provides a great wealth of experience, concepts and materials. This process can be adapted to most organizations and thus avoid the necessity to “reinvent the wheel.”

The quality of planning and decision-making cannot consistently rise above the quality of the information on which it is based. But planning will still be poor in spite of good information unless each manager is able to coordinate adequately with his/her associates and consolidate his/her planning to support the IT objectives. This planning process as illustrated in Figure 36 provides an easy, economical way to collect the process data, retain it, retrieve it and distribute it on a controlled basis for decision-making.

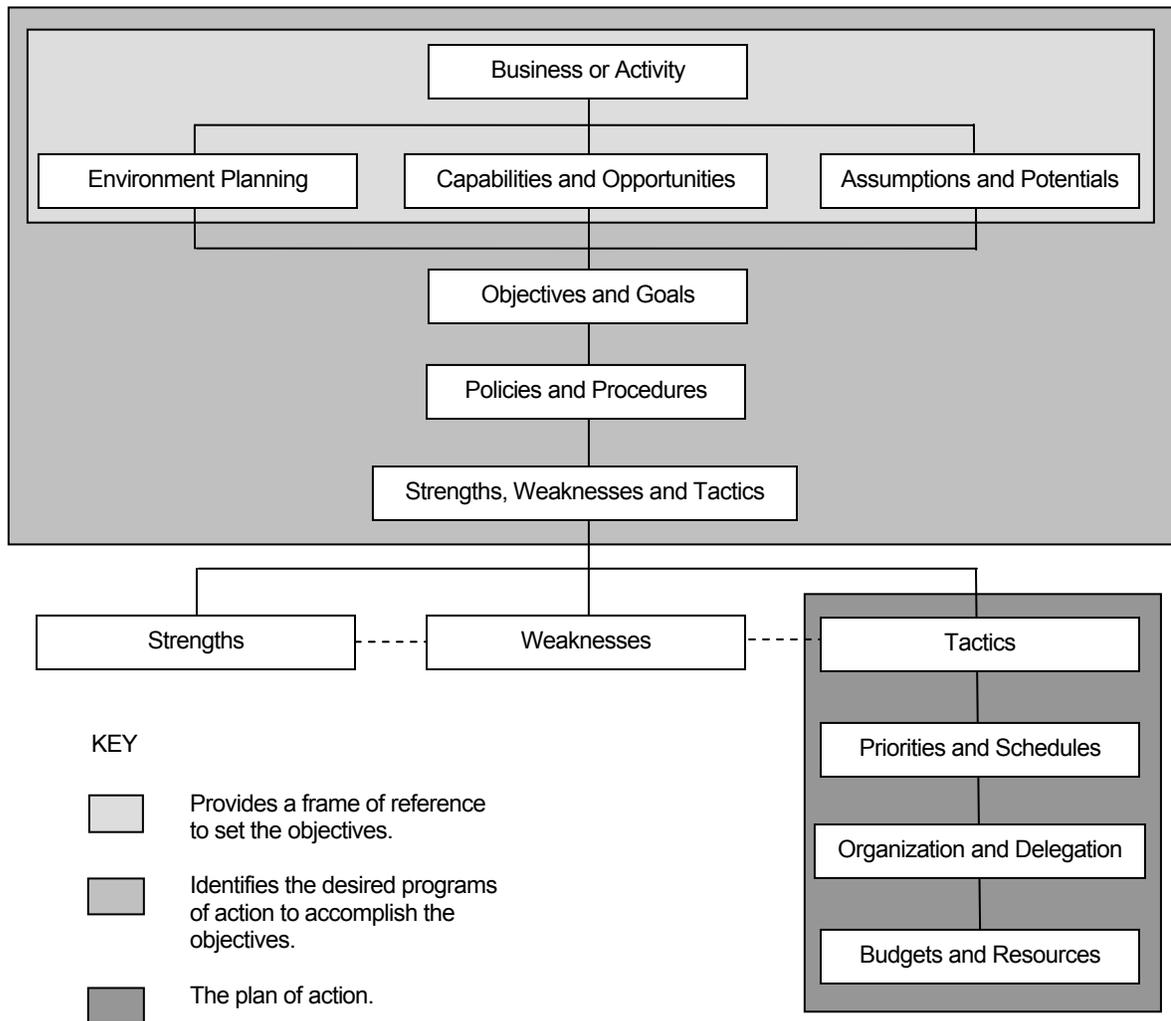


Figure 36. The Planning Process

This planning process is divided into the following ten planning activities:

- Business or Activity Planning
- Environment Planning
- Capabilities and Opportunities Planning
- Assumptions and Potentials Planning
- Objectives and Goals Planning
- Policies and Procedures Planning

- Strengths, Weaknesses and Tactics Planning
- Priorities and Schedules Planning
- Organization and Delegation Planning
- Budgets and Resources Planning

Like most important things in life, it is impossible to describe the scope and purpose of this approach to planning by enumerating the bits and pieces. The primary purpose of planning is not to produce a rigid plan, but to facilitate intelligent delegation with responsible participation by providing a better method of reaching and revising agreements. Most specifically, it minimizes surprise and optimizes performance in a changing environment. The whole must be greater than the sum of its parts. Yet, many managers abhor planning. The common reason heard most frequently is that planning systems cannot be installed because “our business is a dynamic one which is changing daily.” Obviously, there is probably not a single, viable business today which is not changing. If one does seemingly fall into such a category, it is stagnating. A stagnating one soon dies.

The practical, systematic approach to planning was best described by Peter Drucker in “The Practice of Management:”

“There is only one answer: the tasks must be simplified and there is only one tool for this job: to convert into system and method what has been done before by hunch or intuition, to reduce to principles and concepts what has been left to experience and ‘rule of thumb,’ to substitute a logical and cohesive pattern for the chance recognition of elements. Whatever progress the human race has made, whatever ability it has gained to tackle new tasks has been achieved by making things simple through system.”

The following material in this section identifies the contents of the planning activities illustrated in Figure 36 above.

The Six Basic Planning Questions

The ten planning activities described in Figure 36 were designed to answer six basic planning questions as listed below. The planning process then documents the answers to these six questions:

- Where are we?
- Where do we want to go?
- How are we going to get there?
- When will it be done?
- Who is responsible for what?
- How much will it cost?

Table 10 shows how each question links to one or more of the ten planning activities and the information that is needed to answer the question and perform the activities.

Table 10. The Six Basic Quality Planning Questions

Six Basic Questions	Planning Activities	Planning Information Needed
1. Where are we? (Historic and current information, present time, and facts)	Business or Activity Planning	<ul style="list-style-type: none"> ✓ Nature of Business-Purpose, Scope, History ✓ Management Philosophy ✓ Profiles of Business-Revenues, Profits, Products, etc.
	Environment Planning (External to Company)	<ul style="list-style-type: none"> ✓ Organization and IT work environment ✓ Economic, Social, Political, Industry Regulations and Laws ✓ Identify and Analyze input on other organizations
	Capabilities and Opportunities Planning	<ul style="list-style-type: none"> ✓ Capabilities (strengths, weaknesses – internal/controllable) ✓ Problems (external/partially controllable) ✓ Opportunities ✓ Analysis by Key Result Areas
2. Where do we want to go? (Dealing with the future, cannot be predicted with accuracy)	Assumptions and Potentials Planning	<ul style="list-style-type: none"> ✓ Temporary future estimates of probable developments beyond our control. e.g., populations, interest rates, market potentials, government regulations and impact of competitive actions.
	Objectives and Goals Planning	<ul style="list-style-type: none"> ✓ Temporary estimates of desirable results achieved by our own efforts. Quantified measurable objectives (5-year and fiscal year month-by-month). For example, revenue, products, expenses, profits, productivity objectives.
3. How are we going to get there?	Policies and Procedures Planning	<ul style="list-style-type: none"> ✓ Current policies/procedures hindering performance ✓ Required policies/procedures to improve performance
	Strengths, Weaknesses, and Tactics Planning	<ul style="list-style-type: none"> ✓ Strategy is a course of action selected from among alternatives as the optimum way to obtain major objectives. ✓ Select tactics that maximize strengths and minimize weakness. ✓ Define tactics.
4. When will it be done?	Priorities and Schedules Planning	<ul style="list-style-type: none"> ✓ Assign order of accomplishment for programs. ✓ Identify specific milestones to measure progress on a month-by-month basis.
5. Who is responsible for what?	Organization and Delegation Planning	<ul style="list-style-type: none"> ✓ Specify organizational relationships, organizational charts, and responsibility profiles. ✓ Specify who is responsible for the program of action and identify areas of decision-making and the accompanying authority required to accomplish the programs. ✓ Plan now for your organization requirement 2-3 years from now so you have the right person, at the right place, doing the right work, in the right way at the right time.
6. How much will it cost?	Budget and Resources Planning	<ul style="list-style-type: none"> ✓ The operational budget should place price tags on the tactics. Monthly operating budgets by department. ✓ Capital budgets by month and by year ✓ List of major resources—dollars, facilities, information

There are a large number of planning processes that are effective. What are important in understanding an effective planning process are the activities that must be performed and the

information needed to develop an effective plan. From a quality profession perspective, any planning process that works would be an acceptable response to any question on the CSQA exam about planning. QAI recognizes that quality professionals should follow the planning process adopted by their organization.

The Common Activities in the Planning Process

The ten planning activities listed in Figure 36 are common to most planning processes. Some planning activities have more activities, while others will combine the ten into a fewer number. The description of each of the planning activities that follows can be used for two purposes. First to understand the planning model presented in this section. Second to perform a self-assessment on your organization's planning process to identify potential improvement opportunities.

Business or Activity Planning

The purpose of this planning activity is to both define the planning objectives, and to relate the plan to other documents that involve individuals. Specifically, this planning activity should address:

- Vision, mission and goals
- Who are the customers/users
- What are the business needs of the customers/users
- Interfacing software systems
- Profile/description of customer/user activities.

Environment Planning

Planners need to know the environment in which the deliverables will be developed as well as the environment in which the deliverables will be operated. Specifically, this planning activity should address:

- The environment established by the organization and the IT function that impacts the means by which work is performed
- Laws and regulations affecting the products produced and operated
- Other organizations and systems that are interfaced or impacted by the products being developed and operated (e.g., payroll systems automatically sending tax information to governmental agencies).

Capabilities and Opportunities Planning

This planning activity needs to identify criteria that are required for the success of the project. The criteria need to be ranked so the proper emphasis can be placed on the most important criteria. The planning activity needs to identify the capabilities and competencies of the individuals who will develop the products. The business opportunities that can be achieved by the project also need to be identified. Specifically, this planning activity should address:

- Critical success factors
- Strengths and weaknesses of the assigned staff
- IT's ability to meet the project goals (e.g., turnaround time, number of clicks to get information, etc.)

Assumptions/Potential Planning

This planning activity identifies those probable developments that will affect the ability to achieve the project objectives. These developments should be as specific as possible in terms of how much and when. In this activity the planners are trying to describe what will happen in the next month or years so that implementation can seize any new opportunities that may develop. Specifically, this planning activity should address:

- Assumptions which if not correct will impact the success of the project
- Current opportunities received from implementing the project
- How future opportunities will be identified during the implementation and operation timeframe of the project.

Objectives/Goals Planning

Setting the project objectives/goals is an important component of the planning cycle. Goals and objectives should be measurable and achievable. If goals are not measurable it will be difficult to determine whether or not the project is successful. If the goals are not achievable the assigned staff are in a “no win” situation. However, some management philosophies believe in using stretch goals to push people to achieve a higher level of performance. Specifically, this activity should address:

- Project objectives and goals expressed in quantitative terms
- Any qualifications for objectives that can impact the sequence of work, or alternative strategies can be determined
- Quality and productivity goals.

Policies/Procedures Planning

This planning activity needs to identify all of the policies, procedures and practices that will impact the implementation and operation of the project. This analysis needs to determine whether that impact can be positive or negative. Specifically, this planning activity should address:

- Documenting the processes to be used in implementing and operating the project (i.e., policies, standards, procedures and practices)
- Changes needed to processes
- Existing processes or parts of processes, not applicable to this project
- Process variances needed and how those variances will be obtained.

Strategy/Tactics Planning

The strategy defines what to do, and the tactics how to do it. This planning activity is normally the most time consuming activity as planners need to explore multiple strategies and tactics. Specifically, this planning activity should address:

- Select preferred strategy among alternatives
- Select best tactics among alternatives
- Select tactics that maximize strength and minimize weakness
- Document tactics
- Get buy-in from those involved in the project.

Priorities/Schedules Planning

This activity develops precise milestones that identify steps or activities that are essential to accomplishing the project in the sequence in which they must occur. Many organizations use critical path planning or equivalent for this planning activity. If the completion date is predetermined then the milestones must consider what can be accomplished within the available time span. Specifically, this planning activity should address:

- Required and realistic completion date
- Milestones that need to be met to finish by the scheduled completion date
- Sequence in which activities must be performed to determine whether or not the scheduled date can be met.

Organization/Delegation Planning

The only way to develop people is to give them an opportunity to make decisions. Therefore, a responsibility profile should be developed by each staff member and discussed and revised with management so that understanding and agreement are accomplished. Specifically, what decisions (and limits) and related responsibilities are delegated to accomplish the task. Specifically, this planning activity should address:

- Responsibilities for each employee assigned to the project
- Responsibilities of support staff/individual
- Agreement by the individual that those responsibilities are adequate and reasonable

Budget/Resources Planning

The resources needed for projects include employees, supplies, capital items such as software and hardware, information, support staff, education and training, and other monetary needs. These resources can be incorporated into a budget. Specifically, this planning activity should address:

- Monetary resources needed
- Skills/competencies needed
- Hardware/software needed
- Support needed
- Information needed
- Training needed

Planning Activities for Outsourced Work

Most of the planning activities that occur for a project developed in-house must also be performed for outsourced projects. However, since the planners do not have direct control over a lot of the resources in the outsourced organization, many of the planning activities need to be incorporated into contract negotiations. To better understand the changes in planning when work is outsourced, refer to Skill Category 10.

Planning to Mature IT Work Processes

A major component of most quality plans will be defining, deploying, and improving IT work processes. The quality plan should identify specific processes to be defined, deployed and improved with specific objectives and goals defined for those work processes. Quality assurance should be involved in identifying needed new processes and where existing processes need to be improved.

A quality plan should include a process model whose achievement would be a quality goal. Many organizations have chosen the SEI CMMI Capability Maturity Model as that goal. Others have selected the ISO Model. For information on those models, refer to Skill Category 3.

Many industry process models are designed for software development and not an entire IT organization. Quality professionals need a model that represents the entire IT organization because it is the environmental components of the IT organization that drive quality.

Specifically, some of the components missing from many industry process maturity models are management processes, leadership processes, testing processes, training processes, motivation processes and customer/user-focused processes, such as handling customer complaints. The Malcolm Baldrige National Quality Award Model includes these, but does not focus heavily on IT processes.

Note, that no one model is appropriate for, or applicable to all IT organizations. Therefore, quality professionals need to know the logical steps that IT organizations should follow in maturing their IT work processes, assure adequate competency in this area, understand that industry or equivalent models, and how to implement that model to mature IT work processes.

QAI Model and Approach to Mature IT Work Processes

The Quality Assurance Institute (QAI) has developed an approach that defines six categories of processes that need maturing. This QAI model is included in the CSQA CBOOK because it incorporates the components of SEI Capability Maturity Model, the ISO models, and the Malcolm Baldrige National Quality Award Model.

In an organization whose focus is on managing people, processes are not managed effectively. In an engineering-oriented organization, processes are managed; these processes become the core competencies of an IT organization.

In the QAI model, the processes needed to manage an IT organization fall into six categories of process management as illustrated in Figure 37. The organization's vision, principles, assumptions, and goals will shape and customize the individual processes used within each category.

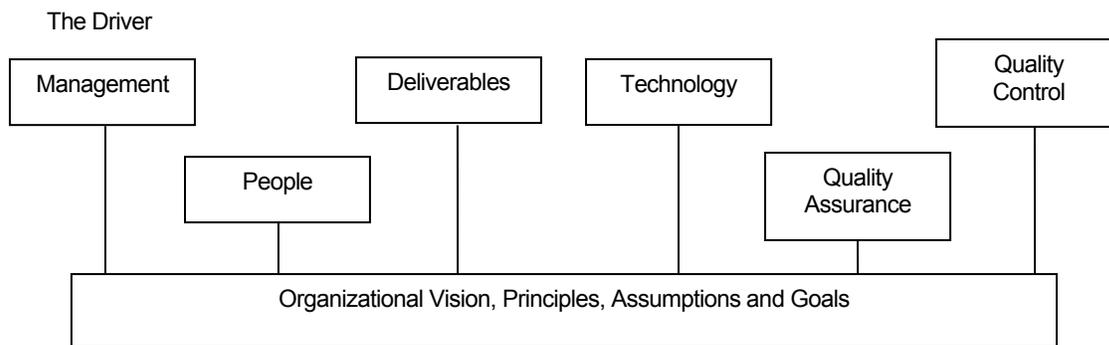


Figure 37. QAI's Tactical Approach to Quality Management Process Categories

Each category QAI has identified for the tactical approach for process management are:

- **Management Category**
The top levels of management within an IT function are responsible for planning and managing the short- and long-term activities within that function, as well as assuring they are aligned with the organization's overall vision, mission, principles, and assumptions.
- **People Category**
The people management activities are governed by human resources, salary administration, and related activities within the organization.
- **Deliverables Category**
These are the activities within the IT function held accountable for developing and maintaining software. In some organizations, this function will be performed through

contracting or purchase of software. The activity is responsible for the software operating capabilities and the data needed to support those capabilities.

- Technology Category

This primarily involves the computer operations activities, such as selection of hardware, communications, and operating software, environmental processes, security, and configuration management.

- Quality Assurance Category

The individuals, groups, and activities who are responsible for defining and improving work processes. In some organizations this responsibility falls under the standards committee and the subcommittees or work groups that develop individual standards and processes.

- Quality Control Category

These are the activities and individuals that have responsibility for the check and act components of the PDCA cycle. These activities include quality assurance, independent test, problem tracking, and other audit-type activities.

Why Six Process Categories Were Chosen

There are two basic reasons why QAI chose the above six categories:

- A splintered approach to manage by processes will not be effective.

All aspects of an information services activity must be managed by processes to gain the benefit of relying on processes. The Software Engineering Institute Capability Maturity Model focused on the software development process. Many organizations were disappointed in their inability to use that model to gain significant improvements in quality and productivity. The reason being that SEI's CMM ignored the people that use the process, the technology used by the people, as well as the technology used by the process. Other organizations that have tended to concentrate on testing as the means to improve quality and productivity have found it difficult to "test quality into software." Until the totality of the processes needed to manage information services are identified and addressed, using individual processes to enable results to be achieved has only minimal probability of success.

- The six categories encompass the management initiatives needed for IT success.

If management knows where they are going, and can identify the drivers that will enable them to achieve those results, the organization will be successful. QAI's two decades of experience with over 1,000 member organizations has identified the six drivers of information services success. These drivers have been restated as categories of processes. QAI believes that by managing these six categories of processes the probability of information services success is significantly increased.

Manage by Processes Tactical View of Process Maturity

Maturity means evolving slowly toward a desired goal. In the case of the six process categories, it means they will mature to the point they perform in an optimal manner. Many models describe this optimum process level as *world-class*, meaning as good as it can be, given today's technology.

Maturing a process is like building a building. It requires starting with a solid base and then continually adding until construction is complete. Like construction of a building, it is important in process maturity that the pieces be put in place in the proper sequence to ensure that the finished product is both efficient and effective. Management must recognize that the quickest way to optimize a category of processes is through a planned maturity process, as opposed to editing the optimum process in place without going through the preliminary steps. Just like buildings will collapse without the proper foundation, so will processes collapse and fall into disuse without the proper foundation.

Figure 38 shows the names of the five maturity levels for each process category. A more detailed explanation of each process category is individually discussed.

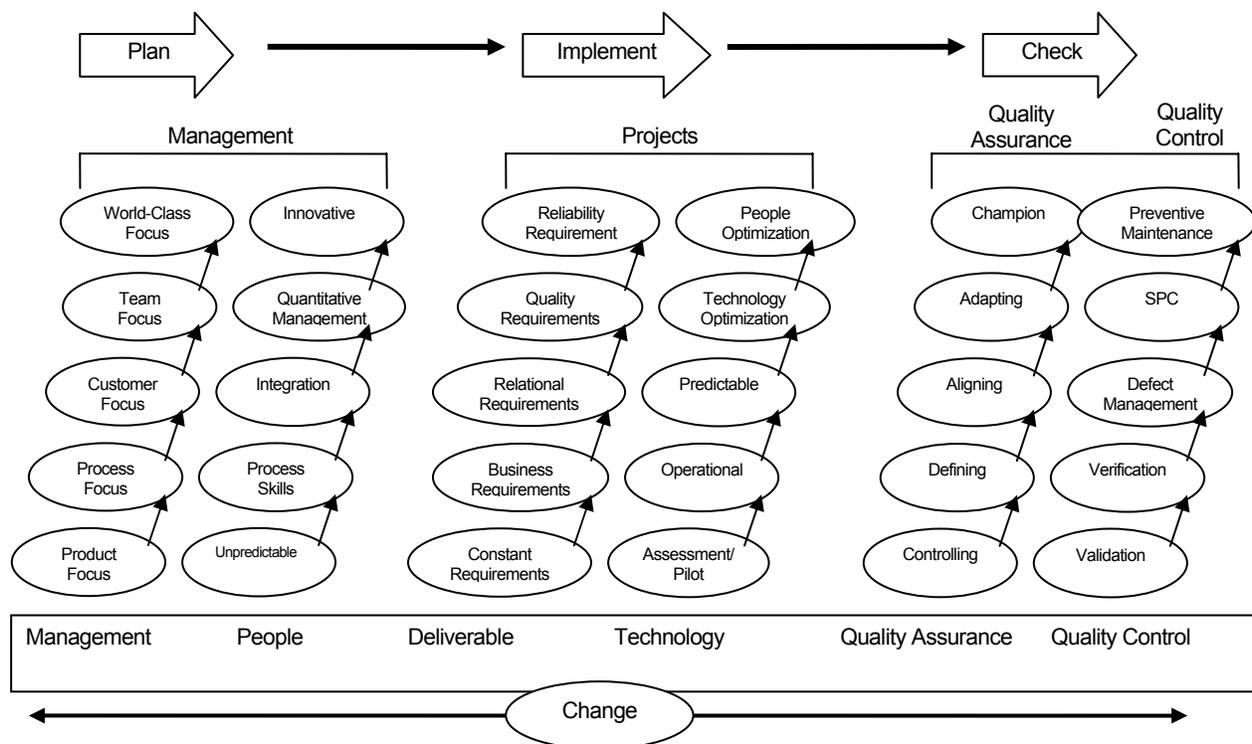


Figure 38. Maturing the Six Individual Process Categories

Tactics for Maturing the Management Processes

Accounting theory defines two categories of control. These are the management controls and the application controls. Of the six process categories, management processes are in fact what accountants refer to as management controls and the remaining five process categories are what

accountants refer to as application controls. The significant concept from accounting is that if the management controls are not in place and working, the application controls will not work. Put in the context of QAI's approach, if the management processes are not in place and working, the other process categories have little chance of working.

This concept is significant for two purposes. First, it helps define the sequence in which the categories of processes must be installed. The management process must be in place and working prior to the other process categories being developed, put in place, and working. Second, and perhaps more important, is that without the management processes in place, the movement to higher levels will not occur. For example, if the management processes which emphasize management by processes are not in place and working, it will be very difficult to get the staff to design and follow work processes such as system development, because they know their management will not evaluate them on following those processes.

Figure 39 and Figure 40 illustrate the five levels of maturing the management processes. The tactics and practices commonly used at each of the five levels are shown below and are described in the following paragraphs.

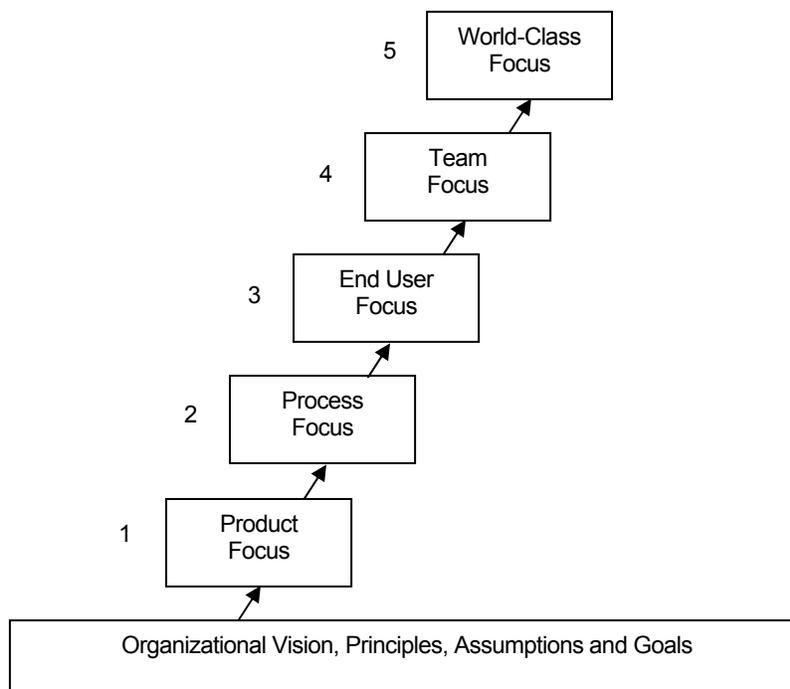


Figure 39. Tactics for Maturing the Management Processes

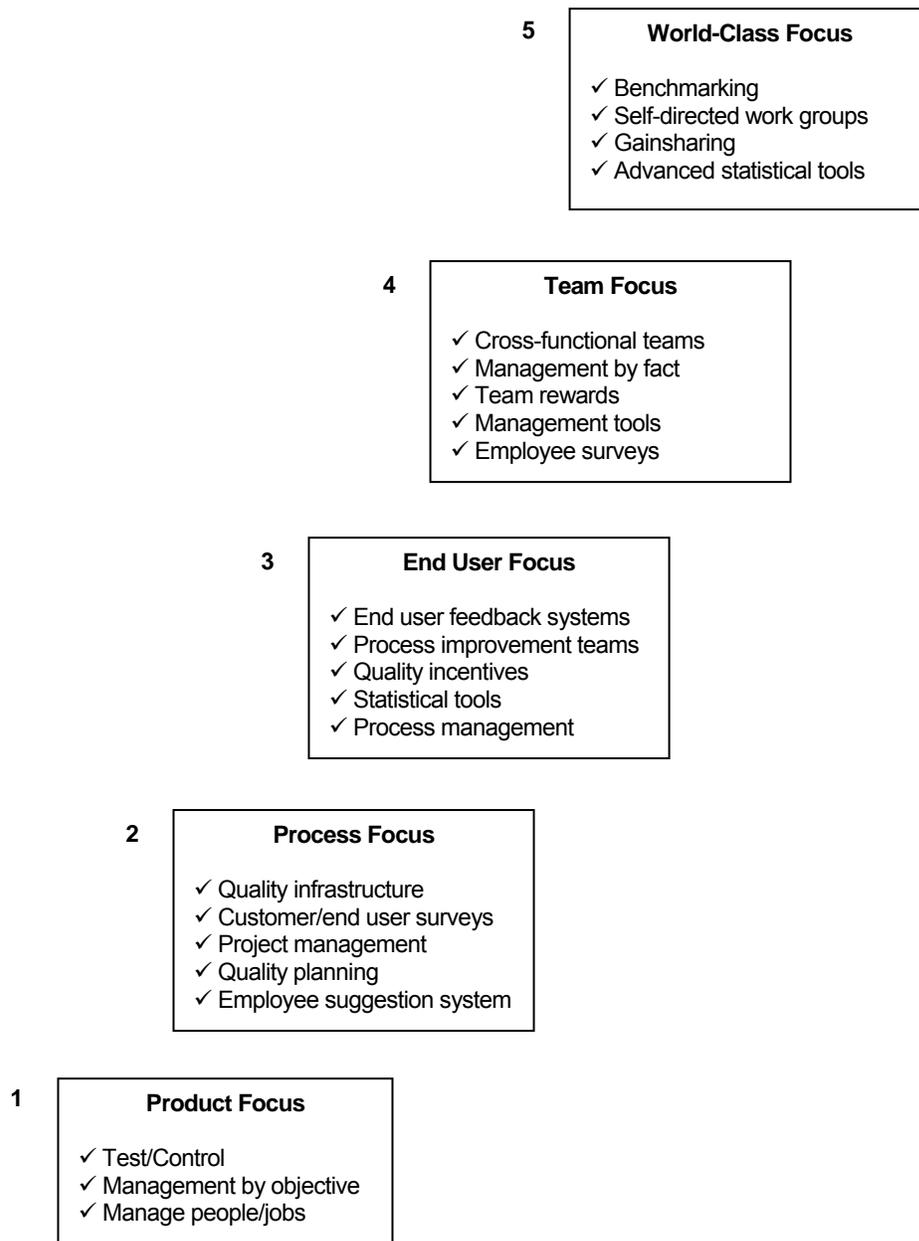


Figure 40. Most Common Practices for People Management Processes

Level 1 -- Product Focus

In this process category, management's focus is on products. The product focus emphasizes management by objective; specifically meeting schedules, managing people and jobs, and using test and control to determine whether those objectives have been met. At this level, status reports, budgets, checkpoints, and meetings are an important component of how management executes their management function.

Level 2 -- Process Focus

Management's emphasis is to bring discipline to their organization through definition and compliance to processes. The processes are usually an accumulation of the best practices currently in use within the information activity. Management creates an environment for discipline by establishing a quality infrastructure, usually a quality council or committee with several lower-level committees such as a standards or process management committee. At this level, management begins quality planning, conducts end user surveys, encourages employees to make suggestions and recommendations for improvement, and changes the methods by which projects are managed to a more scientific method using process management tools with emphasis on managing the process.

Level 3 -- End User Focus

At this level, management is truly able to focus on the end user. This level is achieved by building a competency-based organization. Management has focused the resources of the organization to meeting its mission/objectives. The practices common at this phase are process management practices based on competency, quality incentives based on competency, and end user feedback systems that impact the establishment of competencies. This level formalizes the use of statistical tools in evaluating processes, and formalizes staff's process improvement teams.

Level 4 -- Team Focus

The emphasis at this level is moving from committees directed and controlled by management to teams empowered to take action on their own. Turning over the day-to-day operation of the team is possible because the quantitative data initiated at Level 4 enables management to manage by fact as opposed to managing people and jobs. The management practices common at this level include team building, team rewards, cross-functional teams, employee surveys, and the widespread use of management tools by the teams.

Level 5 -- World-Class Focus

The teams become truly self-directed and motivated through sharing in the gains made by the organization. Management practices now include benchmarking and advanced statistical tools such as quality function deployment.

There are five subcategories of processes within the management processes, which are illustrated in Table 11. The subcategories define the initiatives that management must take as they move from level to level within the management process category. Table 11 shows the type of practices that would be included within each of these seven initiatives as management matures their processes from Level 1 to Level 5.

Table 11. Best Practices for IS Management (by process category)

Process Subcategory		Initiatives						
Focus	Level	Leadership and Planning	Infrastructure	End User Focus	Training	Employee Involvement	Incentives	Tools
World-Class Focus	5	Focus is on world-class benchmarking	Continuous improvement is natural	End user satisfaction, not profit, is	Statistics is a common language	People empowerment self-	Gainsharing (cross-functional)	Use of advanced tools such as

			behavior even for routine tasks	primary goal	among all employees	directing work groups	teams)	quality function deployment
Team Focus	4	Focus is on improving the system	Use of cross-functional improvement teams	End user feedback used in decision-making	Training on measurement (management by fact) by process category	Manager defines limits: asks group to decide	More team than individual incentives and rewards	Management tools used
End User Focus	3	Adequate money and time allocated to continuous improvement	Use of improvement teams	Tools used to include end user wants and needs in decision	Training on process management/improvement	Manager presents problem, gets suggestions, makes decisions	Quality-related employee selections and promotion criteria	Statistical tools used for variation reduction
Process Focus	2	Balance of long-term goals with short-term objectives	Executive steering committee, standards committee established	End user rating of company is known	Training on work processes	Manager presents ideas and invites questions, makes decisions	Effective employee suggestion program used	Budgeting, scheduling, status
Product Focus	1	Traditional approach to quality control: a) testing is the primary tool (control of defects, not prevention); b) better quality equals higher cost; c) MBO used to motivate/reward.						

Tactics for Maturing the People Management Processes

The people management process is a maturity framework that describes the key elements of managing and developing the talent of an organization. It describes an evolutionary improvement path from an ad hoc, careless approach to managing the talent, to a mature, disciplined development of the knowledge, skills, and motivation of the people that fuels enhanced business performance. The people management process helps software organizations:

- Characterize the maturity of their human resource practices,
- Set priorities for improving their level of talent,
- Integrate talent growth with process improvement, and
- Establish a culture of software engineering excellence.

Each maturity level, as illustrated in Figure 41, in the people management process, institutionalizes new capabilities in the improvement program, resulting in an overall increase in the people management capability of the organization. Each maturity level adds new power and sophistication to how talent is developed and motivated in the organization. Growth through the maturity levels creates fundamental changes in how people are managed and the culture in which they work.

Figure 41 and Figure 42 illustrate the five levels of maturing the people management processes. The tactics and practices commonly used at each of the five levels are shown below and are described in the following paragraphs.

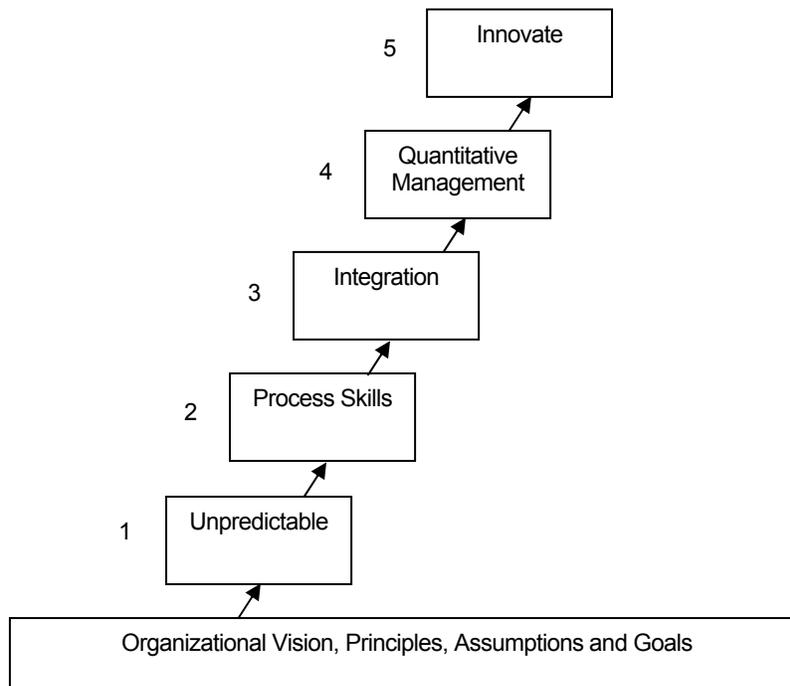


Figure 41. Tactics for Maturing the People Management Processes

Figure 42 displays the key process areas for each of the five maturity levels in the people management processes. Each key process area identifies a cluster of related activities that, when performed collectively, achieve a set of goals considered important for enhancing people management capability. Key process areas have been defined to reside at a single maturity level. Key process areas identify the issues that must be addressed to achieve a maturity level. They are building blocks that indicate the practices an organization should focus on to improve its level of talent.

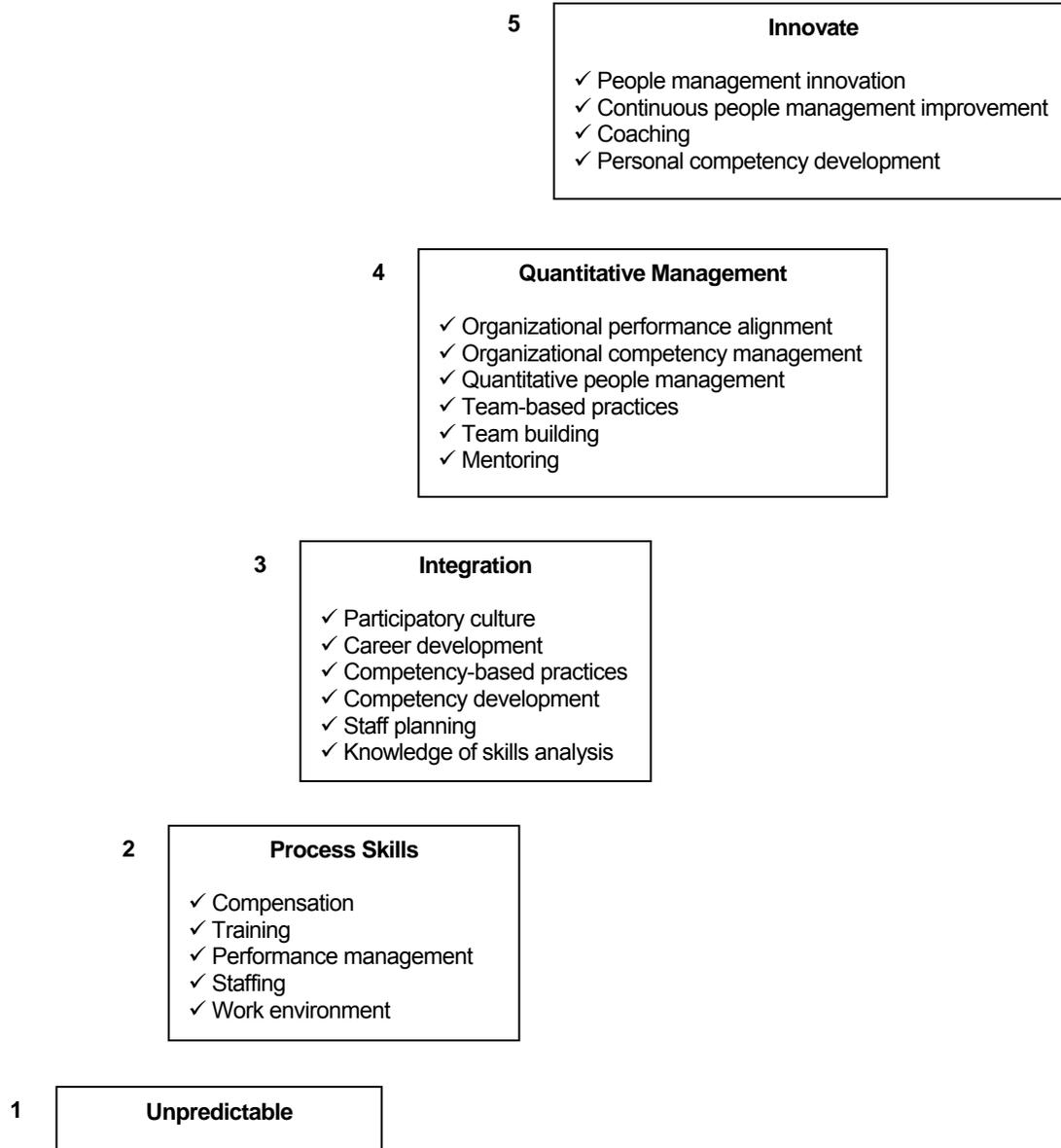


Figure 42. Best Practices for People Management Processes

Level 1 – Unpredictable

At the unpredictable level, the organization typically does not provide a consistent environment for managing its people. Human resource activities are too often treated as necessary bureaucratic overhead and are performed with little care. Typically, managers have not been trained in performing most of their people management responsibilities, so their ability to manage those who report to them is based on previous experience and their personal "people skills." The management of the organization's talent is very inconsistent, and there is no basis for improving it.

Many important people management practices, such as recruiting, are not accepted as serious responsibilities to be performed, at least in part, by those responsible for a unit's performance. Other practices such as selection, while taken seriously, are not performed in a disciplined way, often resulting in poor staffing decisions. Many managers in Level 1 organizations do not accept developing the talent of their unit as a serious personal responsibility, with the result that they rely on someone else, usually the human resources department, to perform many of these functions for them. The human resources department too often borrows practices and applies them with little analysis of the effectiveness of the resulting implementation, or may not interface effectively with the units they service. Staff members in most Level 1 organizations do not take the people management practices seriously, since they do not believe the practices have much relation to their real work and level of contribution to the organization.

The people management capability of a Level 1 organization is unpredictable because people management practices are unevenly applied. Staff members are motivated to pursue their own agendas, since there are few incentives in place to align their motivations with the business performance objectives of the organization. In the worst case, the staff's level of commitment to the organization is modest at best, resulting in frequent turnover. Consequently, the level of knowledge and skills available in the organization does not grow over time because of a variety of reasons, ranging from the constant need to replace experienced and knowledgeable staff that has terminated due to the inadequate training provided by some organizations.

Level 2 – Process Skills

At this level, there are policies, procedures, and practices that commit the organization to implementing and performing consistent, established management of its people. Those who have been assigned responsibility for performance of people management activities accept personal responsibility for ensuring that all people management practices are performed effectively. In doing so, they accept the growth and development of their staff to be a primary responsibility of their position. When these responsibilities are taken seriously, managers or empowered teams will begin to repeat methods they have found to be most successful in implementing people management practices. Members of the staff will find greater consistency in the performance of people management functions as they transfer across IS activities, although different managers or groups may have individual variations in the specific methods they use.

A primary objective in achieving process skills capability is to institutionalize the effective performance of basic people management activities. This institutionalization establishes a discipline within the organization in implemented people management activities on which improved practices can be built. In such an environment improved people management practices have a chance to yield their intended results, since there will be greater discipline in their performance across the organization. Without this discipline, it is difficult to improve the performance of the organization.

The effort to implement improved people management practices begins when the organization implements a statement of its values on managing its talent. This statement of values becomes a commitment by senior management to ensure that the organization constantly improves the knowledge, skills, motivation, and performance of its staff. Based on the organization's documented policies, units develop plans for satisfying their people management needs and responsibilities. The specific areas they take responsibility for include recruiting and selection,

performance management, training and career development, compensation and reward, the work environment, and the development of a participatory culture.

When these people management practices are institutionalized and carried out as a matter of plan and preparation, then the organization has laid a foundation to use its people management practices as a basis for systematically growing the knowledge and skills of the staff and beginning to align performance with goals at all levels of the organization. However, until these basic practices become commonplace, the organization will have difficulty adopting more sophisticated people management practices.

Level 3 – Integration

At this level, the organization begins to tailor its people management practices to the specific nature of its business. A Level 3 organization focuses its practices on developing the specific knowledge and skills that are needed to perform the organization's business activities. Through tailoring, the organization begins to standardize some of its people management practices around its unique characteristics. Further, the organization begins to identify best practices in its own people management activities or those of other organizations that it can adopt in a common framework for people management.

Since the organization has established a basic discipline for performing people management functions in each of its units at the process skills level, it can begin developing strategic and near-term plans for developing talent across the organization.

The organization begins to define its people management practices by analyzing its business processes and decomposing these processes into the tasks to be performed. These tasks are analyzed to determine the knowledge and skills required to perform them. These knowledge and skill requirements are further analyzed to determine the core competencies required by the business of the organization.

The organization bases the administration of its people management practices on developing and rewarding competence in its core competencies and on demonstrating effective application of these skills through improved performance. Common practices are defined for use throughout the organization for growing its core competencies. A program is defined for systematically developing core competencies, and career development strategies are matched to different clusters of knowledge and skills. The people management practices established at Level 2 are now tailored to develop and reward growth in the core competencies required by the business.

A common organizational culture can develop in Level 3 organizations, because the organization becomes focused on developing and rewarding a set of core competencies across the organization, along with the unique knowledge and skills required in each unit. This culture places importance on growing the organization's capability in its core competencies and the entire staff begins sharing responsibility for this growth. Such a culture is empowered when the people management practices are tailored to encourage and reward growth in the organization's core competencies.

The people management capability of Level 3 organizations is to focus on developing and rewarding the knowledge and skills that contribute directly to enhanced business performance. The

organization now has the ability to predict the performance of its different activities based on assessing the knowledge and skills it has available for application.

Level 4 – Quantitative Management

At this level, the organization sets quantitative objectives for the effectiveness of its people management practices, its growth in core competencies, and the alignment of performance across the individual, team, unit, and organizational levels. An organization-wide database is used to collect and analyze the data available from performance of different people management functions. These measures establish the quantitative foundation for evaluating trends in the organization's people management capability.

At the management level, the organization makes use of competency-based teams, as appropriate, and ensures the alignment of performance at the individual, team, unit, and organizational levels. Teams are built around complementary knowledge and skill sets, and team-building activities are employed wherever possible. The organization undertakes formal team-building to integrate the knowledge and skills required to accomplish its business functions. Mentoring uses the experience of the organization's staff to provide personal support and guidance to less experienced members of the staff.

Organizational growth in each of the organization's areas of primary competency is quantitatively managed. Standard data is defined on the performance of the organization's people management practices. The organization uses the data to analyze trends in the effectiveness of the practices and to identify unexpected results that may need corrective action. Data on the level of core competencies in the organization is analyzed to determine trends and capability. These competency trends are compared to trends in the effectiveness of people management practices. Performance data is collected and analyzed for trends in the alignment of performance at the individual, team, unit, and organizational levels. Trends in the alignment of performance are compared to trends in the effectiveness of people management practices and in the growth of the organization's capability in its core competencies. These trends are tracked against the objectives set in the strategic and near-term work force plans.

The people management capability of Level 4 organizations is predictable, because the current capability of the staff is known quantitatively. This predictability is further enhanced by knowing how effectively the performance results and trends are aligned with the appropriate goals at all levels of the organization. Future trends in staff capability and performance can be predicted because the capability of the people management system to produce improvements in the knowledge and skills of the organization is known quantitatively. This level of people management capability provides the organization with an important predictor of trends in its business capability.

Level 5 – Innovate

At this level, the entire organization is focused on continually improving the organization's people management capability. The organization has the means to identify opportunities to strengthen its people management practices proactively. Data on the effectiveness of people management practices is used to conduct analyses of potential improvements resulting from innovative people management practices, technologies, or proposed changes to existing practices. Innovative

technologies or practices that demonstrate the greatest potential for improvement are identified and transferred throughout the organization.

The people management capability of optimizing organizations is continuously improving because Level 5 organizations are perpetually improving their people management practices. Improvement occurs both by incremental advancements in their existing people management practices and by adopting innovative new practices and methods. The culture created in a Level 5 organization is one in which every member of the staff is striving to improve their own, their team's, and their unit's knowledge, skills, and motivation in order to improve the organization's overall performance. The people management system is honed to create a culture of performance excellence.

Tactics for Maturing the Deliverables Processes

Information services produce both products and services. This category, while called products, includes both products and services. The products and services are the deliverables required by the end user. The requirements for these products and services need to be understood by the information professionals and then converted into a format suitable for building, maintaining, and operating information systems.

Requirements are the most risk-prone part of information services. This risk can occur because the end user fails to disclose the complete or correct requirements, as well as the inability of the information professional to correctly gather and document the complete and correct requirements. The challenge of gathering requirements correctly and completely the first time is growing as information becomes a major strategic weapon of an organization. In the early days of computers, the systems were primarily a conversion from manual systems to automated systems. Today, through better understanding of how to use information, as well as reengineering efforts, business processes are being innovated, meaning the precise requirements for information evolve as the business process evolves.

There are two components associated with requirements. The first is defining the requirements, and the second is the method used to collect and document the requirements. QAI believes that the most serious risk is defining the requirements. The maturity of the deliverables processes will help define the totality of requirements needed for an information system.

Figure 43 and Figure 44 illustrate the five levels of maturing the deliverables processes. The tactics and practices commonly used at each of the five levels are shown below and are described in the following paragraphs.

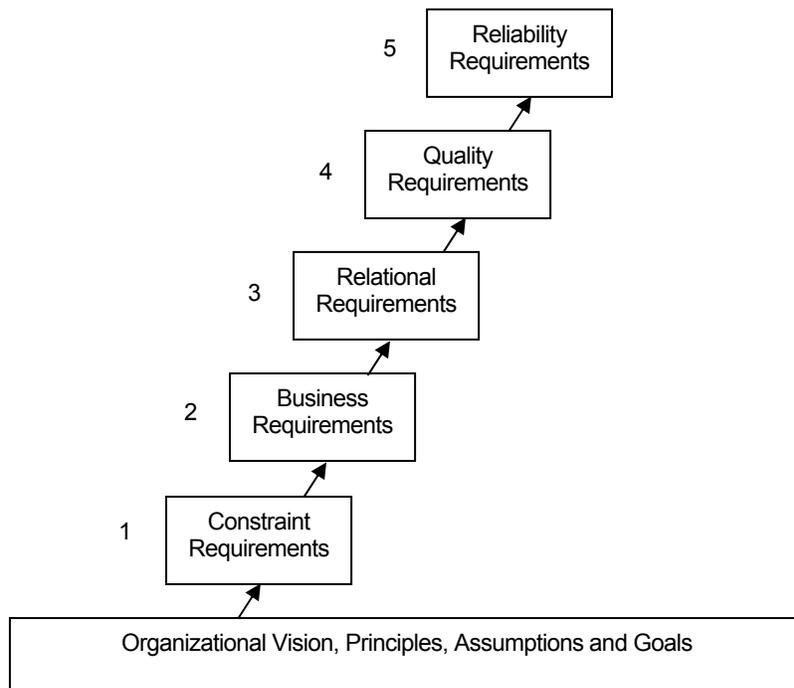


Figure 43. Tactics for Maturing the Deliverables Processes

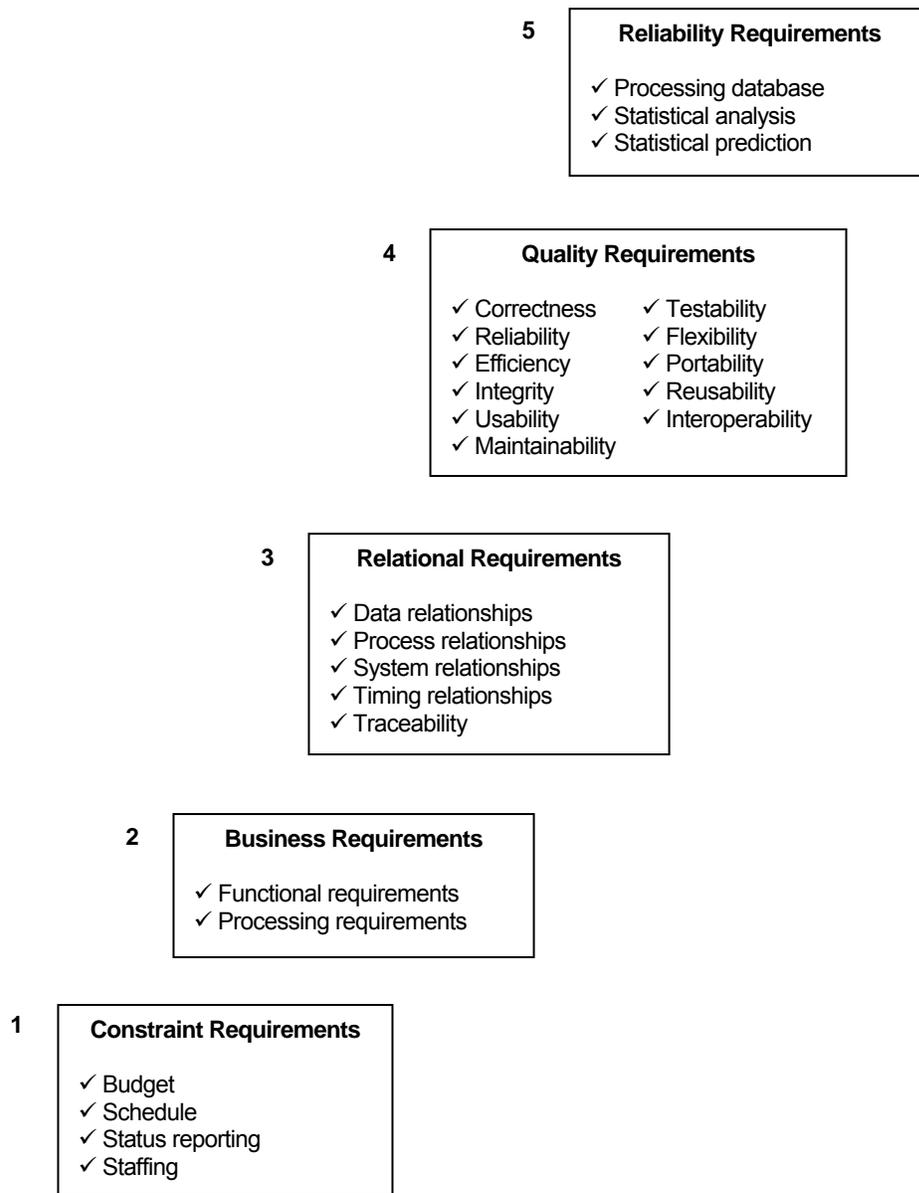


Figure 44. Best Practices for Deliveries Process

Level 1 – Constraint Requirements

This level involves staffing, budget, and schedule. The business requirements are frequently defined as a set of objectives described on several pages of narrative description. Thus, it becomes the role of the project team to expand the initial definition and build the software. However, management and the end user of the software will manage more by the constraint requirements than by the business requirements. The result is that the software put into production on the scheduled date may miss many key requirements, which will need to be installed after the operational date but under new budgets and schedules.

Level 2 – Business Requirements

This level places its emphasis on the functional and operating requirements. Level 2 establishes the criteria that the deliverables must meet in order to be accepted by the end user. At this level, standards are established for the requirements document which specifies the totality of requirement attributes that must be included for the requirements to be considered complete. These often include defining how the requirements will be tested, the objective of the requirement, the problem that the requirement is designed to solve, a detailed description of the requirement, and how the requirement is related to the objectives to be accomplished. Many information organizations at this level introduce uniquely identifying requirements, and then utilize processes that trace requirements through the entire software-building process.

Level 3 – Relational Requirements

This level emphasizes relationships within an information system and between information systems. Level 2 defines relationships, but normally at a much lower level than Level 3. Level 2 relationship specifications tend to be within an individual application system. Level 3 relationships focus on organizational databases, processing sequences and priorities, intersystem relationships, and processing cycle relationships. Data relationships help eliminate data redundancy and help assure consistency of use of data throughout an organization. Processing relationships address processing priorities and processing sequences. The system relationships deal with interfaces between systems and changes to systems. The larger challenge is assuring that all systems are changed concurrently when they are impacted by a common change(s). The processing cycle or timing relationship assures that all data that belongs in a particular accounting processing cycle is processed during that cycle.

Level 4 – Quality Requirements

Quality requirements are the characteristics surrounding the functional requirements that facilitate end user satisfaction. An example of quality requirements is ease of use. The functional requirements deal with the data specified for a particular product, but if it is very difficult to use (i.e., not meeting the quality ease of use requirement); the end user tends to be dissatisfied. Table 12 shows a listing of the more common quality requirements.

Table 12. More Common Quality Requirements

Factor	Definition
Correctness	Extent to which a program satisfies its specifications and fulfills the user's mission objectives.
Reliability	Extent to which a program can be expected to perform its intended function with required precision.
Efficiency	The amount of computing resources and code required by a program to perform a function.
Integrity	Extent to which access to software or data by unauthorized persons can be controlled.
Usability	Effort required learning, operating, preparing input, and interpreting output of a program.
Maintainability	Effort required locating and fixing an error in an operational program.
Testability	Effort required testing a program to ensure it performs its intended function.
Flexibility	Effort required modifying an operational program.
Portability	Effort required transferring a program from one hardware configuration and/or software system environment to another.
Reusability	Extent to which a program can be used in other applications – related to the packaging and scope of the functions that program performs.
Interoperability	Effort required to couple one system with another.

Level 5 – Reliability Requirements

Reliability primarily deals with mean time between failures. To achieve reliability requirements necessitates building a database of processing information and performing adequate statistical analysis on that data. Then a statistical prediction of reliability can be made. While the end user needs to specify reliability requirements, they can only be measured and achieved through the use of statistical tools. Figure 44 shows the best practices for deliveries process.

Tactics for Maturing the Technology Processes

The management of technology is an important category of processes. The technology frequently dictates the type of processes and staffing needed to fulfill the information organization's mission. However, the mission should drive the selection of technology rather than selecting technology and then determining how to use it to accomplish the mission. The exception to this rule is technological upgrades needed to maintain ongoing vendor support for software and maintenance.

The accelerating rate of change introducing new technology necessitates maturing each generation of new technology through the five phases. Thus, with the technology category of processes, there is a constant reversion to Level 1 and movement through Level 5 with each new breakthrough in technology.

Figure 45 and Figure 46 illustrate the five levels of maturing the technology processes. The tactics and practices commonly used at each of the five levels are shown below and are described in the following paragraphs.

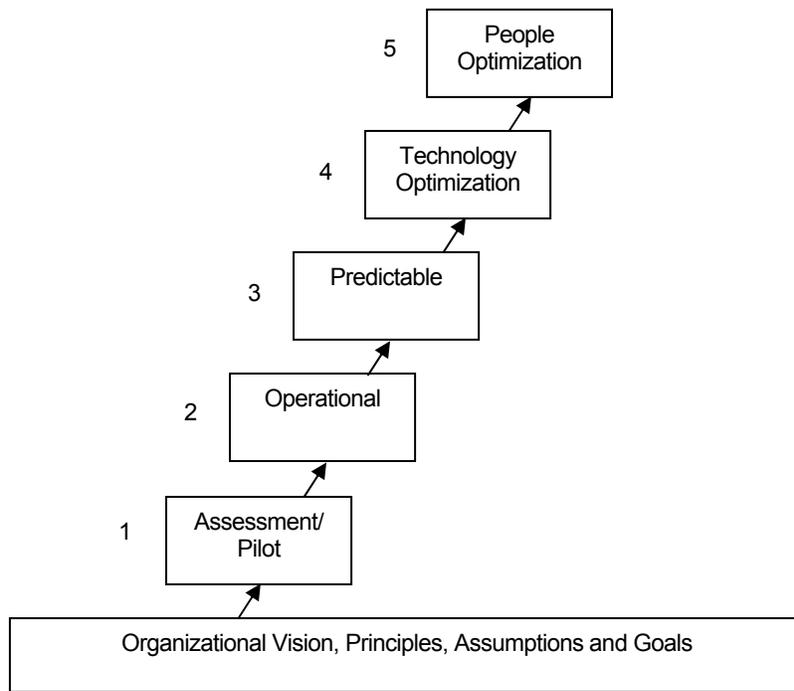


Figure 45. Tactics for Maturing the Technology Processes

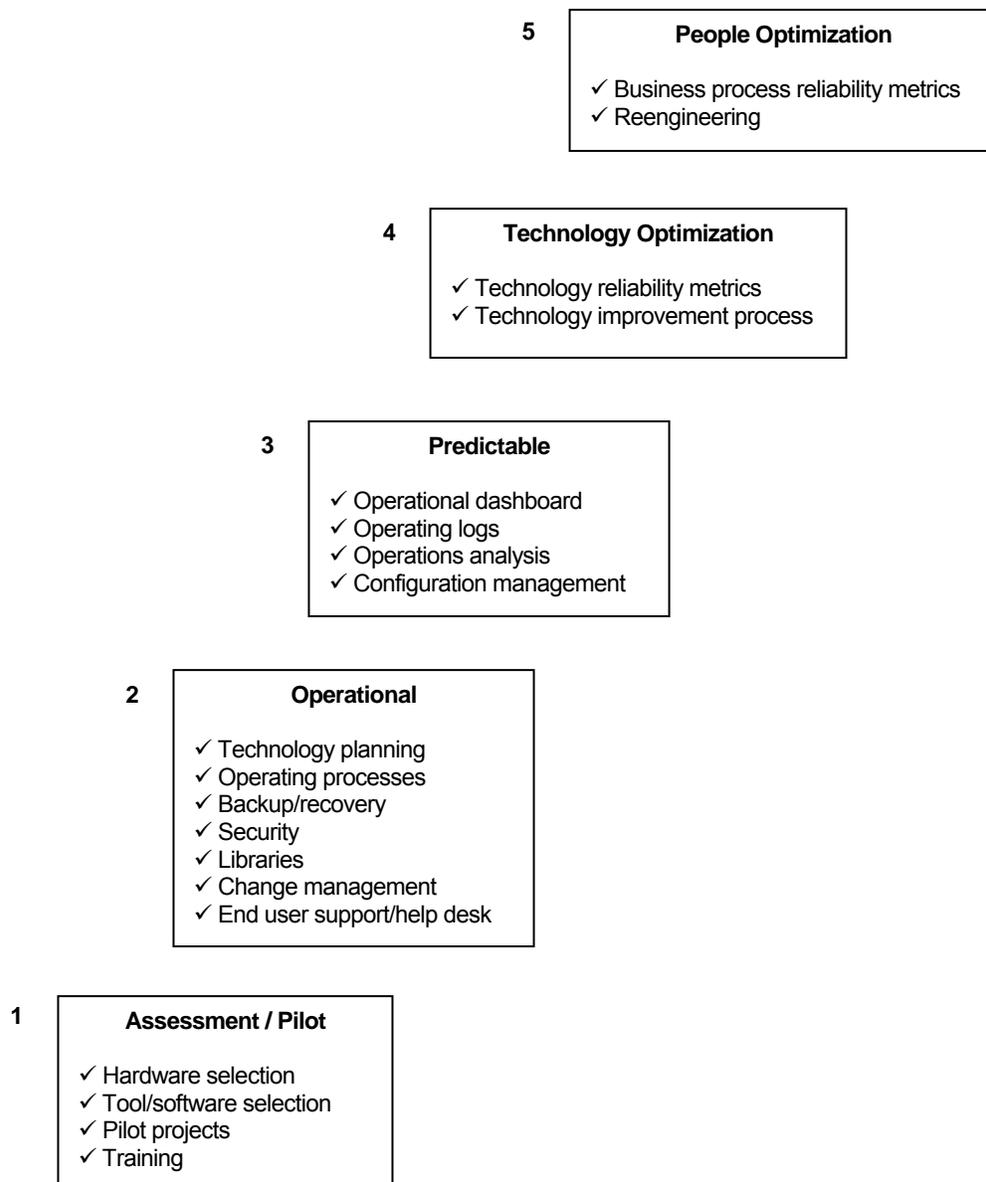


Figure 46. Best Practices for the Technology Processes

Level 1 – Assessment/Pilot

At this level, the organization is evaluating new technology using assessments and pilot projects. The objectives are to determine the value of the new technology and how to control the technology. The practices include hardware selection, tools/software selection, pilot projects, and training in a new technology.

Level 2 – Operational

The new technology may need to be coupled with existing technology or may replace existing technology. The practices necessary to make new technology operational include operating

processes, backup/recovery processes, security processes, library processes, and change management processes. At this level, effective technology planning occurs to address capacity, upgrades, and selection of new technology. There are frequently some changes in infrastructure such as the formation of a security committee. An important process that begins at Level 1, but matures at Level 2, is end user service/help desk.

Level 3 – Predictable

The operations level establishes the processes needed for the introduction of quantitative data and management by fact. Measurement occurs first in technology, because the operating environment must be predictable prior to applying the same quantitative techniques to the software. The techniques that are effective in moving technology to a predictable level are the use of operating logs, analysis of the operation data contained on those logs, and the development of an operational dashboard/use of key metrics. The infrastructure change that occurs at this level is usually the organization of a configuration management group to manage change across all applications and operational platforms.

Level 4 – Technology Optimization

The technology, or operating environment, includes hardware, operating software, database technology, servers, communication lines and associated communication software, security systems, backup hardware and software, and multiple libraries. Emphasis at lower levels is on operating capacity using concepts such as MIPS (millions of instructions processed per second). At Level 4, the emphasis is on throughput. Practices at this level include the use of reliability metrics, and technology improvement processes.

Level 5 – People Optimization

The concern at this level is more related to the effectiveness of the combination of people plus technology. For example, in a banking environment the people optimization concern would be whether the combination of a loan officer plus the technology available to that loan officer can make good loans, as opposed to the type of information and delivery of information to the loan officer. Practices at this level include business process reliability metrics and reengineering practices as shown in Figure 46 above.

Tactics for Maturing the Quality Assurance Processes

Continuous process improvement is based on many small, evolutionary steps rather than revolutionary innovations. The quality assurance processes provide a framework for organizing these evolutionary steps into five maturity levels that lay successive foundations for continuous process improvement. These five maturity levels define an ordinal scale for measuring the maturity of an organization's software process and for evaluating its software process capability.

A maturity level is a well-defined evolutionary plateau on the path toward becoming a mature software organization. Each maturity level provides a layer in the foundation for continuous process improvement. Each level comprises a set of process goals that, when satisfied, stabilize an important component of the software process. Achieving each level of the maturity framework establishes a different component in the software process, resulting in an increase in the process capability of the organization.

Organizing the process management processes into the five levels shown in Figure 47 prioritizes improvement actions for increasing software process maturity. The boxes shown below indicate the type of process capability being institutionalized by the organization at each step of the maturity framework.

Figure 47 and Figure 48 illustrate the five levels of maturing the quality assurance processes. The tactics and practices commonly used at each of the five levels are shown below and are described in the following paragraphs.

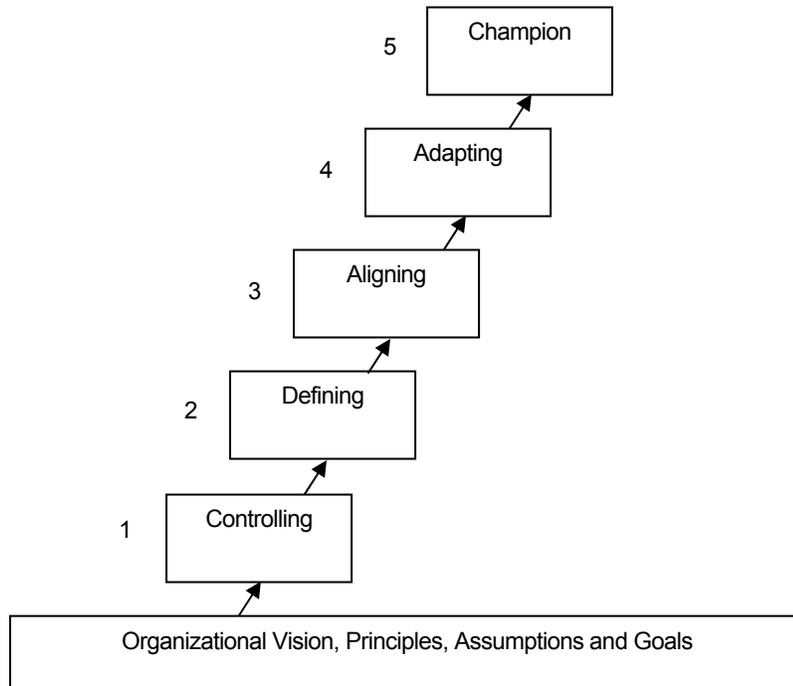


Figure 47. Tactics for Maturing the Quality Assurance Processes

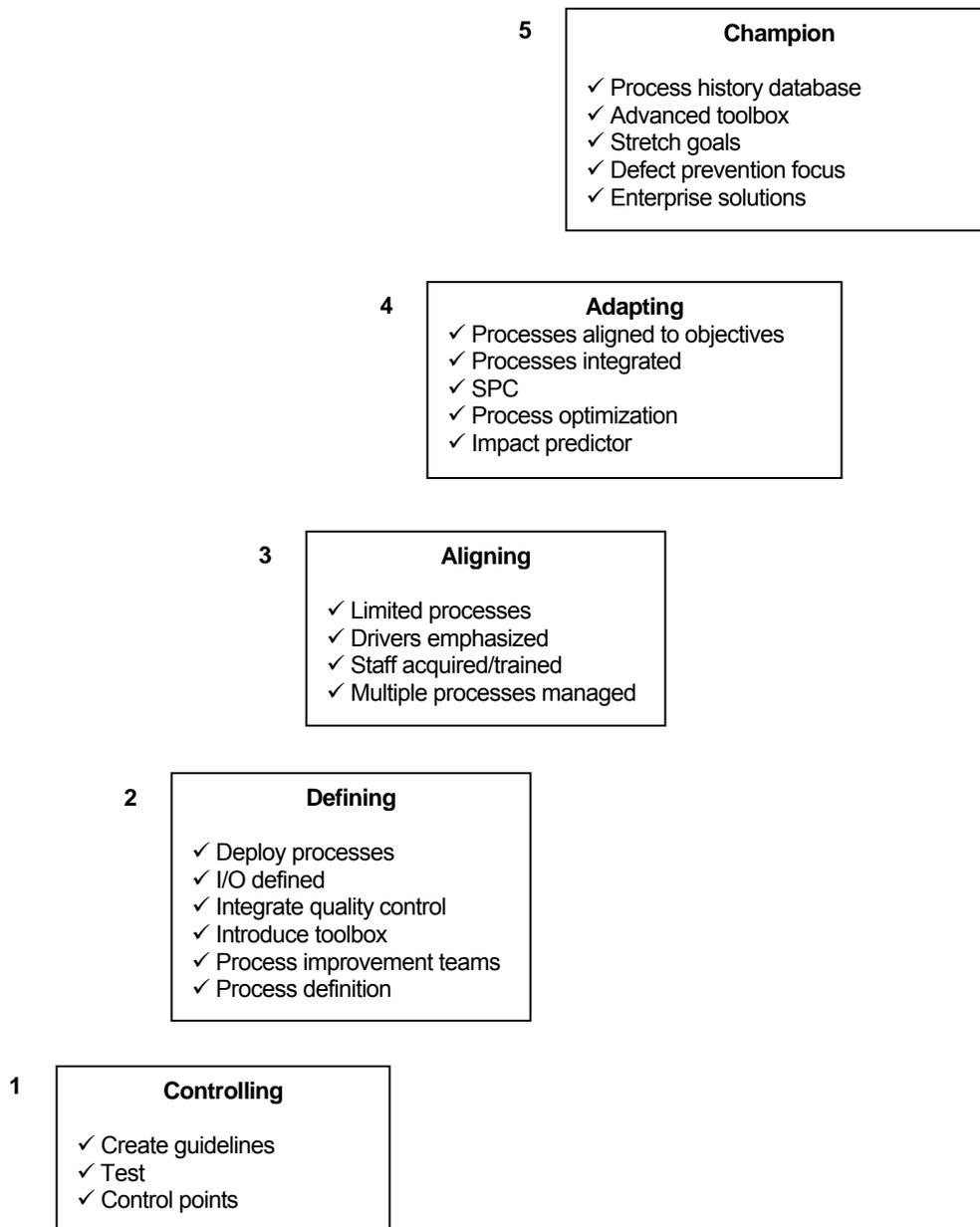


Figure 48. Best Practices for Quality Assurance Processes

Maturity Levels 2 through 5 can be characterized through the activities performed by the organization to establish or improve the software process, by activities performed on each project, and by the resulting process capability across projects as shown in Figure 48. A behavioral characterization of Level 1 is included to establish a base of comparison for process improvements at higher maturity levels.

Level 1 – Controlling

The role of the quality professional at this level is to assure that the people in the IS function do what management wants them to do. At this level, the quality professional is frequently viewed as an inspector or police person.

The types of activities that the quality professional frequently performs at Level 1 include:

- Management establishes predetermined control points or checkpoints and has the quality professional verified that what should have been done at that checkpoint has, in fact, been done. At Level 1 the quality professional will frequently have to "sign off" that the appropriate work products have been produced.
- Perform tests at various levels to validate the functioning of the completed products.
- Develop and disseminate work processes to the workers, but because of the management philosophy at Level 1 these are rarely enforceable and become guidelines for doing work.

Level 2 -- Defining

The role of the quality professional at Level 2 is to create and improve individual work processes. The quality professional should maintain an inventory of work processes, determine which processes are more critical to accomplish the mission, and then develop and/or improve those processes in the sequence of importance. There is normally an infrastructure established to do this, which may be called a process management committee or a standards committee. The committee is supported by individual work groups, which will define and document the work processes.

The activities that are included in the Level 2 role include:

- The quality professional will assist in developing the process for process definition. This will include both the infrastructure and the work tasks.
- The products and services to be received and produced by the information services function need to be defined.
- The methods for controlling the work processes must be defined and integrated into the work processes.
- Set up, perform, or participate in a variety of activities to examine the work products at predetermined points during the project to verify that they meet standards and/or requirements.
- The tools that will be needed by the workers to perform the work more effectively, as well as those needed to define and improve processes, need to be incorporated into a toolbox and taught to all the members of the IS function.
- A formal method must be established and given to the workers who will determine where processes are ineffective and then make the necessary changes to improve the efficiency and effectiveness of the work processes.

Level 3 -- Aligning

The role of the quality professional at Level 3 is to create an environment, which is focused on meeting organizational needs. In other words, the quality professional must help the IS function define their core competencies. These competencies should be based on the needs of the organization as normally expressed by the user of the IS function. At this level, the energies of the quality professional will be directed toward ensuring that those competencies, which support the IS mission, are supported effectively.

The activities that are performed by the quality professional at Level 3 include:

- Those work processes, which are not supportive of the mission, and/or are not deemed to be within the competencies of the IS function, are eliminated so that the remaining work processes define the core competencies of the IS function.
- The quality professional will help define which work processes/subprocesses are the driving processes so that management can focus their attention on managing those driving processes, as well as improving the driving processes.
- The competencies of the IS function as defined in the work processes will be the basis for hiring and training staff. Staff will be acquired and trained to optimize the competencies of the IS function, as opposed to hiring highly skilled individuals whose skills may or may not be needed.
- The interrelationship of the processes will be defined so that workflow will be facilitated, avoiding bottlenecks caused by failing to identify the critical paths for completing products and services when multiple processes are involved.

Level 4 -- Adapting

The role of the quality professional at Level 4 is to emphasize adapting the processes to the specific project needs. In other words, prior to commencing a project, the processes will be customized for that specific project. This is equivalent in automobile manufacturing to rearranging the production line to build a specific automotive model. The quality professional will assist in creating the processes and skills needed to adapt the work processes to specific product needs.

The activities that are performed at Level 4 include:

- Work processes are defined in a generic mode. In other words, there is a process for defining requirements, a process for testing a software system, and so forth. These will need to be modified based on specific objectives. In other words, defining or testing requirements when the objective is for high degrees of accuracy in results will be different from when a major objective for requirements is ease of use, with high degrees of accuracy not necessary.
- Processes, which were defined independently, now, must be integrated. For example, the system development process and the test process, which would have been independently developed at Level 2, become integrated at Level 4. Figure 49 shows the integrated process model for QAI's categories of processes.
- The stability of processes enables the effective use of quantitative measurement for management. At the highest level, these are strategic and tactical dashboards

supported by many lower-level measures to enable managers to determine whether the process is in or out of control and the type of adjustments that are necessary to bring the process back into control.

- Quantitative measurement indicates effectiveness in performing processes which, once identified, can be eliminated or minimized to optimize process performance.
- Quantitative data, plus statistical methods, enable managers to document cause-effect relationships. The cause, once identified, can then be used to predict the impact.

Level 5 -- Champion

The role of the quality professional at Level 5 is to champion innovation and creativity in the way work is performed. Professional work processes are a combination of the skill sets of the individual, plus the detailed step-by-step procedures in the work process as shown in Figure 50. In the start of a new work process, the procedures are minimally defined and thus the success of a project is heavily dependent on the skill set of the individuals. However, over time the combined learning of the individuals performing the process can be incorporated into the process. Thus, as processes mature they move toward more emphasis on the process and less emphasis on the need for highly skilled individuals. By the fourth level of maturity, the processes are highly optimized. Improvements from Level 4 optimization status require innovation and creativity. However, it knows the limits of optimization that enables innovation and creativity to occur.

The activities that are involved in the innovation and creativity level include:

- A history of the performance levels of processes for use in analysis.
- The more advanced statistical tools can now be used because of the reliability of the quantitative data collected at Level 4. These advanced tools include such items as regression analysis and the Rayleigh Curve.
- Establishing goals that appear unachievable causes people to develop innovative, creative solutions to achieve those goals.
- Rather than emphasizing defect identification and correction, processes are restructured to prevent defects from occurring.
- This emphasis is on optimizing the IS processes with processes throughout the organization. In other words, rather than optimizing IS processes exclusively, there is a totality of processes within and outside the IS function that become optimized in totality.

Tactics for Maturing the Quality Control Management Processes

The two components of work are doing the work and then checking the work. These need to be separate functions but performed consecutively. Generally, the more iterations of do and check that occur in a processing cycle, the lower the cost for building the product. The reason for this is that repeating the do/check cycle frequently catches defects close to the point where they occur.

As the control/test processes mature, emphasis switches from detecting defects to preventing them. While prevention is the desirable level, it generally cannot be achieved until organizations

build experience in detecting defects and performing enough analysis to develop defect patterns. Organizations at Levels 4 and 5 of this control/test model normally have defect profiles showing the type and frequency of defects that occur during various tasks in the processing cycle.

Risk is the probability that something might go wrong. Control is the means organizations use to reduce risk. Test is a form of control. Thus, if there is no risk there is no need for control or test. This concept is critical in understanding the entire control/test process maturity. The lower levels tend to be the less effective controls, while the higher levels are the more effective controls.

Figure 49 and Figure 50 illustrate the five levels of maturing the quality control processes. The tactics and practices commonly used at each of the five levels are shown below and are described in the following paragraphs.

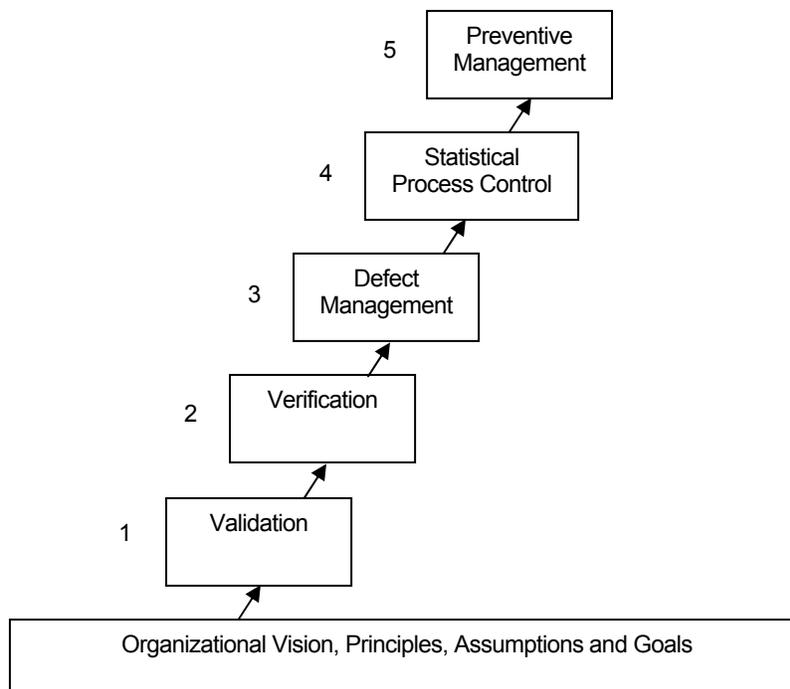


Figure 49. Tactics for Maturing the Quality Control Processes

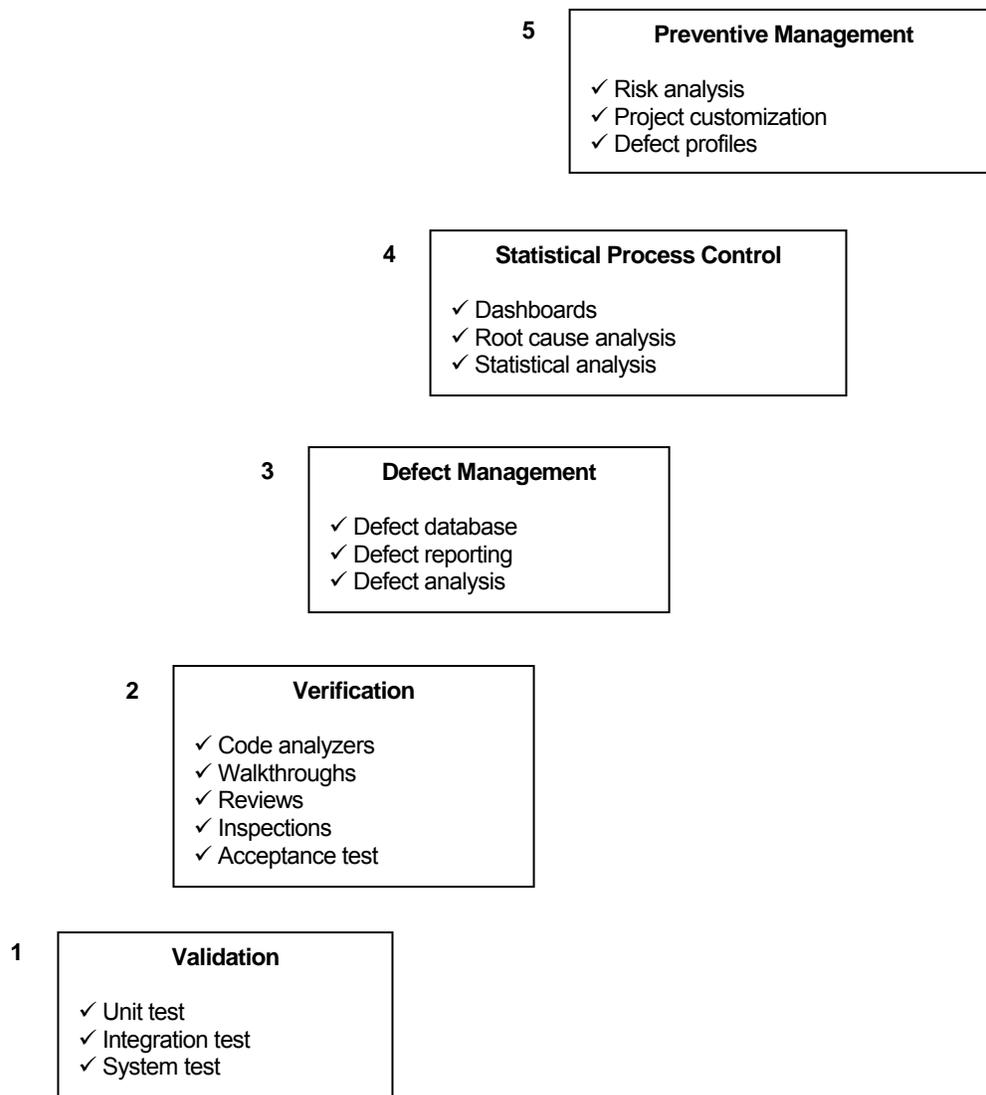


Figure 50. Best Practices for Quality Control Management Processes

Level 1 – Validation

Validation is testing the software in an operational state. While there are several iterations of testing, they do not occur until the latter part of the system development life cycle. Thus, validation does not take advantage of the control theory stating that the control should be placed as close as possible to where the defect occurs. The most common types of validation are unit testing, integration testing, and system testing. Unit testing tests the smallest operating module of a system; integration testing tests the connecting logic between the modules/units; and the system testing validates that the system executes in the organization's operating environment.

Level 2 – Verification

Verification is the static testing of the system in a nonoperational status. It primarily tests the documentation produced by the project team. The documentation includes requirements, design, code, test, and operating documentation. Verification is performed through a variety of processes, including code analyzers, walkthroughs, reviews, and inspections. These verification processes also interact with the project team, end users, and other interested parties. Level 2 also incorporates acceptance testing, in which the end users define the criteria, which the system must meet in order to, be acceptable, and then test against those criteria.

Level 3 -- Defect Management

This level involves naming defects, counting defects, recording defects in a database, analyzing them, and then managing them throughout the entire life cycle of software. There are two types of defects, just as there are two types of quality. One defect is a variance from requirements/specifications, and the second defect is anything that dissatisfies the end user. For example, if the system was to add A plus B to get C, but in fact added A to X that would be a variance from specification defect. If the output was correct, but hard to use from an end user viewpoint, it would be an ease of use defect. Defects are generally only considered to be defects when they are uncovered in a task beyond the task in which the defect occurred. If the individual who makes the defect finds it during the same task, it is generally not considered to be a defect.

Since defects provide a trail of the processing problems, analysis of those defects can lead to the means by which processes can be improved. One of the reasons that defect management does not occur effectively until Level 3 is that employees at lower levels are concerned that by recording defects they will negatively impact their performance appraisal.

The major processes that occur during defect management are the naming and recording of defects, the creation and operation of a defect database, the analysis of defects, and the reporting of defects to the involved managers so that they have the information needed to make process improvements.

Level 4 -- Statistical Process Control

To be effective, statistical process control needs stabilized processes and defect recording. Some statistical process control techniques will be introduced at Level 3, but their use will be confined more to process improvement than for process management. Good statistical process control techniques require the type of measurement data that is only produced at Level 4. The statistical process control techniques will expand on root cause analysis, use sophisticated statistical tools and analysis, and create dashboards to be used in managing processes.

Level 5 - Preventive Management

This level is focused on preventing defects. It uses the lessons learned from the previous four levels of control/test to address the problems before they can become defects. During the previous four levels, the causes of defects are available and some addressed through process improvement. This level accelerates that trend and in addition provides the professional information staff with methods to modify the way work is performed so that the root causes of defects will not be built into software. This involves building risk models, developing defect expectation charts and/or

defect profiles, and processes to customize processes so that the do component of processes can build software that is almost defect free.

How to Plan the Sequence for Implementing Process Maturity

The QAI Manage by Processes Tactical model identifies six process categories. As explained earlier, the six were selected because they are frequently managed by six different individuals or groups. This section explains some of the relationships and strategies that are important to understand in maturing an information services function, and thus how to sequence the implementation of the maturity of the six process categories.

Twelve relationships are presented for use in maturing your IT function to meet your specific improvement objectives and timetable. For each relationship, a strategy is provided on the sequencing of maturity, as well as a discussion on skipping levels and reverting back to lower levels. The twelve relationships are:

- People skills and process definitions
- Do and check procedures
- Individuals' assessment of how they are evaluated to work performed
- What management relies on for success
- Maturity level to cost to do work
- Process maturity to defect rates
- Process maturity and cycle time
- Process maturity and end user satisfaction
- Process maturity and staff job satisfaction
- Process maturity to an organization's willingness to embrace change
- Tools to process maturity
- Control/test process category and quick paybacks

Relationship between People Skills and Process Definitions

Professional work processes are a combination of the skill sets of the individual performing the work process plus the procedures and standards within the work process. Since there is a continuum of work processes, from well-defined and routine work processes to highly creative work processes, the mix of written procedures and people skills change as this continuum moves. Figure 51 illustrates that the defined routine processes are comprised primarily of detailed written procedures to be followed with minimal variation, while at the creative level the procedures are very cryptic and generalized, and they are very dependent upon people skills.

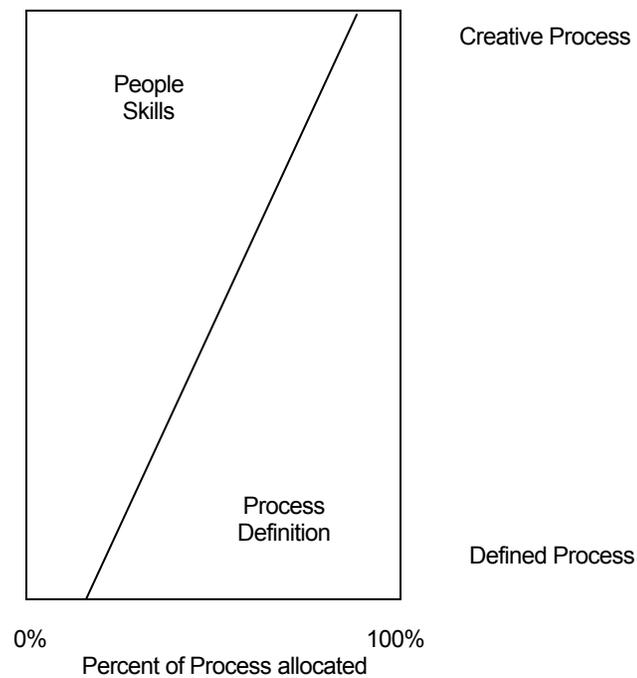


Figure 51. Performing a Work Task Definition versus People

Relationship of Do and Check Procedures

Work processes are a combination of Do and Check procedures. The worker is told what to do, and then how to check that what was done was done correctly. Figure 52 shows that the mix of Do and Check procedures changes as processes mature from defined routine to highly creative. For the defined routine work processes, the Do procedures are very detailed and are designed to be followed with minimal variation. This is because organizations know how to do the routine processes. As processes move toward creative, there is less knowledge on how to do it, but the ability to recognize whether it is done correctly exists. Thus, for creative processes there may be minimal guidance on how to do it, but well-defined processes to check to see if it is done right, can be developed. Many of these checking processes involve groups of peers.

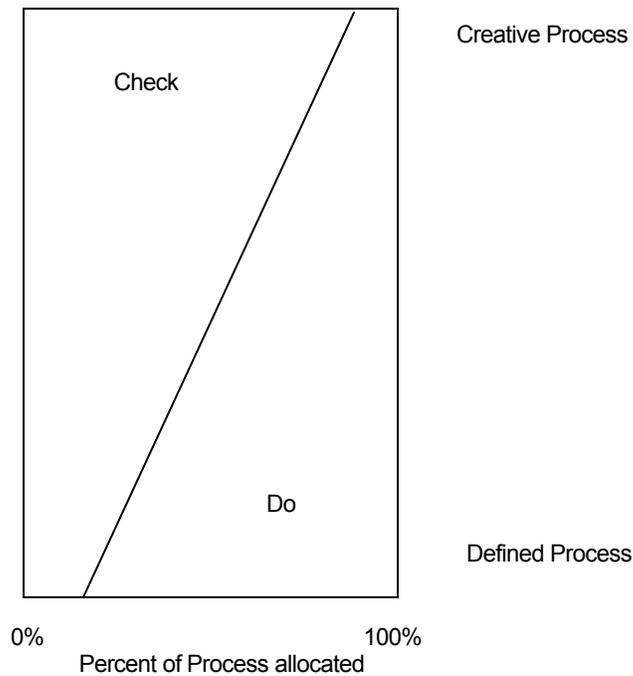


Figure 52. Performing a Work Task Do versus Check

Relationship of Individuals' Assessment of How They are Evaluated to Work Performed

People do what they believe they are evaluated on. If they are evaluated on meeting schedules, they meet schedules; if they are evaluated on producing defect-free code, they produce defect-free code; and so forth. Because immature processes have poorly defined standards of work, most assessments of individual performance at maturity Level 1 are highly subjective. Figure 53 shows that, as processes mature, the assessment can move from subjective to objective, looking at the results produced against the standards for those results. Thus, at low levels of process maturity, people believe they are subjectively evaluated and focus their attention on organizational politics, while at the highest levels of process maturity their emphasis switches to the results they are paid to produce because those results can be measured against specific standards.

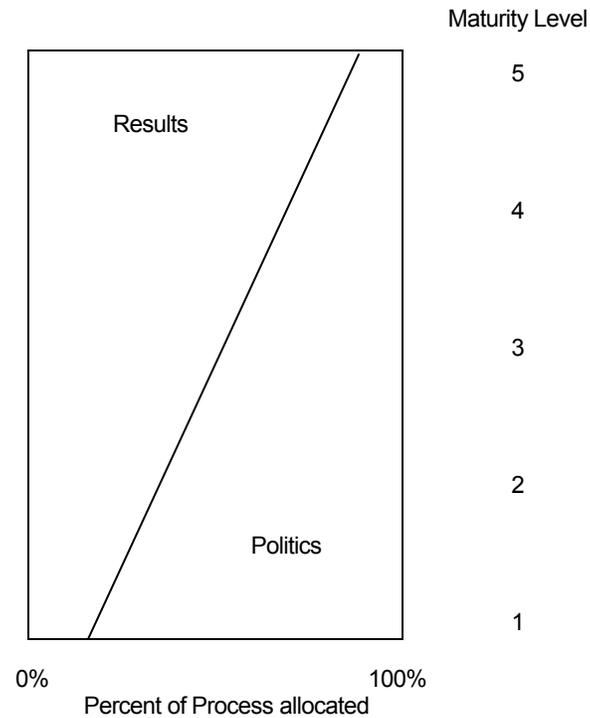


Figure 53. Individual Assessment of What Contributes to Performance Assessment

Relationship of What Management Relies on for Success

Current literature emphasizes the use of teams, empowered teams, and self-directed teams. On the other hand, organizations at low process maturity levels tend to rely much more on individuals, even if those individuals are members of teams. Figure 54 shows that as the maturity level increases, management relies more on teams than individuals.

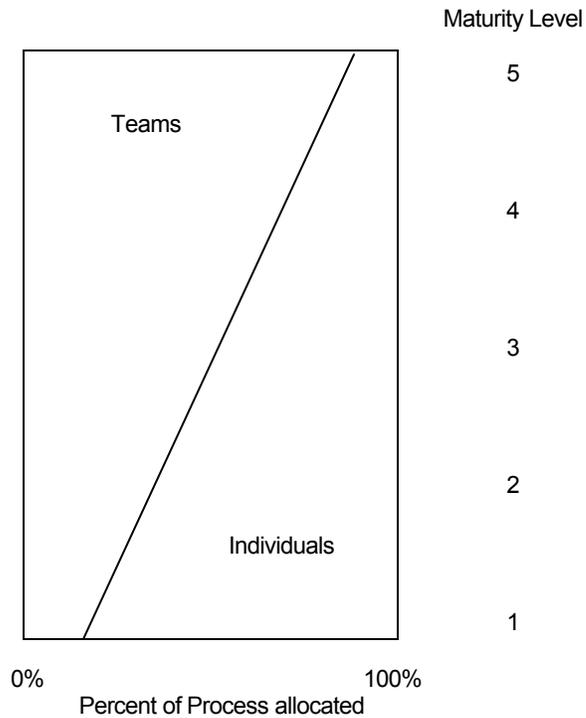


Figure 54. What Management Relies on for Success

Relationship of Maturity Level to Cost to Do Work

Maturity level has a significant impact on cost as shown in Figure 55. As the maturity level increases, the cost per unit of work decreases. Many organizations now measure cost in dollars to produce a function point. (Note: They could also use cost to produce a thousand lines of code.) It has been estimated that an increase of one level in process maturity doubles an organization's productivity.

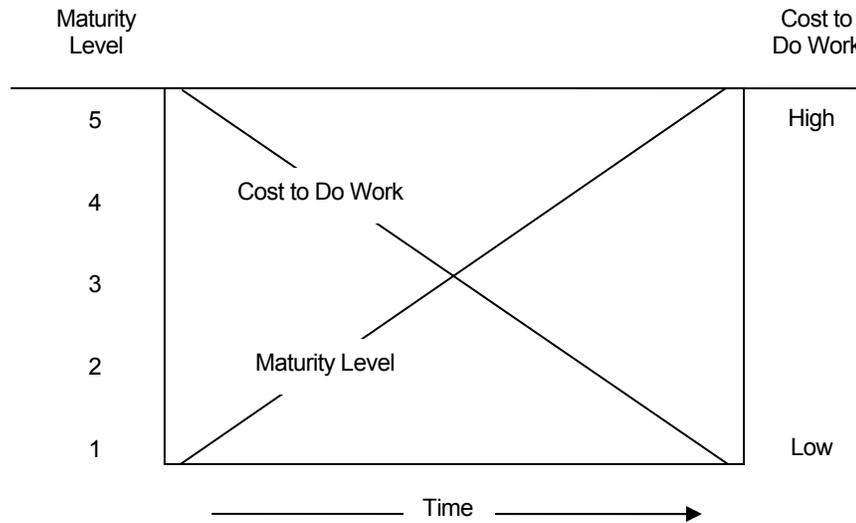


Figure 55. Relationship of Maturity Level to Cost to Do Work

Relationship of Process Maturity to Defect Rates

There is a strong correlation between process maturity and defect rates as shown in Figure 56. As the process maturity level increases, the defect rate decreases. Many organizations measure defect rate in defects per thousand function points or defects per thousand lines of code. As processes become better defined, controls become more effective, and as people become more motivated, the defect rate drops significantly.

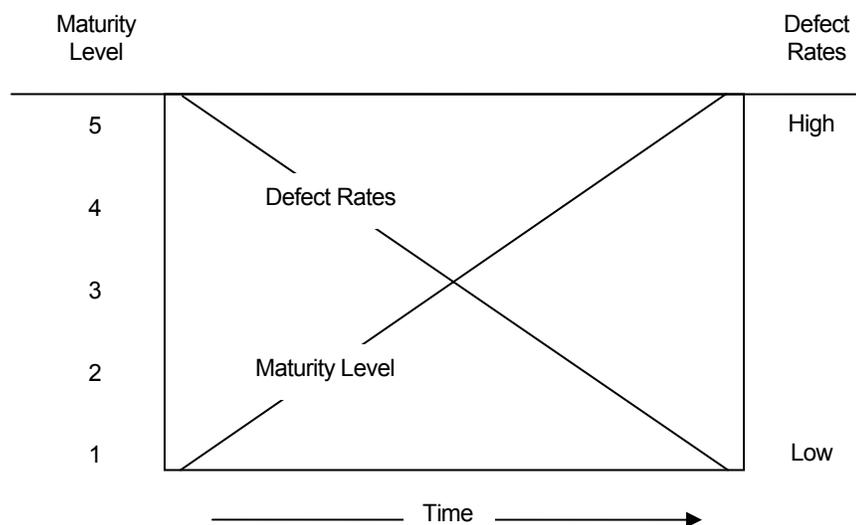


Figure 56. Relationship of Maturity Level to Defect Rates

Relationship of Process Maturity and Cycle Time

There is a high correlation between process maturity and cycle time. As shown in Figure 57, as processes mature, there is a decrease in the cycle time to build software products. The maturity of processes, people, and controls has associated with it a search for the root cause of problems. It is these problems that lead to rework which, in turn, extends the cycle time. Thus, improvements also focus on facilitating transitions of deliverables from work task to work task, which also significantly reduces cycle time.

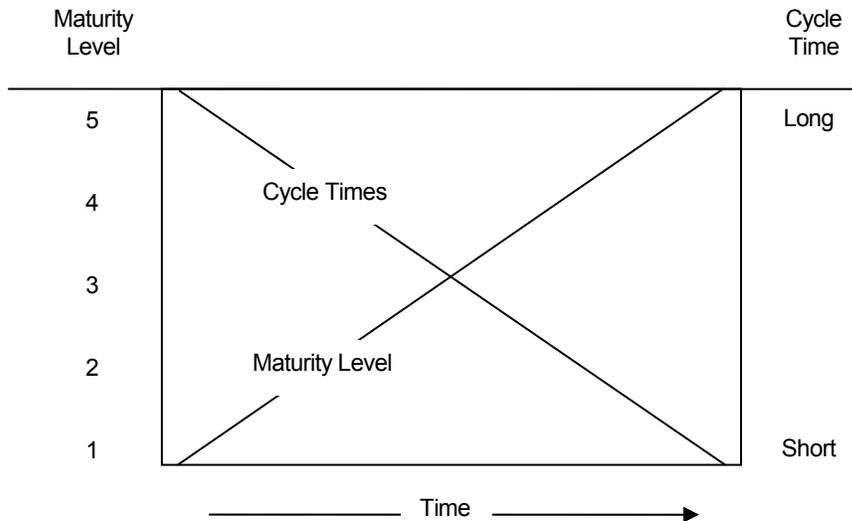


Figure 57. Relationship of Process Maturity to Cycle Time

Relationship of Process Maturity and End User Satisfaction

There is significant research to support the premise that end user satisfaction increases as process maturity increases. End users are dissatisfied for many reasons, which are addressed by process maturity. One is variability, meaning that the service or product they receive one time is significantly different from product received at a later point in time. As shown in Figure 58, process maturity levels 2 through 5 greatly reduce this variability. End users are also dissatisfied by high costs and long cycle time, which are both reduced by process maturity. In addition, end users are dissatisfied when the information system does not provide significant value in accomplishing their mission. Levels 3 through 5 have a significant impact on value received from the information areas.

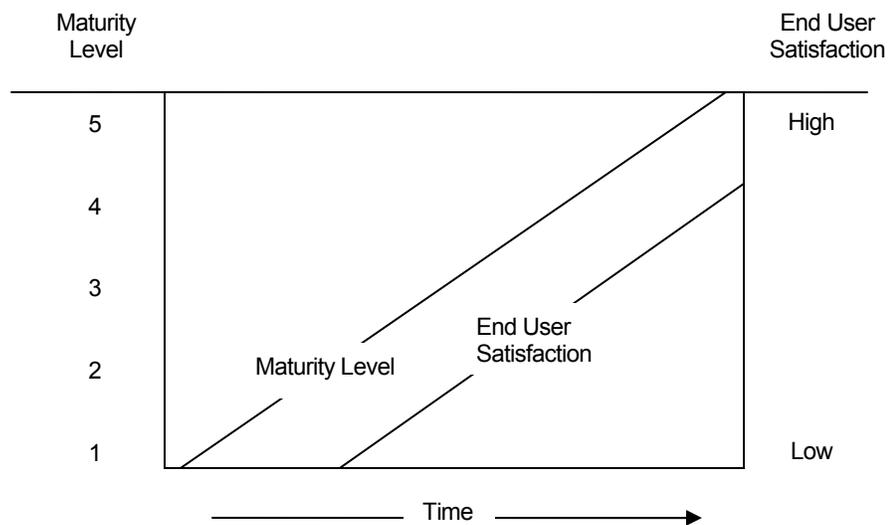


Figure 58. Relationship of Process Maturity to End User Satisfaction

Relationship of Process Maturity and Staff Job Satisfaction

Staff job satisfaction increases significantly as processes mature as shown in Figure 59. The reason for this is that stabilized effective processes increase an individual's ability to be successful. The facts that process maturity increases end user satisfaction, reduces defect rates, and reduces rework and cost, are all contributors to an individual's personal success. Individuals tend to want to work in an organization in which they are successful, and if the organization's processes make them successful they are motivated and anxious to stay with that organization.

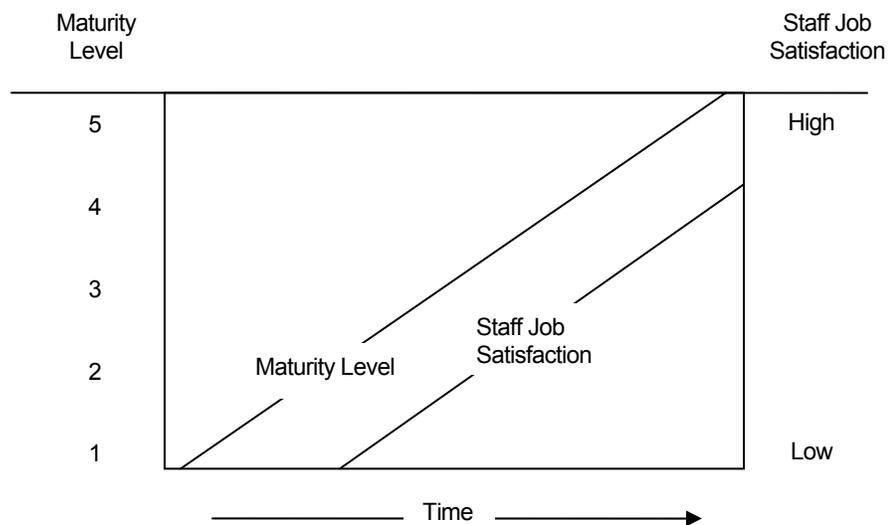


Figure 59. Relationship of Maturity Level to Staff Job Satisfaction

Relationship of Process Maturity to an Organization's Willingness to Embrace Change

There is a high correlation between process maturity and an organization's willingness to embrace change as shown in Figure 60. People resist change for a variety of reasons. Some of these are personal, but others relate to a lack of confidence that the change will both improve the way work is done and have a positive impact on their career. However, as processes mature, people gain confidence in the processes and the willingness of management to reward for results. The willingness to change parallels process maturity but lags slightly at the lower levels of maturity and accelerates during the higher levels of maturity.

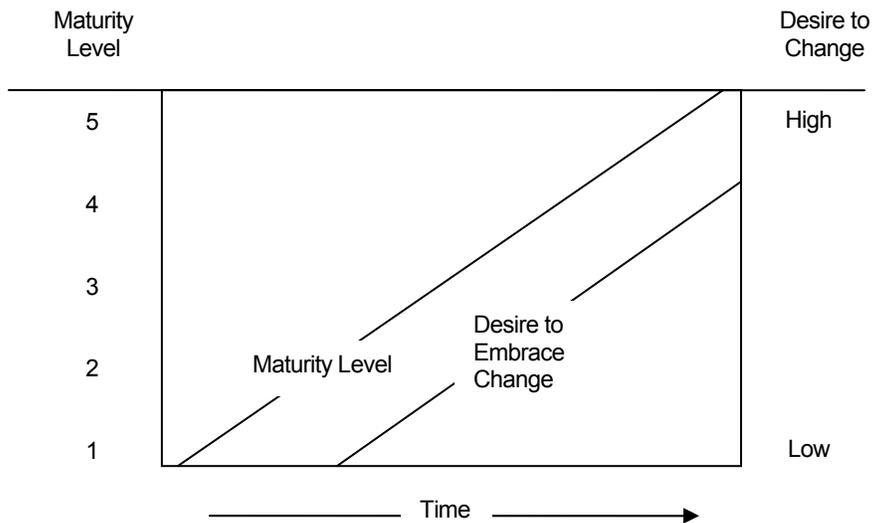


Figure 60. Relationship of Process Maturity to an Organization's Willingness to Embrace Change

Relationship of Tools to Process Maturity

Tools are a vital component of maturing processes. At Level 1, tools tend to be optional and not well taught. The lack of good tools, and the lack of consistent use of those tools, holds organizations at lower maturity levels. There is a strong relationship between the acquisition and integration of tools into the work processes and the movement from process maturity Level 1 to Level 5.

Relationship of the Control and Test Process Category to Quick Paybacks

Organizations requiring a quick payback from maturing processes should consider maturing the control and test category first. This generally has no long-term impact on how work is performed, but does have an impact on cost and cycle time. This is because defects can be found closer to their source, which enables the defect to be corrected more quickly and at lower cost. Studies have shown that the cost to correct a defect can increase by a factor of up to 100 times by waiting until later phases of the life cycle to correct the defect.

Strategy for Moving to Higher Maturity Levels

An understanding of the relationship between the six process categories is helpful in developing a strategy to mature to higher maturity levels. At Level 1, the six process categories tend to be competing for management attention, and with one another. In fact, in many organizations management may want to use one category, such as control, against another, such as process.

In moving to Level 2, there is a preferred sequence to mature the six process categories as shown in Table 13. The management processes and the technology processes should be matured first to provide leadership and technology to provide a stable computer operating environment. The product, control and test, and process categories can be moved second. While it is desirable to also move the people category, the people may have to be convinced that process maturity is a worthwhile endeavor before they buy in to the concept. Thus, by having the product, process, and control categories moved to demonstrate the benefits, the people can be convinced that management is serious. Once they are convinced management is serious, the people management process changes are easily accepted by the people.

Table 13. Strategy for Moving to Higher Maturity Levels

Process Category	Sequence for Moving Process Categories	
	From Level 1 to Level 2	From Level 2 to Levels 3, 4, 5
Management Processes	1 st	1 st
Technology Processes	1 st	3 rd
Deliverables Processes	2 nd	3 rd
Quality Control Processes	2 nd	2 nd
Quality Assurance Processes	2 nd	3 rd
People Processes	3 rd	1 st

In moving from Level 2 to higher levels, the management and the people need to move first. These are the leadership and people buy-in categories. Control is moved second, as that tends to provide the data needed on how to mature the other categories and at the same time provides the type of stability and confidence needed to know that maturing will be controlled.

Skipping Levels and Reverting Back to Lower Levels

The theory of levels of maturity is that organizations move from level to level. The theory, supported by practical experience, demonstrates that skipping a level is not practical because each level builds the base for moving to a higher level. In the QAI model, with six process categories, the theory is to move all six process categories approximately concurrently to the next level.

In reality, installing individual work processes at a higher level may be advantageous to an organization, as well as moving one process category prior to moving the remaining process categories. For example, organizations at Level 1 may find some of the measurement practices at Level 4 helpful to introduce discipline. While measurement at Level 1 will not have the same reliability and validity as measurement at Level 4, it may serve a specific purpose. Thus, selected

practices from higher levels may be used effectively at lower levels. However, skipping a level to move to a higher level rarely works.

In the section on relationships of the process categories, the strategy of moving selected categories first is described. It is generally good practice to move one of the categories, for example, the management processes, before moving any of the other categories.

Reversion back to lower levels will occur constantly when there is either significant business or technology change. This does not mean that the information organization drops to a lower level, but for specific technology or business change those changes will have to be begun at a lower level and move rapidly up to the organization's current level. For example, if an organization was at technology Level 3 with legacy systems, but was considering introducing information services project technology, they would revert to technology Level 1 for information services project technology. Likewise, the movement to information services project technology may involve different groups of people and, although the organization had matured to Level 3 in the people category, it may have to revert to people Level 1 for those newly involved in information services project technology. The good news is that when an organization is at a higher level it can move new technology or new business change rapidly through the lower levels to the current level of performance.

Define, Build, Implement and Improve Work Processes

The world is constantly changing. Customers are more knowledgeable and demanding and, therefore, quality and speed of delivery are now critical needs. Companies must constantly improve their ability to produce quality products that add value to their customer base. Defining and continuously improving work processes allows the pace of change to be maintained without negatively impacting the quality of products and services.

<i>Process Management Concepts</i>	281
<i>Process Management Processes</i>	289

Process Management Concepts

Process management is a term used by many IT organizations to represent the totality of activities involved in defining, building, deploying and maintaining the work processes used to achieve the IT mission. What is referred to as process management in this skill category is also called *process engineering* and *the standards program*.

Definition of a Process

A process is a vehicle of communication, specifying the methods used to produce a product or service. It is the set of activities that represent the way work is to be performed. The level of communication (detail of the process) is normally commensurate with the skill level associated with the job. Table 14 shows some sample IT processes and their outputs.

Table 14. Sample IT Processes and Their Outputs

Examples of IT Processes	Process Outputs
Analyze Business Needs	Needs Statement
Conduct JAD Session	JAD Notes
Run Job	Executed Job
Develop Strategic Plan	Strategic Plan
Recognize Individual Performance	Recognized Individual
Conduct Project Status Meeting	Updated status information

Why Processes Are Needed

Processes add value to both management and the workers, although the reasons differ.

From a *management perspective*, processes are needed to:

- Explain to workers how to perform work tasks
- Transfer knowledge from more experienced to less experienced workers
- Assure predictability of work activities so that approximately the same deliverables will be produced with the same resources each time the process is followed
- Establish a basic set of work tasks that can be continuously improved
- Provide a means for involving workers in improving quality, productivity, and customer satisfaction by having workers define and improve their own work processes
- Free management from their activities associated with "expediting work products" to spend more time on activities such as planning, and customer and vendor interaction

From a *worker perspective*, work processes are important to:

- Increase the probability that the deliverables produced will be the desired deliverables
- Put workers in charge of their own destiny because they know the standards by which their work products will be evaluated
- Enable workers to devote their creativity to improving the business instead of having to develop work processes to build products
- Enable workers to better plan their workday because of the predictability resulting from work processes

Process Workbench and Components

A quality management approach is driven through processes. As Figure 61 shows, the workbench is a graphic illustration of a process, documenting how a specific activity is to be performed. Workbenches are also called phases, steps, or tasks. A process can be viewed as one or more workbenches. Depending on the maturity of the organization, process workbenches may be defined by process management (or standards) committees, QA analysts, or work teams.

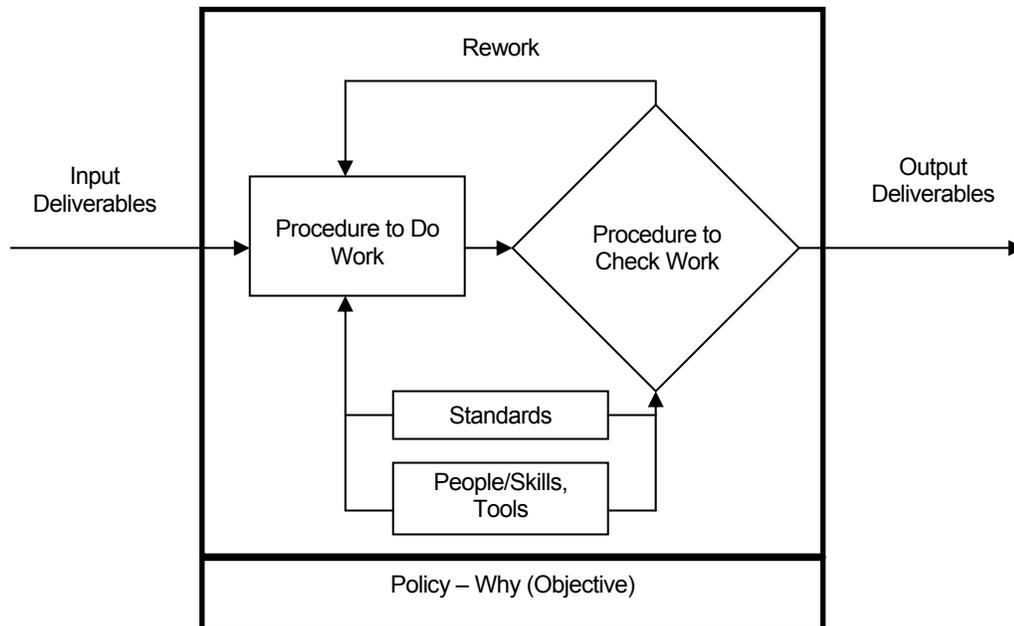


Figure 61. Components of a Process Workbench

From the perspective of the PDCA cycle, the process workbench is created during the Plan phase, and improved during the Act segment.

The workbench transforms the input to produce the output. The workbench is comprised of two procedures: Do and Check, which correspond to the Do and Check phases of the PDCA cycle. If the Check procedure determines that the standards for the output product are not met, the process engages in rework until the output products meet the standards, or management makes a decision to release a nonstandard product. People, skills, and tools support the Do and Check procedures.

A process is defined by workbench and deliverable definitions. A process is written with the assumption that the process owners and other involved parties possess certain skill and knowledge levels (subject matter expertise).

A workbench definition contains:

- A policy statement (why - the intent)

- Standards (what - the rules)
- Procedures (one or more tasks) in the form of procedural statements (how)

A *deliverable definition* contains:

- A policy statement (why – the intent)
- Standards (what – the rules)
- Templates that specify document format
- Policies, standards, and procedures may refer to people, methods, and tools.

Note that some workbenches and deliverables may not contain standards or procedures. It is assumed that if a defined process contains standards, they will be complied with; and if the process contains task procedures, they will be performed.

The following components of a process represent the vocabulary of process management.

Policy

The policy states why a process exists or its purpose. A policy indicates intentions or desirable attributes or outcomes of process performance, and should link to the organization's strategic goals and support customer needs/requirements.

Standards

The standards state what must happen to meet the intent of the policy. Standards may relate to a deliverable produced in the process or to the task procedures within the process. Regarding deliverables, the standard is used to determine that the delivered product is what is needed. Regarding the task procedures, standards may specify things such as the time frame or a sequence that must be followed. A standard must be measurable, attainable, and necessary.

Inputs

Inputs are the entrance criteria or materials needed to perform the work.

Procedures

Procedures describe how work must be done - how methods, tools, techniques, and people are applied to perform a process (transform the input into the output). Procedures indicate the "best way" to meet standards. There are procedures to Do and Check work. People, skills, and tools are incorporated into the Do or Check procedures, and, therefore, are not considered separate components of the workbench.

- People or Skills are the roles (such as suppliers, owners, and customers), responsibilities, and associated skill sets needed to execute a process. For example, a programmer may require written communication skills and knowledge of Visual Basic.

- Manual and automated tools such as CASE tools, checklists, templates, code compilers, capture/playback testing tools, and e-mail may be used to aid in the execution of the procedures.

Output or Deliverables

Output or deliverables are the exit criteria, products, or results produced by the process. Deliverables can be interim or external. Interim deliverables, such as JAD notes, are produced within the workbench, but never passed on to another workbench. External deliverables, such as a requirements specification, may be used by one or more workbenches, and have one or more customers. Deliverables serve as both inputs to, and outputs from, a process.

Process Categories

Approaches for controlling businesses have evolved over many decades. These approaches have been accepted by the American Institute of Certified Public Accountants, chartered accountant societies worldwide, and the U.S. General Accounting Office (which has issued control guidelines for the U.S. Government).

The business control model includes three general categories of control, which are implemented through the processes below. Examples of these controls are given in Skill Category 7.

Management Processes

These are the processes that govern how an organization conducts business, including human resources, planning, budgeting, directing, organizational controls, and processes governing responsibility and authority. Management processes are referred to as the quality management system. A model for this system is the Malcolm Baldrige National Quality Award model. Models are discussed in Skill Category 3.

Work Processes

These processes include the standards and procedures that govern the performance of a specific work activity or application, such as systems development, contracting, acquisition of software, and change management.

Check Processes

These controls assure that the work processes are performed in accordance with the product standards and the customer needs. They also assure that management processes are performed according to organizational standards and needs. Examples include document reviews, program reviews, and testing.

In the context of the PDCA cycle (see Skill Category 1), management processes perform the Plan and Act components; work processes represent the Do component; and the Check processes represent the Check component. This means that management must resolve noncompliance to processes, not punish the people assigned to the work processes. Responsibility for resolving a noncompliance may be enforced automatically through controls.

As processes mature, so do the methods and approaches for managing them. While the Software Engineering Institute (SEI) Capability Maturity Model (see Skill Category 3) is directed at the software development process, the same concepts apply to management processes. Management processes at SEI Level 1 are very unpredictable and have great variability, while those at Level 5 are highly predictable, with minimal variability. For example:

- With management processes at Level 1, organizational politics become extremely important in the management style. Politics can influence unpredictable processes. In contrast, a mature, stable management process is less influenced by organizational politics, and trust increases.
- When organizations have immature management and work processes, management does not know what workers are doing, and workers may not know what management wants, or how management will react to work situations. Immature management processes are usually managed through controls such as budget and schedule, rather than relying on work processes.
- Management processes should be stabilized and matured in conjunction with the work processes. As management and work processes mature, predictability and consistency enter the workplace. The need for status reports and status meetings decreases, and there is more reliance on management by fact. The organization tends to flatten as layers of middle management are eliminated.
- Check processes are typically associated with work processes, but they exist in management processes as well. The maturing of the Check processes help mature the management and work processes. Check processes are a major source of quantitative data for the tactical dashboards that are used by management (see Skill Category 8). They are also the source of data needed for continuous process improvement. As the Check and Do processes mature, workers need less supervision, leaving management free to perform their planning responsibilities rather than acting as inspectors for the work products and services. A study of quality models shows that the major factor in maturing the work processes is the addition of check processes.

The Process Maturity Continuum

Work processes, check processes, and customer involvement, are interrelated as shown in Figure 62. The type of product determines the work and Check processes, and dictates the level of customer involvement.

Product and Services Continuum

The IT industry deals with a continuum of products, ranging from similar to professional. For example, many repetitive tasks are performed by computer operations to produce similar products such as invoices and checks. In the middle of the product continuum are job shops that produce one-of-a-kind products with the same characteristics but are customized and unique, such as a software system. At the high end of the continuum the product may also be a service in the form of professional advice or consulting.

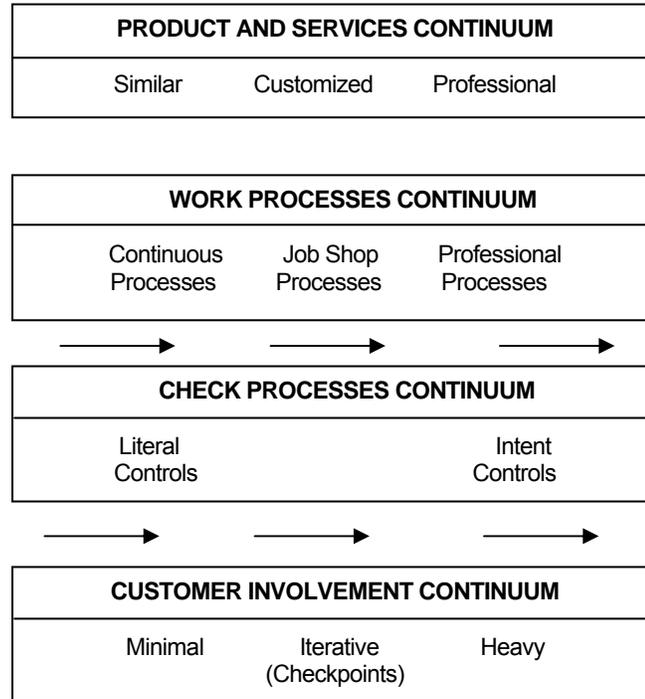


Figure 62. Process Management Continuum

As the type of product changes on the continuum, so do the work processes. The primary change in work processes is the amount of worker skill and personal contribution to the product or service produced.

Work Process Continuum

Continuous work processes are process-dependent. The worker knows the standards that the product or service must meet, and is provided with the procedures and tools needed to complete the job on time and within budget. It is important for the worker to follow the processes precisely so that each product produced is similar to the previous product produced. While individuals use experience and personal skills, the probability of success is significantly increased at this end of the continuum because previous use of the process has proven it to work.

Professional work processes are people-dependent, and may be referred to as crafts or art. They depend as much on the skills of the worker as on the steps of the process. For example, a software developer might have C++ programming skills. Or, in creating a design, the process focuses on the way the design is documented and the constraints rather than the actual design. The process assumes that the designer has certain skills and creativity and utilizes them to create the design.

With professional work processes, management depends upon the creativity and motivation of people, and must try to hire the best people and then encourage them to perform the job tasks needed. This is sometimes called "inspirational management." The inspiration can be positive, with the promise of a reward; or negative, threatening to withhold rewards if the job task is not completed on time, within budget, and to the customer's satisfaction. There are normally few

standards at this level, and the worker's manager or customer, judge the quality of the completed product.

Check Processes Continuum

The continuum of check processes parallels that of work process methodologies, and represents the types of controls associated with the work processes. Continuous work processes use literal controls. Controls of professional processes focus more on intent, and require a group of peers to assess their effectiveness. In the system design example above, the objective of placing controls on a worker's skills and creativity is to enhance the probability that the skills and creativity will be used effectively. Intent controls used in design processes would focus on such things as whether the design:

- Uses the hardware configuration effectively
- Can be implemented given the skills of the implementers
- Can be easily maintained

Customer Involvement Continuum

The customer involvement continuum shows the level of involvement needed by the customer for the type of product produced. For similar products produced by continuous work processes using literal controls, there is minimal or no customer involvement. For example, in daily computer operations, the customer would rarely be involved unless problems occurred. As an IT function moves towards customized products and services, the customer usually becomes more involved on an iterative or checkpoint basis.

At certain points during the work process, the user becomes still more involved to verify that the products being produced are those wanted by the customer. When the products become professional, the customer is heavily involved in the work or check processes. In consulting, the work process involves how the customer uses that advice. If the advice is ignored, or used incorrectly, the end product will likely be ineffective. Thus, heavy involvement is not only needed, but the involvement actually shapes the final product or service.

How Processes Are Managed

The infrastructure in a quality management environment supports process management, and is shown in Figure 9 in Skill Category 2. Process management is primarily a line (not a staff) responsibility. All levels of the organization should be involved in both establishing and using processes in their daily work. The most effective means for managing and building work processes is to have managerial responsibilities fall to a process management committee, and give teams responsibility for the activities of building and improving processes.

Other approaches have been used, but not as successfully. Generic purchased processes are not customized for the culture in which they are installed. Either they fail completely, or are only partially followed. Some organizations engage a single individual or small group to write processes for the entire organization. This frequently fails because the users do not feel ownership

of the process, or they feel that they know better ways to do work than those defined by someone else.

As a supporting staff function, the quality function also has some process management responsibilities. Their primary role should be to help and support the line organization, but not to manage the processes. As discussed in Skill Category 4, the involvement of the quality function varies with the maturity level of the quality management system.

The quality function may:

- Participate on committees, providing process management expertise.
- Provide team support, such as training, coaching, and facilitation.
- Serve as a centralized resource for measurement analysis and reporting.
- Play a "custodial" role for processes - formatting, editing, and publishing and distributing process definitions; controlling access or change to process definitions, etc.
- Occasional failure is the price of improvement.
- Audit for process deployment and compliance, when an organization is not able to separate the auditing and quality functions (separate functions are recommended).

Occasional failure is the price of improvement.

Process Template

A process template is a pictorial representation of what is needed to comply with the process requirements. For example, an order entry check might have a computer screen of the fields needed to complete the customer order. The computer screen represents the template used to accomplish the order entry process.

Process Management Processes

Process management is a PDCA cycle. Process management processes provide the framework from within which an organization can implement process management on a daily basis. Figure 63 shows how this set of practices can be viewed as a continuous improvement cycle.

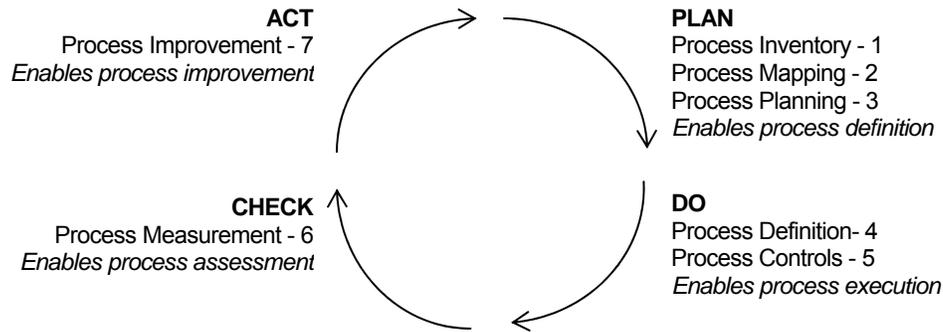


Figure 63. Process for Process Management

The process management PDCA cycle includes seven processes, together with the infrastructure group that uses that process. These processes are summarized and then discussed in more detail below.

Plan Cycle

The Plan cycle is used by the Process Management Committee and includes these processes:

1. Process Inventory defines a list of processes that support an organization in accomplishing its goals.
2. Process Mapping identifies relationships between processes and the organization's mission/goals, its functions (people), and its deliverables (products and services).
3. Process Planning sets priorities for process management projects (defining or improving processes).

Do Cycle

The Do cycle is used by the Process Development Team and includes these processes:

4. Process Definition defines a process' policies, standards, task procedures, deliverables, people and skill requirements, and tools.
5. Process Controls identifies the level and types of quality controls needed within a process, and incorporates QC procedures into the process.

Check Cycle

The Check cycle is used by the Process Management Committee and includes this process:

6. Process Measurement determines what measures and metrics are needed to strategically and tactically manage by fact, and incorporates tactical measurement into the appropriate processes.

Act Cycle

The Act cycle is used by the Process Development Team and includes this process:

7. Process Improvement uses facts (measurement results) to identify root causes of problems and to change the processes in order to improve results, prevent problems, and reduce variation.

The seven process management processes should be used in the sequence in which they are described. Planning should occur before processes are defined to ensure that the most critical processes are defined first. Implemented processes should then be measured to determine first whether they are repeatable (approximately the same product set is produced each time the process is used); and second, to determine where the process could be improved. This enables the process improvement process to focus on those process components that will provide the greatest benefit to the organization when improved.

The process management PDCA cycle is continuously performed. The Check and Act components cause new planning to occur, as will the introduction of different technology and approaches, such as client/server or the Internet. The plan then redefines the sequence in which processes should be defined, checked, and acted upon; and the cycle continues.

Planning Processes

Figure 63 showed the following three processes within the Plan cycle of the process management process.

Process Inventory

A process inventory is a major process management deliverable containing the "master list" of processes that support an organization in accomplishing its goals. Before producing an inventory, the scope of the effort must be defined, focusing on processes owned and used by the organization. The inventory is developed as part of an overall process management framework, but is also updated and improved on an ongoing basis.

Inventories can be developed by:

- Referencing existing policies, standards, procedures, and system development life cycle manuals
- Conducting brainstorming and affinity grouping sessions (see Skill Category 4)
- Surveying and interviewing employees

- Starting with existing process inventories (such as other companies' inventories, or Information Systems Process Architecture) and updating to reflect organizational structure and terminology

The inventory should list processes that produce a major outcome or deliverable. Each process listed should contain a brief description and its status or state (e.g., the CMM levels of Undefined, Initial, Repeatable, Defined, Managed, Optimized can be used). Optionally, a high-level “class” could be included to categorize processes, such as “run jobs” or “manage facilities”. Sample processes include:

- Develop Strategic Plan
- Purchase Tools
- Perform Internal Assessment
- Conduct Market Research
- Identify Product Requirements

Process Mapping

Process mapping identifies or "maps" relationships between processes and the organization's mission and goals, its functional units or roles (people), and its deliverables (products and services). The three main objectives of process mapping are to understand how a process contributes to meeting the organization's mission and goals, who is responsible for the process, and how the process interfaces to produce the organization's outcomes.

To map processes, executive management (Quality Council) must have identified the organization's mission, long and short-term goals, organizational structure, and deliverables. If formal strategic planning is not regularly performed, identifying an organization's mission and goals is difficult.

Processes should be mapped to mission and goals, to functional units (people), and to deliverables in separate matrices. The following generic process can be used for each mapping:

1. Create a matrix.
2. List processes across the top of the matrix.
3. On the left side of the matrix, list the goals, functional units or roles or deliverables.
4. Identify the linkages between the rows and columns, as follows:
 - A process may support multiple goals. Put an X in the intersection to acknowledge a linkage only if the stability or quality of the process could influence meeting goals. If all processes contribute to all goals, decompose the mission and goals further. The resulting Manage-by-Process matrix is a process map.

- Processes have primary owners, suppliers and customers. Identify a linkage in this matrix by using “O”, “S” or “C” in the intersection to distinguish between the roles.
 - Deliverables can be internal such as design specifications, or external such as user manuals. In this matrix indicate the usage of the deliverable by placing a “C”, “R”, “U” and/or “D” in the intersection. “C” is used when the deliverable is created, or the service is provided through the process. “R” indicates the deliverable is referenced or used as input to the process. “U” indicates the deliverable is updated or revised in the process. “D” means the deliverable is deleted, or retired, etc.
5. For mission and goal mapping, look for gaps where goals are not supported by processes. Consider removing any processes that do not support goals, as they do not add value. For functional unit mapping, look for gaps where units do not have processes. For deliverable mapping, look for gaps where deliverables do not have processes or vice versa.
 6. If the mapping identified any new processes, add them to the inventory.

Process Planning

Process planning allows priorities to be set for process management projects (defining or improving processes). Priorities are set based on the relative importance of the process to accomplishing the organization's mission and goals, organizational constraints or readiness, and an assessment of the project's status as follows:

- Assessing mission alignment

The degree to which each process contributes to the organization's mission and helps to accomplish organizational goals should be ranked. This involves weighting goals by importance and then applying the weighting scale to the one to three processes that most strongly align to each goal.
- Assessing organizational capability or readiness

The degree to which an organization is capable of defining or improving each process should be assessed. A score of 1-3 represents the readiness of the organization. Readiness is influenced by three main factors:

 - Motivation: Are most of the process owners committed to managing by process and motivated to define or improve it?
 - Skills: Is the process understood, and is there subject matter expertise in the methods and tools used to define or improve the process?
 - Resources: Are appropriate and adequate resources (people, time and money) allocated to define or improve the process?
- Status of each process

This was determined as part of the inventory process.

After assessing the alignment, readiness, and process status, the process management projects can be prioritized. The alignment and readiness assessments are represented by a numerical value. Process status values can be converted to numbers by setting Undefined/Initial to 3, Repeatable or Defined to 2, and Managed or Optimized to 1. Total the assessment values to provide an overall score, and assign priorities based on the scores. Each company should develop a prioritization scheme that fits their own needs. A simple scheme is to assign the top-scoring process a Priority 1, and so on.

After establishing priorities, develop the tactical plan by assigning resources and time frames to the highest-priority projects. While each process definition or improvement team should develop its own work plan, the tactical plan is a higher-level plan that the Quality Council and process management committee use to establish and track multiple process management projects. This plan should contain a project title, the resources (manpower and materials) and time frame (start and stop dates) for the project.

Many organizations use a "time box" approach to process management, which specifies a preset period of time, such as six weeks, for the project's duration. The scope of the work effort and the specific tactical plan is then driven by this time frame. The time box approach helps organizations use an incremental, iterative approach to process definition.

Do Processes

The two processes within the Do cycle of the process management process as shown in Figure 63 are discussed below.

Process Definition

The process that a team uses to scope a single process by defining its policies, standards, procedures, deliverables, people or skill requirements, and tools is called *Process Definition*. The core activity of Process Definition is defining the process; but other activities include performing walkthroughs of the process before publication, piloting the process, marketing the process, etc. Only the core activity is discussed within the scope of this guide.

The core team should contain 3-5 members (typically process owners). When multiple people or units exist for the same role, a representative group should be used with the others acting as reviewers. Team members should include a process owner, supplier, customer, process administrator, manager of the process owner, and a process auditor. Roles such as team leader, facilitator, and scribe should be assigned and the team should be trained in consensus building (see Skill Category 4).

During Process Definition the following activities occur:

Define the Scope of the Process

Use the Process Inventory, Process Maps, and existing standards and procedures to clarify the scope of the process. This involves developing a high-level process flow (major workbenches

and interfaces with other processes), major inputs and outputs (deliverables), and major customers and their requirements.

Develop the Workflow

Brainstorm the tasks and deliverables in the process and then group the tasks. Select the current best practices, adding missing tasks or deliverables and correcting flaws in the workflow. Define the workbenches internal to the process, their sequence, and major interim deliverables. Typically processes contain 3-5 workbenches, with each workbench containing 3-7 process steps or task procedures.

Develop Policies

A policy states why the workbench or deliverable exists, and indicates desired results (desirable attributes of process performance or desirable product quality characteristics). Policies should link to the organization's strategic goals, and support customer needs or requirements. A policy should be realistic, stating a desire that the organization is currently capable of accomplishing.

- **Sample Workbench Policy Statement**

A JAD session is conducted to uncover the majority of customer requirements early and efficiently, and to ensure that all involved parties interpret these requirements consistently.

- **Sample Deliverable Policy Statement**

The requirements specification must reflect the true needs of the ABC organization, and be complete, correct, testable, and easily maintainable so that it can be used throughout application systems development and maintenance.

A process management committee or the process manager usually develops a policy before establishing a process development team. When the team begins scoping and defining the detailed process, they may need to challenge the feasibility and appropriateness of the policy. If the process development team develops the policy, process management committee and/or process manager should review it.

Develop Standards

A standard states what must happen to meet the intent of the policy. Standards are more specific than policies in that they convert intentions into specific rules. Workbench standards deal with performance issues related to time frames, prerequisites, or sequencing of tasks while deliverable standards typically specify content.

A standard must be measurable, attainable, and necessary. It is measurable if it can be verified that the standard has or has not been met. A standard is attainable, if given current resources and time frame; the standard can reasonably be complied with every time. The standard is necessary if it is considered important or needed (not a "nice to have") in order to meet the intent of the policy.

- **Sample Workbench Standard**
Requirements uncovered in each JAD session must be formalized, reviewed, and approved by the JAD participants the following morning.
- **Sample Deliverable Standard**
Each unit of data or information referenced in the requirements specification must be described in the data dictionary.

Process development teams should consider the following guidelines when developing standards:

- Standards contain the basis on which compliance is determined, not the methods by which compliance is achieved.
- Each time a workbench is executed or a deliverable is produced, quality control procedures must be able to evaluate whether the standard has been met.
- It is easier to control quality with binary standards that require no subjective assessments or judgments to determine compliance; however, this is not always possible. Different types of standards may result in different QC methods. For example:
 - **Literal or binary:** Each dataflow line on a dataflow diagram must contain a dataflow descriptor of five words or less that identifies the data being passed between processes and stores.
 - **Judgment or intent:** Each dataflow line on a dataflow diagram must contain a dataflow descriptor that concisely and accurately identifies the data being passed between processes and stores.
- Since customer requirements often determine standards, interview customers to find out how they use the deliverables, any problems they have, and the most important quality characteristics.
- Consider "internal requirements" - the process owner's desires for effective, efficient, and consistent processes.
- If information from other processes is a problem, create standards that serve as entrance criteria. It is better to embed these standards in the supplier's process.
- Policy statements may need to be re-evaluated in light of standards development.

Develop Procedures

Procedures describe how work is done, and indicate the "best current way" to meet standards. Ideally, if the procedures are followed, the standards will automatically be followed and the intent of the policy will be met.

A task is a single step in a procedure. Procedures may have many tasks. Task procedures should refer to people, accessible tools, known techniques or methods, and templates for

deliverables. If appropriate, tasks can also refer to more detailed work instructions, such as training or user's manuals.

Procedures are often written in playscript format (see Skill Category 4). Like a script in a play, each line (task) is identified in its proper sequence, and the actor (role or function) responsible for "saying" each line is noted. A taskflow diagram or flowchart may be used to graphically depict the procedure.

A process development team develops procedures using the following guidelines:

- Procedures are not always required and unless critical, should be minimized.
- Skill set and knowledge (subject matter expertise) requirements or prerequisites for performing the procedure should be identified. The procedures should not be a substitute for skill set development or describe how to do something the person doing the task should know (given the prerequisites).

There are Do procedures and Check procedures. Check procedures are described in the next section on process control. A Do procedure explains in a step-by-step format the tasks needed to produce a product.

- Sample Do procedure for the requirements definition process
 - 1) Scribe: Use the XYZ tool to enter the requirements. Generate a list of any open issues using the XX template.
 - 2) Leader: Walk through the requirements, paraphrasing each item. Address each open issue when its reference column contains the item being covered.
- Sample Do procedure to write a computer program might have the tasks
 - 1) Project manager: get a job number,
 - 2) Programmer: obtain the program specifications, and so forth.

Check Processes

The Check process incorporates the process controls needed to be assured the "Do process" was performed correctly. Process control identifies the appropriate types of quality controls needed within a process, and designs and incorporates them as Check procedures into the process. Check procedures describe how to evaluate whether the right work was done (output meets needs) and whether the work was done right (output meets standards). Controls should be designed based on the criticality of the process and what needs to be checked.

Check procedures are defined in the same way as Do procedures. If the playscript format is used, Check procedures should be incorporated to the taskflow diagram or flowchart at the appropriate spot. For example, a quality control checklist associated with programming would have questions such as, "Was a flowchart produced that describes the processing of the program?" or "How much quality control is enough?"

One of the quality management philosophy objectives is to continually improve process capability by reducing variation and rework. With this strategy quality is built into the products rather than

tested in. As standards and Do procedures are perfected, the need for extensive quality control is reduced.

Quality control procedures are considered appraisal costs. They add to the overall cost, increase the effort required, and increase the cycle time for building products and delivering services. Where standards and Do procedures are not perfected, quality control is necessary in a process to catch defects before they affect the outcome of downstream processes. Appraisal costs are part of the Cost of Quality, which is covered in Skill Category 1.

The challenge is to install the appropriate amount and types of controls to minimize cost, effort, and cycle time; and to minimize the risk that defects will go undetected and "leak" into other processes.

Identify Control Points

Controls are often placed near the end of a process or workbench, but this is not always the most appropriate location. The first step in process control design is to identify the most logical points in the process to add controls. One way to address this issue is to identify and rank process risks. Process risks are those things that could occur during execution of a process that would result in defects and rework. For example, process risks for an "estimate project effort" process may be:

- Use of inaccurate historical data
- Misuse of historical data
- Inaccurate estimation algorithm or mathematical mistakes
- Inadequate contingencies used
- Wrong staffing ratios and loading figures used

The team reviews the process workflow, policies, standards, and procedures, and then brainstorms risks. Risks that are found to be outside the scope and control of the process being defined are best controlled in other processes. If there is any risk that standards may not be followed, that should also be noted.

Ranking risks should be based on two factors: the probability that the risk might occur and the impact (cost of defect and rework) if the risk does occur. A high, medium, low ranking scheme could be used. While this is somewhat subjective without historical defect and cost data, the judgment and knowledge of the process owners on the team is usually accurate.

Plotting where the top-ranking risks lie can help to determine the most appropriate point to insert controls. Using the ranked list of risks, the team should identify where defects potentially originate. In general, the closer the control is to the point of origin of the defect, the better. The severity of the risk will also influence the selection of the control methods. Skill Category 8 contains additional information on risk.

The control point decisions should be adjusted if needed. Control methods should be considered based on the following:

- Risk severity
- Cost, effort, and cycle time impact
- Availability of appropriate resources and people
- Strength of the control method
- Impact on the overall culture

Figure 64 shows five main categories of control methods that are discussed below.

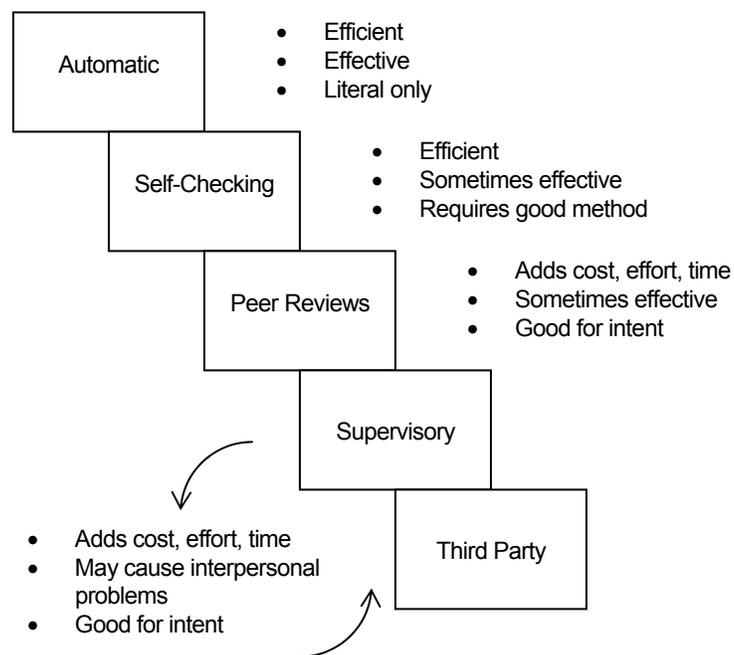


Figure 64. Control Methods Categories

Automatic

When performing a Do procedure, automation is the only way to force compliance. Some tools, such as CASE tools, automatically enforce task completion and sequence, and deliverable standards.

Self-Checking

This is when the process owner (author) uses a method other than the Do procedure, to crosscheck his/her work. Methods in this category are:

- Analysis tools, which parse and analyze deliverables after they have been created, such as spelling checkers, writing analyzers, and code analyzers (e.g., standards compliance, complexity analyzers, and cross-reference tools).
- Checklists, which are worksheets containing a list of questions oriented towards determining whether the standards have been adhered to. They are designed to provide a summary-style self-check for the author. Checklists often mirror policies, standards, and procedures, but address each compliance issue in the form of a question. A "yes" answer means that the author has followed the process and has produced a result that matches the intent of the policy statement. A "no" answer indicates noncompliance.
- Desk-checks, where the author or owner reviews the product against specifications and standards. It is uncontrolled and subject to the time requirements of each individual.
- One-on-one reviews, which are informal reviews conducted between the author/owner and one other person. The objective is to review against specifications and standards.
- Tests, which validate that the actual results are the expected or desired results.

Peer Reviews

One or more process owners review the results of the author. Typically, quality problems are noted, and the author corrects them. Various methods exist, from informal to formal. Methods such as informal walkthroughs, formal inspections, and checkpoint reviews are discussed in Skill Category 7.

Supervisory

The author's supervisor reviews the work and ensures that defects found are corrected. This is problematic because it takes responsibility for quality away from the worker, and may be ineffective because the supervisor is not sufficiently skilled or knowledgeable about the work to make intent or judgment calls regarding compliance. It may also influence the supervisor unfairly regarding a worker's performance. Supervisors may use the informal walkthrough, checklists, or testing for control methods.

Third Party

An independent group evaluates the product. As with supervisory controls, this is problematic because responsibility for quality is taken from the worker and the third party may not have the skills or knowledge about the work to determine compliance. Examples of independent groups are quality functions and independent test teams. Methods used by third parties include informal walkthroughs, checklists, testing, analysis tools, and sampling.

Process Measurement

Process measurement determines what strategic and tactical measures and metrics are needed to manage by fact, and incorporates measurement into the appropriate processes. Measurement provides quantitative feedback to an organization about whether it is achieving its goals - whether it is moving towards its results.

The program starts by building a measurement base, and then identifies goals for the desired business results. To implement measurement in a process, it identifies the relationship of process contributors and results, and how to measure each. The fourth phase discusses measurement by fact.

Results should play a significant role in process definition. Process requirements can be derived from analyzing the factors that contribute to achieving desired results. These factors include identifying:

- The desirable attributes that must be present when processes are performed
- The desirable characteristics that must be included when products are produced in order to achieve desired results

Next, consider which processes should incorporate the requirements to address the contributors. These requirements become policies and standards, and form the basis for tactical measurements. They are needed to ensure that processes are being followed and are also facts that must be analyzed when measuring and interpreting results.

Measurements may already be used to control the process. For example, if "maintainable code" is a desirable outcome for the "develop unit" process, then a standard may have been developed limiting code complexity and size. A self-check QC method to analyze and report code complexity and size may have been incorporated into the process. While the measurement has been selected, how it will be collected in order to evaluate the process' effectiveness over time may not have been considered.

One measure of process effectiveness is compliance. Other process measurements must be derived from the policies, standards, and procedures themselves.

If the process being defined will be a measurement collection, analysis, or reporting point, the tasks that describe how to do these things must be defined. Measurement data should be developed as part of executing processes and collected for use by the line management responsible for those processes. Line management uses the data to control the process and the quality of the products produced by the process. Normally, the QA analyst is a secondary recipient of IT data, and uses it to improve the process itself.

Measurement procedures are defined as are Do and Check procedures. If using the playscript format, measurement procedures should be incorporated at the appropriate spot, and the procedures added to the taskflow diagram or flowchart if those forms are used.

Testing

Testing is the process that determines whether or not the actual results of testing equal the expected results of processing. The concept and practices used to test are covered in Skill Category 7.

Act Processes

The Act cycle of the process management process includes the one process of process improvement, as shown in Figure 65. The purpose of process improvement is to reduce the frequency of defects, including process ineffectiveness. Figure 65 shows how, without process improvement, there is a continuous cycle of uncovering product defects and removing them from the product. This is because the same defects will likely occur every time that product is built. Process improvement uses facts (measurement results) to identify the root causes of problems and to change processes so that results will improve, problems will be prevented, and variation will be reduced.

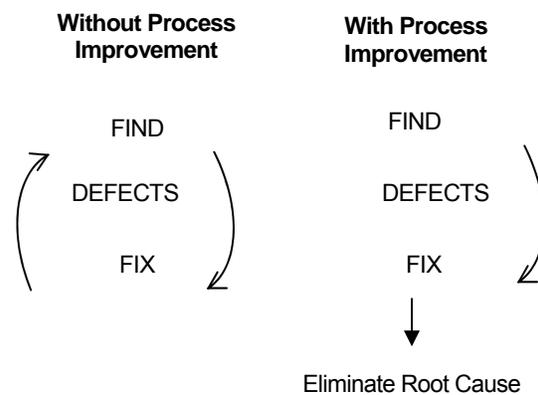


Figure 65. Concept of Process Improvement

The long-range objective for process improvement is to eliminate the need for quality control activities such as testing, reviews, and inspections. If the processes do not generate defects, then there is no need to search out, find, and correct product defects. For example, if an organization identifies data entry errors as a high-defect activity, then an improvement program can be started to drive down those defects. The objective of the program would be to change the products or processes in order to remove the cause of the data entry defects.

Process improvement is a continuous, iterative process. It involves finding the defects, accumulating the defects in a manner that identifies the significant from the insignificant, selecting a single defect, and identifying the root cause of that defect. At that point, an action program is put into place to reduce the frequency of defects or eliminate the root cause of the defect. Then the process selects the next most significant defect and repeats the improvement process.

Process improvement has two components:

- Establishing process improvement teams (PIT), which may or may not include members of the process development team
- Providing the teams with a process to use for process improvement

Process Improvement Teams

Process improvement must be accomplished by emphasizing teamwork. The PIT component addresses this need. Every member of the organization should become involved in the process improvement process. Not everyone must be on a team, although as many teams as practical should be organized. The two methods below are recommended for creating the teams.

- **Natural Work Group**

Natural work groups, such as a system development project team, characterize the way companies currently organize their employees. Using natural work groups for a PIT is the easiest, quickest, and least confusing method. Generally, these teams already exist, and the members have established working relationships. They are also usually aware of improvements that could be made to improve the effectiveness of their work group.

If natural work group teams elect to address problems that impact beyond their team, the improvement process should support the collaboration of multiple teams working on problems with a broad scope. The measurement system would be required to track and report these instances, so appropriate reward and recognition would occur for these shared team efforts.

- **Interdepartmental Teams**

This method of organizing teams across departmental boundaries promotes interdepartmental teamwork and contributes to breaking barriers (and problems) that exist within organizations. It is also an excellent way to address problems that affect more than one work group or department. Because these types of problems are typically more complex, it is only recommended as a method of organizing when the PITs are mature.

Regardless of the method, each team should have between five and eight members. Smaller or larger groups can lose effectiveness. Team member responsibilities should include:

- Identifying problems and selecting the ones on which to work
- Proposing solutions to problems
- Choosing an appropriate solution and improvement approach
- Implementing the chosen improvement
- Documenting required data regarding team activities in the PIT measurement system
- Ensuring consistent use of a common set of statistical process control (SPC) tools and techniques
- Presenting the implemented improvement to a quality improvement administrator for certification that the PIT processes were followed

Team members should allocate 30-45 minutes per week. Some might meet weekly and others might meet 60-90 minutes every other week. If everyone participates, it is important to limit the time so there is not a major impact on member's daily responsibilities. If teams are trained in

effective problem-solving and meeting-leading skills, this small amount of meeting time can be very productive. Process improvement and associated cost savings will soar.

Utilizing 30-45 minutes per week, the Paul Revere Insurance Company was able to save \$3.25 million in annualized savings in their first year, and \$7.5 million in their second (250 teams, 1,200 employees). United Data Services (the IT function for United Telephone System) saved \$4.75 million annualized in the first year (60 teams); and McCormack and Dodge saved over \$2 million in the first four months (150 teams). These are impressive figures, but achievable with minimal time commitments. (NOTE: The time commitment includes both meeting time, plus individual time spent on PIT activities, such as planning, designing, and implementing improvements.)

Process Improvement Process

For organizations that do not have a process improvement process (sometimes called quality or continuous improvement), the eight-step process below is recommended. The first three steps focus on process identification and understanding while steps 4 through 8 focus on the improvement aspect.

1. Select process and team
2. Describe current process
3. Assess process for control and capability
4. Brainstorm for improvement
5. Plan how to test proposed improvement
6. Analyze results
7. Compare results
8. Change process or redo steps 4-8

Identify and Understand the Process

1. Select Process and Team

The process to be improved may be selected by the improvement team or assigned by management. Leaders of improvement teams are usually process owners, and members are process users and may be cross-functional. Organizational improvement team members are usually volunteers; however, if subject experts are required and none volunteer, management will assign them. After the process has been selected and the improvement team formed, customers of, and suppliers to, the process are determined. If not already on the team, customer and supplier representatives should be added when practical. Often the improvement team begins by analyzing data on customer complaints, defects, rework, and cost of quality. The quantitative tools used in this step can include the process flow, Pareto charts, and run charts.

2. Describe the Process

Two simultaneous actions are initiated in this step: defining customer-supplier relationships and determining actual process flow. The customer's requirements are defined using operational definitions to assure complete understanding between those providing the product or service and the customer. The customer's quality characteristics are defined and the current state of satisfying these, determined. In most instances, the customer referred to is the internal customer. This same idea is applied to the suppliers in the process. The process owner's expectations are defined for suppliers of inputs to the process. The customer and supplier must then agree on the specifications to be met and how quality will be measured.

While the customer-supplier relationships are being defined, the team is also building a flowchart of the current process (how it is done at this point in time) if one was not completed in Step 1. Questions the team asks include: Does a person do the process the same every time? Does each person, if more than one does the process, do it the same? Is there variation between projects? The flowchart is used to identify "work around" sources of variation, and recycle, rework, and other causes of poor quality or productivity. The flowchart and discussions with the customer determine process measurement points, which monitor process health. Both input and output of the process are monitored. Data that needs to be collected and analyzed can be identified in this step. A discussion of the ideal way to do the process is initiated.

A brainstorming session should be used to develop a cause-and-effect diagram identifying all possible causes of process input variation. Next categorize the causes. The customer's required quality characteristics are the effects. Separate cause-and-effect diagrams are constructed for each effect. Scatter diagrams are then used to establish the relationship (correlation) between each of the major causes of variation and the process output.

Tools used in this step include measurement and data collection methods, flowcharts, Pareto charts, cause-and-effect diagrams, and scatter diagrams.

3. Assess the Process

To assure that decisions are being made using precise and accurate data, the measurement system used to assess the process must be evaluated. If the measurement is counting, care should be taken to assure that everyone is counting the same things in the same manner. Once it has been determined that the measurement system accurately describes the data of interest, the input and output should be measured and baselines established.

Examples of process output indicators are listed below:

- Amount of rework
- Yield or productivity
- Errors (defects) in products, reports, or services

- Cycle time
- Timeliness
- Number of schedules missed
- Engineering changes per document
- Downtime because of maintenance, parts shortage, or other factors
- Overtime
- Absenteeism or sick leave

Process inputs, determined in Step 2, represent possible sources of variation that must be quantified. Process inputs and outputs can be baselined using Pareto charts, run charts, histograms, scatter diagrams, and control charts. The cause-and-effect diagram developed in the previous step should be reviewed and updated.

Next, the process is assessed for statistical control by constructing a control chart(s) for each important process input. If points are found outside the control limits, these are due to special causes of variation and must be investigated (see Skill Category 4). Control mechanisms may need to be installed in higher-level processes to eliminate them in the future. More data is then gathered to confirm the elimination of the special causes. The improvement team may have to proceed to Steps 4 through 8 to find and eliminate the special causes of variation. Once special causes of variation have been eliminated, the process is stable and, therefore, in a state of statistical control.

Another assessment of the process determines whether it is capable of meeting the customer's expectations (refer also to Skill Category 8). The control limits of the process are compared to the customer's specification limits. If the control limits are inside the specification limits, the process is capable of satisfying the customer. If the control limits are outside the specification limits, the process is not capable and common cause variation must be reduced or the process retargeted, or both. Both process location and variation may be a problem. This change in process location and reduction in variation is accomplished by using Steps 4 through 8 of the continuous improvement strategy.

The tools available for this step are data collection methods, Pareto charts, run charts, control charts, process capability, measurement capability, and advanced statistical techniques.

Improve the Process

4. Brainstorm for Improvement

The team should review all relevant information, such as the process flowchart, cause-and-effect diagrams, and control charts. A brainstorming session may be held to generate ideas for reducing input variation. It may help to visualize the ideal process. Improvement ideas must be prioritized and a theory for improvement developed. The what, how, and why of the improvement should be documented so everyone agrees to

the plan. All statistical tools should be used in this step, although brainstorming is used the most.

5. Plan How to Test Proposed Improvement(s)

In this step a plan for testing the proposed improvement developed in the preceding step is created. This process improvement plan specifies what data is to be collected, how the data is to be collected, who will collect the data, and how the data will be analyzed after collection. The specific statistical tools to be used for analysis are defined. Attention is given to designing data collection forms that are simple, concise, and easy to understand. The forms should be tested before being put into use. The process improvement plan is a formal document(s) that is approved by the improvement team.

6. Analyze Results

The plan developed in the prior step is implemented. The required data is collected using the previously tested forms, and then analyzed using the statistical tools noted in the process improvement plan. The plan is reviewed at improvement team meetings to track progress on process improvement. The tools usually used in this step are data collection methods, flowcharts, run charts, scatter diagrams, histograms, measurement capability, control charts, process capability, and advanced statistical techniques.

7. Compare Results

This step uses the same statistical tools as in Step 3 to compare the test results with that predicted in Step 4. Has the process been improved? Do the test results agree with scientific or engineering theory? If the results are not as expected, can they be explained? When results are not as expected, it is not a failure - something is still learned about the process and its variation. The new information will be used to develop a new theory. Care is taken to document lessons learned and accomplishments. A set of before and after Pareto charts, histograms, run charts, or control charts are generally used in this step.

8. Change Process or Redo Steps 4-8

If the results are not as expected, then based on what was learned, a new method for improvement must be developed by reverting back to Step 4. If the results are as expected, and are practical and cost-effective, the change recommended in the process improvement plan is implemented. The implementation should be monitored to assure that the gains made in reducing variation are maintained. The team then returns to Step 2 to update the process documentation.

A decision is now required concerning the next improvement effort. Should the team continue to improve this process by further reducing variation due to common causes or start working on another process that is not meeting customer requirements? The answer will probably be based on an economic analysis. If it is economically desirable to continue to reduce the process variation, the team develops another improvement

method and repeats Steps 4 through 8. If the decision is made to improve a different process, the process begins again at Step 1 by defining a team.

Quality Control Practices

Quality control practices should occur during product development, product acquisition, product construction at the end of development/acquisition and throughout product change and operation. During development, the quality control process is frequently called verification and at the conclusion of development, it is called validation. This category will address the various types of controls and when they are best used in the process. The quality practitioner should also be familiar with verification and validation techniques, the framework for developing testing tactics, change control and configuration management.

<i>Testing Concepts</i>	309
<i>Developing Testing Methodologies</i>	320
<i>Verification and Validation Methods</i>	325
<i>Software Change Control</i>	332
<i>Defect Management</i>	333

Testing Concepts

Many testers fail to do testing effectively and efficiently because they do not know the basic concepts of testing. The purpose of this section is to describe those basic testing concepts.

The Testers' Workbench

The testers' workbench is the process used to verify and validate the system structurally and functionally. To understand the testing methodology, it is necessary to understand the workbench concept. Process workbenches are discussed in Skill Category 6.

A testers' workbench is one part of the software development life cycle, which is comprised of many workbenches. Two examples of the workbench concept are given below.

- The programmers' workbench for one of the steps to build a system is:
 - Input (program specifications) is given to the producer (programmer).
 - Work (coding and debugging) is performed; a procedure is followed, and a product or interim deliverable (a program, module, or unit) is produced.
 - Work is checked to ensure the product meets the specifications and standards, and that the procedure was followed. If the check finds no problems, the product is released to the next workbench. If the check finds a problem, the product is sent back for rework.
- A project team uses the workbench to guide them through a unit test of computer code. The programmer takes the following steps:
 - Give input products (e.g., program code) to the tester.
 - Perform work (execute unit tests), follow a procedure, and produce a product or interim deliverable (e.g., the test results).
 - Check work to ensure test results meet test specifications and standards and that the test procedure was followed. If the check finds no problems, release the product (test results) to the next workbench. If the check process finds a problem, the product is sent back for rework.

Test Stages

There are four main testing stages in a structured software development process. They are:

Unit Testing

These tests demonstrate that a single program, module, or unit of code function as designed. For example, observing the result when pressing a function key to complete an action. Tested units are ready for testing with other system components such as other software units, hardware, documentation, or users.

Integration Testing

These tests are conducted on tasks that involve more than one application or database, or on related programs, modules, or units of code, to validate that multiple parts of the system interact according to the system design. Each integrated portion of the system is then ready for testing with other parts of the system.

System Testing

These tests simulate operation of the entire system and confirm that it runs correctly. Upon completion, the validated system requirements result in a tested system based on the specification developed or purchased.

User Acceptance Testing

This real-world test is the most important to the business, and it cannot be conducted in isolation. Internal staff, customers, vendor, or other users interact with the system to ensure that it will function as desired regardless of the system requirements. The result is a tested system based on user needs.

Independent Testing

The primary responsibility of individuals accountable for testing activities is to ensure that quality is measured accurately. Often, knowing that quality is being measured is enough to cause improvements in the applications being developed. The existence of a Tester or someone in the organization devoted to test activities is a form of independence, in the loosest definition.

The roles and reporting structure of test resources differ across and within organizations. These resources may be business or systems analysts assigned to perform testing activities, or, less beneficially, they may be Testers who report to the project manager. Ideally, the test resources will have a reporting structure independent from the group designing or developing the application in order to assure that the quality of the application is given as much consideration as the project budget and timeline.

The benefits of independent testing can be seen even in the unit testing stage. Often, successful development teams will have a peer perform unit testing on a program or module. Once a portion of the application is ready for integration testing, the same benefits can be achieved by having an independent person plan and coordinate the integration testing.

Where an independent test team exists, they are usually responsible for system testing, the oversight of user acceptance testing, and providing an unbiased assessment of the quality of an application. The team may also support or participate in other phases of testing as well as executing special test types such as performance and load testing.

An independent test team is usually comprised of a Test Manager or team leader, Testers, and additional Testers. The Test Manager should join the team by the beginning of the requirements definition stage. Key Testers may also join the team at this stage on large projects to assist with test planning activities. Other testers join later to assist with the creation of test cases and scripts. Additional Testers, including users who will participate in test execution, usually join the test team right before system testing is scheduled to begin.

The Test Manager ensures that testing is performed, that it is documented, and that testing techniques are established and developed. The manager is also responsible for:

- Planning and estimating tests
- Designing the test strategy
- Ensuring tests are created and executed in a timely and productive manner
- Reviewing analysis and design artifacts
- Chairing the test readiness review

- Managing the test effort
- Overseeing acceptance tests

Testers are usually responsible for:

- Developing test cases and procedures
- Planning, capturing, and conditioning test data
- Reviewing analysis and design artifacts
- Executing tests
- Utilizing automated test tools for regression testing
- Preparing test documentation
- Tracking and reporting defects

Other Testers primarily focus on test execution, defect reporting, and regression testing. They may be junior members of the test team, users, marketing or product representatives, or others.

The test team should be represented in all key requirements and design meetings, including: JAD or requirements definition sessions, risk analysis sessions, prototype review sessions, etc. They should also participate in all inspection or walkthrough reviews for requirements and design artifacts.

Static versus Dynamic Testing

Static testing is another name for in-process reviewing. It means that the test is being performed without executing the code. Static testing occurs throughout the development life cycle; however, a large part of it takes place during the requirements and design phases in the form of walk-through inspections, and system reviews. Other examples of static testing include code analyzers or writing analyzers.

Dynamic testing (also known as program testing) implies that the code is being executed on a machine.

Verification versus Validation

Verification ensures that the system (software, hardware, documentation, and personnel) complies with an organization's standards and processes, relying on review of non-executable methods. *Validation* physically ensures that the system operates according to plan by executing the system functions through a series of tests that can be observed and evaluated. Verification answers the question, "Did we build the right system?" while validation addresses, "Did we build the system right?"

Keep in mind that verification and validation techniques can be applied to every element of the computerized system. You'll find these techniques in publications dealing with the design and implementation of user manuals and training courses, as well as in industry publications.

Computer System Verification and Validation Examples

Verification requires several types of reviews, including requirements reviews, design reviews, code walkthroughs, code inspections, and test reviews. The system user should be involved in these reviews to find defects before they are built into the system. In the case of purchased systems, user input is needed to assure that the supplier makes the appropriate tests to eliminate defects. Table 15 shows examples of verification. The list is not exhaustive, but it does show who performs the task and what the deliverables are. For purchased systems, the term “developers” applies to the supplier’s development staff.

Table 15. Computer System Verification Examples

Verification Example	Performed By	Explanation	Deliverable
Requirements Reviews	Developers, Users	The study and discussion of the computer system requirements to ensure they meet stated user needs and are feasible.	Reviewed statement of requirements Ready to be translated into system design
Design Reviews	Developers	The study and discussion of the computer system design to ensure it will support the system requirements.	System design Ready to be translated into computer programs Hardware configurations Documentation Training
Code Walkthroughs	Developers	An informal analysis of the program source code to find defects and verify coding techniques.	Computer software ready for testing or more detailed inspections by the developer.
Code Inspections	Developers	A formal analysis of the program source code to find defects as defined by meeting computer system design specifications. Usually performed by a team composed of developers and subject matter experts.	Computer software ready for testing by the developer.

Validation is accomplished simply by executing a real-life function (if you wanted to check to see if your mechanic had fixed the starter on your car, you’d try to start the car). Examples of validation are shown in Table 16. As in the table above, the list is not exhaustive.

Determining when to perform verification and validation relates to the development, acquisition, and maintenance of software. For software testing, this relationship is especially critical because:

- The corrections will probably be made using the same process for developing the software. If the software was developed internally using a waterfall methodology, that methodology will probably be followed in making the corrections; on the other hand, if the software was purchased or contracted, the supplier will likely make the correction. You’ll need to prepare tests for either eventuality.

- Testers can probably use the same test plans and test data prepared for testing the original software. If testers prepared effective test plans and created extensive test data, those plans and test data can probably be used in the testing effort, thereby reducing the time and cost of testing.

Table 16. Computer System Validation Examples

Validation Example	Performed By	Explanation	Deliverable
Unit Testing	Developers	The testing of a single program, module, or unit of code. Usually performed by the developer of the unit. Validates that the software performs as designed.	Software unit ready for testing with other system components, such as other software units, hardware, documentation, or users.
Integrated Testing	Developers	The testing of related programs, modules, or units of code. Validates that multiple parts of the system interact according to the system design.	Portions of the system ready for testing with other portions of the system.
System Testing	Developers, Users	The testing of an entire computer system. This kind of testing can include functional and structural testing, such as stress testing. Validates the system requirements.	A tested computer system, based on what was specified to be developed or purchased.
User Acceptance Testing	Users	The testing of a computer system or parts of a computer system to make sure it will work in the system regardless of what the system requirements indicate.	A tested computer system, based on user needs.

The “V” Testing Concept Example

Life cycle testing involves continuous testing of the system during the developmental process. At predetermined points, the results of the development process are inspected to determine the correctness of the implementation. These inspections identify defects at the earliest possible point.

Life cycle testing cannot occur until a formalized SDLC has been incorporated. Life cycle testing is dependent upon the completion of predetermined deliverables at specified points in the developmental life cycle. If information services personnel have the discretion to determine the order in which deliverables are developed, the life cycle test process becomes ineffective. This is due to variability in the process, which normally increases cost.

The life cycle testing concept can best be accomplished by the formation of a test team. The team is comprised of members of the project who may be both implementing and testing the system. When members of the team are testing the system, they must use a formal testing methodology to clearly distinguish the implementation mode from the test mode. They also must follow a structured methodology when approaching testing, the same as when approaching system

development. Without a specific structured test methodology, the test team concept is ineffective because team members would follow the same methodology for testing as they used for developing the system. Experience shows people are blind to their own mistakes, so the effectiveness of the test team is dependent upon developing the system under one methodology and testing it under another.

The life cycle testing concept is illustrated in Figure 66. This illustration shows that when the project starts, both the system development process and system test process begins. The team that is developing the system begins the systems development process and the team that is conducting the system test begins planning the system test process. Both teams start at the same point using the same information. The systems development team has the responsibility to define and document the requirements for developmental purposes. The test team will likewise use those same requirements, but for the purpose of testing the system. At appropriate points during the developmental process, the test team will test the developmental process in an attempt to uncover defects. The test team should use the structured testing techniques outlined in this guide as a basis of evaluating the system development process deliverables.

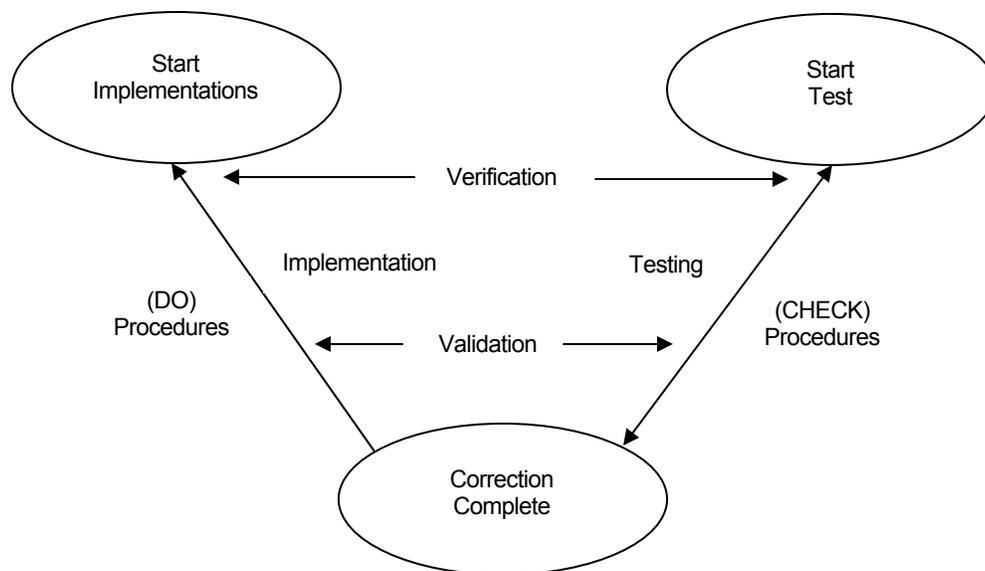


Figure 66. The “V” Concept of Software Testing

During the system test process, an appropriate set of test transactions should be developed, to be completed at the same time as the completion of the application system. When the application meets the acceptance criteria, it can be integrated into the operating environment. During this process, the systems development team and the systems test team work closely together to ensure that the application is properly integrated into the production environment. At that point, the teams again split to ensure the correctness of changes made during the maintenance phase. The maintenance team will make whatever changes and enhancements are necessary to the application

system, and the test team will continue the test process to ensure that those enhancements are properly implemented and integrated into the production environment.

In the V-testing concept, your project's "Do" and "Check" procedures slowly converge from start to finish (see Figure 66), which indicates that as the "Do" team attempts to implement a solution, the "Check" team concurrently develops a process to minimize or eliminate the risk. If the two groups work closely together, the high level of risk at a project's inception will decrease to an acceptable level by the project's conclusion.

Stress versus Volume versus Performance

Many testers use stress and volume testing to mean test the system constraints. A stricter definition is that *stress testing* tests the built-in constraints of the system, such as internal table size; and *volume testing* tests the system's ability in an operating environment to process very large amounts of data. *Performance testing* tests the systems ability to meet performance standards, such as a maximum three-second response to a user's request.

Test Objectives

A test objective (goal) is a statement of what the test team or tester is expected to accomplish during a specific testing activity. Test objectives, are usually defined during requirements analysis, and guide the development of test cases, test scripts, and test data.

Test objectives enhance communication both within and outside the project team by defining the scope of the testing effort, and enabling the test manager and project manager to gauge testing progress and success.

Each test objective should contain a statement of purpose and a high-level description of the expected results stated in measurable terms. Completion criteria for test objectives define the success measure for the tests. Test objectives can be easily derived using the system requirements documentation, the test strategy, results of the risk assessment, and the test team assignments. Test objectives are not simply a restatement of the system's requirements, but the actual way in which the system will be tested in order to assure that the system objective has been met. If requirements are lacking or poorly written, then the test team must have a defined method for uncovering and defining test objectives. Techniques to consider include brainstorming, relating test objectives to the system outputs, developing use cases or relating test objectives to events or system inputs.

The users and project team must prioritize the test objectives. Usually the highest priority is assigned to objectives related to high priority or high-risk requirements defined for the project. In cases where test time is cut short, test cases supporting the highest priority objectives would be executed first.

As a final step, the test team should perform quality control on this activity. This might entail using a checklist or worksheet to ensure that the process to set test objectives was followed, or reviewing the objectives with the system users.

Reviews and Inspections

Reviews are conducted to utilize the variety of perspectives and talents brought together in a team. The main goal is to identify defects within the stage or phase of the project where they originate, rather than in later test stages; this is referred to as “stage containment.” As reviews are generally greater than 65% efficient in finding defects, and testing is often less than 30% efficient, the advantage is obvious. In addition, since defects identified in the review process are found earlier in the life cycle, they are less expensive to correct.

Another advantage of holding reviews is not readily measurable. Reviews are an efficient method of educating a large number of people on a specific product or project in a relatively short period of time. Semiformal reviews (see Review Formats below) are especially good for this, and are often held for just that purpose. In addition to learning about a specific product or project, team members are exposed to a variety of approaches to technical issues (a cross-pollination effect). Finally, reviews provide training in, and enforce the use of, standards, as nonconformance to standards is considered a defect and reported as such.

The timing and the purpose of a review determine what type of review takes place, when it takes place, and how it is conducted. Reviews are performed during the development process, at the end of a phase, and at the end of the project.

Review Formats

There are three review formats as follows:

Informal Review

This review is generally a one-on-one meeting between the producer of a work product and a peer or co-worker, and is initiated as a request for input regarding a particular artifact or problem. There is no agenda, no preparation time, and results are not formally reported. These reviews occur on an as needed basis throughout each phase of a project.

Semiformal Review (or Walkthrough)

This review is facilitated by the producer of the material being reviewed (e.g., documentation or code). The participants are led through the material in one of two formats: the presentation is made without interruptions and comments are given at the end, or comments are made throughout. In either case, the issues raised are captured and published in a report distributed to the participants. Possible solutions for uncovered defects are typically not discussed during the review. Semiformal reviews should occur multiple times during a phase for segments or “packages” of work.

Formal Review (or Inspection)

This review is facilitated by a knowledgeable individual called a moderator, who is not the producer or a team member of the product under review. The meeting is planned in advance, and material is distributed to participants before the review so they will be familiar with the topic and arrive prepared. Full participation by all members of the review team is required;

therefore, the quality of a formal review is directly dependent on the preparation of the participants. A recorder assists the moderator by capturing issues and action items, and publishing them in a formal report with distribution to participants and management. Defects found are tracked through resolution, usually by way of the existing defect-tracking system. Formal reviews may be held at any time, and apply to both development and test products.

Regardless of the format, three rules apply to all reviews:

1. The product is reviewed, not the producer
2. Defects and issues are identified, not corrected during the session
3. All members of the review team are responsible for the results of the review

In-Process Reviews

In-Process reviews are used to examine a product during a specific time period of its life cycle, such as during the design activity. They are usually limited to a segment of a project, with the goal of identifying defects as work progresses, rather than at the close of a phase or even later, when they are more costly to correct. These reviews may use an informal, semiformal or formal review format.

Checkpoint Reviews

These are facilitated reviews held at predetermined points in the development process. The objective is to evaluate a system as it is being specified, designed, implemented, and tested. Checkpoint reviews focus on ensuring that critical success factors are being adequately addressed during system development. The participants are subject matter experts on the specific factors to be reviewed against, and could include customer representatives, analysts, programmers, vendors, auditors, etc. For example, if system performance was identified as a critical requirement, three checkpoint reviews might be set up at the end of the requirements, design, and coding phases to ensure there were no performance issues before proceeding to the next phase. Instead of walking team members through a general checklist (as would be done in a phase-end review), a designated performance expert would look specifically at whether performance requirements were being met.

Phase-End Reviews

Phase-end reviews (also called Decision-Point or Gate reviews) look at the product for the main purpose of determining whether to continue with planned activities. In contrast to the checkpoint reviews, which focus on critical success factors, phase-end reviews are more general in nature.

Phase-end reviews are held at the end of each phase, in a formal review format. Defects found are tracked through resolution, usually through a defect-tracking system. Although there may be more, the most common phase-end reviews are listed below. Project status, risks, and non-technical issues are also reviewed.

Software Requirements Review

This review is aimed at verifying and approving the documented software requirements for the purpose of establishing a baseline and identifying analysis packages. The Development Plan, Software Test Plan, Documentation Plan, Training Plan and Configuration Management Plan derived from the requirements are also verified and approved.

Critical Design Review

This review baselines the Detailed Design Specification (the “build to” document). Normally, coding officially begins at the close of this review. Test cases are also reviewed and approved.

Test Readiness Review

This review is performed when the appropriate application components are near completion. The review determines the readiness of the application or project for system and acceptance testing.

It is important to note that although the completion of a phase-end review signals the formal beginning of the next phase, subsequent phases may have already been started. In fact, in iterative development methodologies, each analysis or design “package” or segment of the application may be in a different phase of the project simultaneously. Careful analysis and planning are critical to ensure that the iterations are sequenced appropriately to minimize the risk of a defect found in one iteration causing excessive rework in previous iterations.

Post-Implementation Reviews

Post-implementation reviews (also known as “postmortems”) are conducted in a formal format up to six months after implementation is complete, in order to audit the process based on actual results. They are held to assess the success of the overall process after release, and to identify any opportunities for process improvement.

These reviews focus on questions such as: “Is the quality what was expected?” “Did the process work?” “Would buying a tool have improved the process?” or “Would automation have sped up the process?” Post-implementation reviews are of value only if some use is made of the findings. The quality assurance practitioner draws significant insight into the processes used and their behaviors.

Inspections

Inspections are formal manual techniques that are a natural evolution of desk checking. This procedure requires a team, usually directed by a moderator. The team includes the developer, but the remaining members and the moderator should not be directly involved in the development effort. Both techniques are based on a reading of the product (e.g., requirements, specifications, or code) in a formal meeting environment with specific rules for evaluation. The difference between inspection and walkthrough lies in the conduct of the meeting. Both methods require preparation and study by the team members, and scheduling and coordination by the team moderator.

Inspection involves a step-by-step reading of the product, with each step checked against a predetermined list of criteria. These criteria include checks for historically common errors. Guidance for developing the test criteria can be found elsewhere. The developer is usually required to narrate the reading product. The developer finds many errors just by the simple act of reading aloud. Others, of course, are determined because of the discussion with team members and by applying the test criteria.

At the problem definition stage, inspections can be used to determine if the requirements satisfy the testability and adequacy measures as applicable to this stage in the development. If formal requirements are developed, formal methods, such as correctness techniques, may be applied to ensure adherence with the quality factors.

Inspections should be performed at the preliminary and detailed design stages. Design inspections will be performed for each module and module interface. Adequacy and testability of the module interfaces are very important. Any changes that result from these analyses will cause at least a partial repetition of the verification at both stages and between the stages. A reexamination of the problem definition and requirements may also be required.

Finally, the inspection procedures should be performed on the code produced during the construction stage. Each module should be analyzed separately and as integrated parts of the finished software.

Developing Testing Methodologies

The eight considerations listed below provide the framework for developing testing tactics. Each is described in the following sections.

- Acquire and study the test strategy
- Determine the type of development project
- Determine the type of software system
- Determine the project scope
- Identify the tactical risks
- Determine when testing should occur
- Build the tactical test plan
- Build the unit test plans

Acquire and Study the Test Strategy

A team familiar with the business risks associated with the software normally develops the test strategy, and the test team develops the tactics. Thus, the test team needs to acquire and study the test strategy, focusing on the following questions:

- What is the relationship of importance among the test factors?

- Which of the high-level risks are the most significant?
- Who has the best understanding of the impact of the identified business risks?
- What damage can be done to the business if the software fails to perform correctly?
- What damage can be done to the business if the software is not completed on time?

Determine the Type of Development Project

The type of project refers to the environment in which the software will be developed, and the methodology used. Changes to the environment also change the testing risk. For example, the risks associated with a traditional development effort are different from the risks associated with off-the-shelf purchased software. Table 17 illustrates characteristics and testing tactics that can be used for different types of projects.

Table 17. Characteristics and Test Tactics for Different Project Types

Project Type	Characteristics	Test Tactics
Traditional system development (and most perfective maintenance)	Uses a system development methodology User knows requirements Development determines structure	Test at end of each task, step and phase Verify that specs match need Test function and structure
Iterative development, prototyping, CASE	Requirements unknown Structure predefined	Verify that CASE tools are used properly Test functionality
System maintenance	Modify structure	Test structure Works best with release methods Requires regression testing
Purchased or contracted software	Structure unknown May contain defects Functionality defined in user documentation Documentation may vary from software	Test functionality Verify functionality matches need Test fit into environment

Determine the Type of Software System

The type of software system refers to the processing that will be performed by that system. There are sixteen different software system types; however, a single software system may incorporate more than one of these types. Identifying the specific combinations of software making up the project can help analyze lessons learned on past projects with similar types of software.

Batch (General)

Can be run as a normal batch job and makes no unusual hardware or input-output actions (e.g., payroll program and wind tunnel data analysis program).

Event Control	Processes real-time data from external events, such as a computer program that processes telemetry data.
Process Control	Receives data from an external source and issues commands to that source to control its actions based on the received data.
Procedure Control	Controls other software; for example, an operating system that controls execution of time-shared and batch computer programs.
Advanced Mathematical Models	Resembles simulation and business strategy software, but has the additional complexity of heavy use of mathematics.
Message Processing	Handles input and output messages, processing the text, or information contained therein.
Diagnostic Software	Detects and isolates hardware errors in the computer where it resides, or in other hardware that can communicate with that computer.
Sensor and Signal Processing	Similar to message processing, but it requires greater processing to analyze and transform the input into a usable data processing format.
Simulation	Simulates an environment, mission situation, or other hardware. Uses inputs from these to enable a more realistic evaluation of a computer program or a piece of hardware.
Database Management	Manages the storage and access of (typically large) groups of data. Such software can also prepare reports in user-defined formats based on the contents of the database.
Data Acquisition	Receives information in real-time and stores it in some form suitable for later processing; for example, software that receives data from a space probe and files it for later analysis.
Data Presentation	Formats and transforms data, as necessary, for convenient and understanding displays; typically, such displays would be for some screen presentation.

Decision and Planning Aids	Uses artificial intelligence techniques to provide an expert system to evaluate data and provide additional information and consideration for decision and policy makers.
Pattern and Image Processing	Generates and processes computer images; such software may analyze terrain data and generate images based on stored data.
Computer System Software	Provides services to operational computer programs (i.e., coordinates processing of components required to meet need).
Software Development Tools	Provides services to aid in the development of software (e.g., compilers, assemblers, static and dynamic analyzers).

Determine the Project Scope

The project scope refers to the totality of activities to be incorporated into the software system being tested – the range of system requirements and specifications to be understood. The scope of the testing effort is usually defined by the scope of the project. New system development has a much different scope from modifications to an existing system. When defining the scope, consider the following characteristics and then expand the list to encompass the requirements of the specific software system being tested.

New Systems Development

- Automating manual business process?
- Which business processes will or won't be affected?
- Which business areas will or won't be affected?
- Interfacing to existing systems?
- Existing systems will or won't be affected?

Changes to Existing Systems

- Corrective only?
- Maintenance reengineering standards?
- Correction to known latent defects in addition to enhancements?
- Other systems affected?
- Risk of regression?

Identify the Tactical Risks

Strategic risks are the high-level business risks faced by the software system. They are decomposed into tactical risks to assist in creating the test scenarios that will address those risks. It is difficult to create test scenarios for high-level risks.

Tactical risks are divided into three categories:

- **Structural Risks**
These risks are associated with the application and the methods used to build it.
- **Technical Risks**
These risks are associated with the technology used to build and operate the application.
- **Size risks**
These risks are associated with the magnitude in all aspects of the software.

Determine When Testing Should Occur

The previous steps have identified the type of development project, the type of software system, the type of testing, the project scope, and the tactical risks. That information should be used to determine the point in the development process at which testing should occur.

For new development projects, testing can, and should, occur throughout the phases of a project. For modifications to existing systems, any or all of these may be applicable, depending on the scope. Examples of test activities to be performed during these phases are:

Requirements Phase Activities

- Determine test strategy
- Determine adequacy of requirements
- Generate functional test conditions

Design Phase Activities

- Determine consistency of design with requirements
- Determine adequacy of design
- Determine adequacy of the test plans
- Generate structural and functional test conditions

Program (Build) Phase Activities

- Determine consistency with design
- Determine adequacy of implementation

- Generate structural and functional test conditions for modules and units

Test Phase Activities

- Test application system
- Installation Phase Activities
- Place tested system into production

Maintenance Phase Activities

- Modify and retest

Build the System Test Plan

Using information from the prior steps, develop a System Test Plan to describe the testing that will occur. This plan will provide background information on the system being tested, test objectives and risks, the business functions to be tested, and the specific tests to be performed.

The Test Plan is the road map that will be followed when conducting testing. The plan is then decomposed into specific tests and lower-level plans. After execution, the results from the specific tests are rolled up to produce a Test Report.

Build the Unit Test Plans

During internal design, the system is divided into the components or units that perform the detailed processing. Each of these units should have an individual Test Plan. The plans can be as simple or as complex as the organization requires based on its quality expectations.

The importance of a Unit Test Plan is to determine when unit testing is complete. It is not cost effective to submit units that contain defects to higher levels of testing. The extra effort spent in developing Unit Test Plans, testing units, and assuring that units are defect free prior to integration testing can have a significant payback in reducing overall test costs.

Verification and Validation Methods

Verification and validation represents both static testing (verification) and dynamic testing (validation). Together they comprise the test activities. The methods available for verification and validation are briefly described.

Management of Verification and Validation

Management of software development verification and validation (V&V) activities begins at the start of the project, and is performed for all software life cycle processes and activities. This activity continuously reviews the V&V effort, revises the Software V&V Plan as necessary based upon updated project schedules and development status, and coordinates the results with the project team. The V&V manager assesses each proposed change to the system and software, identifies the software requirements that are affected by the change and plans the V&V tasks to

address the change. Each proposed change must also be assessed to determine whether any new hazards or risks are introduced in, or eliminated from, the software. The V&V plan is revised as necessary by updating tasks or modifying the scope and intensity of existing V&V tasks.

At key project milestones, such as the requirements review, design review, or test readiness review, the V&V manager consolidates the V&V results to establish supporting evidence regarding whether to proceed to the next set of software development activities. Whenever necessary, it must also be determined whether a V&V task needs to be repeated as a result of changes in the application or work products.

The minimum tasks performed by V&V management include:

- Create the Software V&V Plan
- Conduct Management Review of V&V
- Support Management and Technical Reviews
- Interface with Organizational and Supporting Processes
- Creation of V&V

Verification Techniques

Verification is the process of confirming that interim deliverables have been developed according to their inputs, process specifications, and standards. Verification techniques are listed below.

Feasibility Reviews

Tests for this structural element verify the logic flow of a unit of software (e.g., verifying that the software could conceivably perform after the solution is implemented the way the developers expect). Output from this review is a preliminary statement of high-level market requirements that becomes input to the requirements definition process (where the detailed technical requirements are produced).

Requirements Reviews

These reviews examine system requirements to ensure they are feasible and that they meet the stated needs of the user. They also verify software relationships; for example, the structural limits of how much load (e.g., transactions or number of concurrent users) a system can handle. Output from this review is a statement of requirements ready to be translated into system design.

Design Reviews

These structural tests include study and discussion of the system design to ensure it will support the system requirements. Design reviews yield a system design, ready to be translated into software, hardware configurations, documentation and training.

Code Walkthroughs

These are informal, semi-structured reviews of the program source code against specifications and standards to find defects and verify coding techniques. When done, the computer software is ready for testing or more detailed code inspections by the developer.

Code Inspections or Structured Walkthroughs

These test techniques use a formal, highly structured session to review the program source code against clearly defined criteria (System Design Specifications, product standards) to find defects. Completion of the inspection results in computer software ready for testing by the developer.

Requirements Tracing

At each stage of the life cycle (beginning with requirements or stakeholder needs) this review is used to verify that inputs to that stage are correctly translated and represented in the resulting deliverables. Requirements must be traced throughout the rest of the software development life cycle to ensure they are delivered in the final product. This is accomplished by tracing the functional and non-functional requirements into analysis and design models, class and sequence diagrams, and test plans and code. The level of traceability also enables project teams to track the status of each requirement throughout the development and test process.

Validation Techniques

Validation assures that the end product (system) meets requirements and expectations under defined operating conditions. Within an IT environment, the end product is typically executable code. Validation ensures that the system operates according to plan by executing the system functions through a series of tests that can be observed and evaluated for compliance with expected results.

Table 18 illustrates how various techniques can be used throughout the standard test stages. Each technique is described below.

Table 18. Validation Techniques Used in Test Stages

Techniques \ Test Stages	White-box	Black-box	Incremental	Thread	Regression
Unit Test	X				X
String/Integration Test	X	X	X	X	X
System Test		X	X	X	X
Acceptance Test		X			X

White-Box

White-box testing (logic driven) assumes that the path of logic in a unit or program is known. White-box testing consists of testing paths, branch by branch, to produce predictable results. Multiple white-box testing techniques are listed below. These techniques can be combined as appropriate for the application, but should be limited, as too many techniques can lead to an unmanageable number of test cases.

Statement Coverage	Execute all statements at least once.
Decision Coverage	Execute each decision direction at least once.
Condition Coverage	Execute each decision with all possible outcomes at least once.
Decision/Condition Coverage	Execute all possible combinations of condition outcomes in each decision, treating all iterations as two-way conditions exercising the loop zero times and once.
Multiple Condition Coverage	Invoke each point of entry at least once.

When evaluating the paybacks received from various test techniques, white-box or program-based testing produces a higher defect yield than the other dynamic techniques when planned and executed correctly.

Black-Box

In black-box testing (data or condition driven), the focus is on evaluating the function of a program or application against its currently approved specifications. Specifically, this technique determines whether combinations of inputs and operations produce expected results. As a result, the initial conditions and input data are critical for black-box test cases.

Three successful techniques for managing the amount of input data required include:

Equivalence Partitioning

An equivalence class is a subset of data that represents a larger class. Equivalence partitioning is a technique for testing equivalence classes rather than undertaking exhaustive testing of each value of the larger class. For example, a program which edits credit limits within a given range (at least \$10,000 but less than \$15,000) would have three equivalence classes:

- Less than \$10,000 (invalid)
- Equal to \$10,000 but not as great as \$15,000 (valid)
- \$15,000 or greater (invalid)

Boundary Analysis

This technique consists of developing test cases and data that focus on the input and output boundaries of a given function. In the credit limit example, boundary analysis would test the:

- Low boundary plus or minus one (\$9,999 and \$10,001)
- Boundaries (\$10,000 and \$15,000)
- Upper boundary plus or minus one (\$14,999 and \$15,001)

Error Guessing

This is based on the theory that test cases can be developed from the intuition and experience of the tester. For example, in a test where one of the inputs is the date, a tester may try February 29, 2000 or February 29, 2001.

Incremental

Incremental testing is a disciplined method of testing the interfaces between unit-tested programs and between system components. It involves adding unit-tested programs to a given module or component one by one, and testing each resultant combination. There are two types of incremental testing:

Top-Down

This method of testing begins testing from the top of the module hierarchy and works down to the bottom using interim stubs to simulate lower interfacing modules or programs. Modules are added in descending hierarchical order.

Bottom-Up

This method of testing begins testing from the bottom of the hierarchy and works up to the top. Modules are added in ascending hierarchical order. Bottom-up testing requires the development of driver modules, which provide the test input, call the module or program being tested, and display test output.

There are pros and cons associated with each of these methods, although bottom-up testing is generally considered easier to use. Drivers tend to be less difficult to create than stubs, and can

serve multiple purposes. Output from bottom-up testing is also often easier to examine, as it always comes from the module directly above the module under test.

Thread

This test technique, which is often used during early integration testing, demonstrates key functional capabilities by testing a string of units that accomplish a specific function in the application. Thread testing and incremental testing are usually used together. For example, units can undergo incremental testing until enough units are integrated and a single business function can be performed, threading through the integrated components.

When testing client/server applications, these techniques are extremely critical. An example of an effective strategy for a simple two-tier client/server application could include:

1. Unit and bottom-up incrementally test the application server components.
2. Unit and incrementally test the GUI or client components.
3. Test the network.
4. Thread test a valid business transaction through the integrated client, server, and network.

Regression

There are always risks associated with introducing change to an application. To reduce this risk, regression testing should be conducted during all stages of testing after a functional change, reduction, improvement, or repair has been made. This technique assures that the change will not cause adverse effects on parts of the application or system that were not supposed to change. Regression testing can be a very expensive undertaking, both in terms of time and money. The test manager's objective is to maximize the benefits of the regression test while minimizing the time and effort required for executing the test.

The test manager must choose which type of regression test minimizes the impact to the project schedule when changes are made, and still assures that no new defects were introduced. The types of regression tests include:

Unit Regression Testing

This retests a single program or component after a change has been made. At a minimum, the developer should always execute unit regression testing when a change is made.

Regional Regression Testing

This retests modules connected to the program or component that have been changed. If accurate system models or system documentation are available, it is possible to use them to identify system components adjacent to the changed components, and define the appropriate set of test cases to be executed. A regional regression test executes a subset of the full set of

application test cases. This is a significant timesaving over executing a full regression test, and still helps assure the project team and users that no new defects were introduced.

Full Regression Testing

This retests the entire application after a change has been made. A full regression test is usually executed when multiple changes have been made to critical components of the application. This is the full set of test cases defined for the application.

When an application feeds data to another application, called the “downstream” application, a determination must be made whether regression testing should be conducted with the integrated application. Testers from both project teams cooperate to execute this integrated test, which involves passing data from the changed application to the downstream application, and then executing a set of test cases for the receiving application to assure that it was not adversely affected by the changes.

Structural and Functional Testing

Structural testing is considered white-box testing because knowledge of the internal logic of the system is used to develop test cases. Structural testing includes path testing, code coverage testing and analysis, logic testing, nested loop testing, and similar techniques. Unit testing, string or integration testing, load testing, stress testing, and performance testing are considered structural.

Functional testing addresses the overall behavior of the program by testing transaction flows, input validation, and functional completeness. Functional testing is considered black-box testing because no knowledge of the internal logic of the system is used to develop test cases. System testing, regression testing, and user acceptance testing are types of functional testing.

As part of verifying and validating the project team’s solution, testers perform structural and functional tests that can be applied to every element of a computerized system. Both methods together validate the entire system. For example, a functional test case might be taken from the documentation description of how to perform a certain function, such as accepting bar code input. A structural test case might be taken from a technical documentation manual. To effectively test systems, both methods are needed. Each method has its pros and cons, which are listed below:

Structural Testing

Advantages

The logic of the software’s structure can be tested.

Parts of the software will be tested which might have been forgotten if only functional testing was performed.

Disadvantages

Its tests do not ensure that user requirements have been met.

Its tests may not mimic real-world situations.

Functional Testing

Advantages

Simulates actual system usage.

Makes no system structure assumptions.

Disadvantages

Potential of missing logical errors in software.

Possibility of redundant testing.

Software Change Control

Controlling software changes requires both a configuration management process and a change control process. Both are described in this section.

Software Configuration Management

The dynamic nature of most business activities causes software or system changes. Changes require well-formulated and well-documented procedures to prevent the manipulation of programs for unauthorized purposes. The primary objective of configuration management (or change control) is to get the right change installed at the right time. Change control concerns should be identified so that proper control mechanisms can be established to deal with the concerns.

Some key points regarding changes include:

- Each release of software, documentation, database, etc., should have a unique version number. Changes should be incorporated through new versions of the program. There should be a process for moving versions in and out of production on prescribed dates.
- Procedures should exist for maintaining the production and source libraries. They should address when to add to the library and when prior versions should be deleted. Care should be taken to regularly review libraries for obsolete programs, as large libraries can negatively impact operations performance.
- Project documentation such as requirements specifications, design documents, test plans, standards, procedures, and guidelines should also be identified with version numbers and kept under version control to ensure the project team is working with the latest, approved documents.
- Other environmental considerations to keep under version control are the operating system and hardware, as changes to either of these have the potential for impacting the project.

- Testing will not uncover all of the problems. As a result, people should be assigned to review output immediately following changes. If this is a normal function, then those people should be notified that a change has occurred.
- Each time an application is changed, the backup data required for recovery purposes may also have to be changed. Since this step occurs outside the normal change procedures, it may be overlooked. Backup data includes the new program versions, the job control language associated with those programs and other documentation procedures involved in making the system operational after a problem occurs.
- Modifying an application system may also require modifying the recovery procedures. If new files have been established, or if new operating procedures or priorities have been designed, they must be incorporated into the recovery procedures.

Change Control Procedures

Several procedures are necessary to maintain control over program changes.

- The nature of the proposed change should be explained in writing, and formally approved by a responsible individual. Major changes should be approved by the systems-planning steering committee, commonly called the CCB or Configuration Control Board, in the same manner as for new systems. Minor changes may only require the joint approval of the IT manager and senior personnel in the user department. Documenting the proposed change clears up any initial misunderstandings that may arise when only verbal requests are made. In addition, written proposals provide a history of changes in a particular system.
- Developers should make the program changes, not the operations group. Any change should be supported by adequate systems documentation. If the operators were authorized to make minor changes, it would greatly increase the difficulty of controlling versions and of maintaining up-to-date documentation.
- Someone independent of the person who designed and made the change should be responsible for testing the final revised program. The results should be recorded on program change registers and sent to the IT manager for approval. Operations should accept only properly approved changes.
- Finally, the documentation system should be updated with all change sheets or change registers and printouts.

Defect Management

A defect is a variance from expectations. To manage defects properly requires a process that prevents, discovers, tracks, resolves, and improves processes to reduce future defect occurrences.

Defect Management Process

The general principles of a Defect Management Process are as follows:

- The primary goal is to prevent defects. Where this is not possible or practical, the goals are to find the defect as quickly as possible and to minimize the impact of the defect.
- The defect management process, like the entire software development process, should be risk driven, i.e., strategies, priorities, and resources should be based on an assessment of the risk and the degree to which the expected impact of a risk can be reduced (see Skill Category 8).
- Defect measurement should be integrated into the development process. Information on defects should be captured at the source as a natural by-product of doing the job. It should not be done after the fact by people unrelated to the project or system.
- As much as possible, the capture and analysis of the information should be automated. The QA analyst should look for trends and perform a root-cause analysis to identify special and common cause problems.
- Defect information should be used to improve the process. As imperfect or flawed processes cause most defects, processes may need to be altered to prevent defects.

Defect Reporting

Recording the defects identified at each stage of the test process is an integral part of a successful life cycle testing approach. The purpose of this activity is to create a complete record of the discrepancies identified during testing. The information captured is used in multiple ways throughout the project, and forms the basis for quality measurement.

A defect can be defined in one of two ways. From the producer's viewpoint, a defect is a deviation from specifications, whether missing, wrong, or extra. From the Customer's viewpoint, a defect is anything that causes customer dissatisfaction, whether in the requirements or not. It is critical that defects identified at each stage of the life cycle be tracked to resolution.

Defects are recorded for four major purposes:

- To ensure the defect is corrected
- To report status of the application
- To gather statistics used to develop defect expectations in future applications
- To improve the software development process

Most project teams use some type of tool to support the defect tracking process. This tool could be as simple as a white board or a table created and maintained in a word processor, or one of the more robust tools available today on the market. Tools marketed for this purpose usually come with a number of customizable fields for tracking project specific data in addition to the basics. They also provide advanced features such as standard and ad-hoc reporting, e-mail notification to developers or testers when a problem is assigned to them, and graphing capabilities.

At a minimum, the tool selected should support the recording and communication of all significant information about a defect. For example, a defect log could include:

- Defect ID number
- Descriptive defect name and type
- Source of defect - test case or other source that found the defect
- Method – SDLC phase of creation
- Phases – SDLC phase of detection
- Component or program that had the defect
- Defect severity
- Defect priority
- Defect status (e.g., open, fixed, closed, user error, design, and so on) - more robust tools provide a status history for the defect
- Date and time tracking for either the most recent status change, or for each change in the status history
- Detailed description, including the steps necessary to reproduce the defect
- Screen prints, logs, etc., that will aid the developer in resolution process
- Stage of origination
- Persons assigned to research and correct the defect

Severity versus Priority

Based on predefined severity descriptions, the test team should assign the severity of a defect objectively. For example a “severity one” defect may be defined as one that causes data corruption, a system crash, security violations, etc. Severity levels should be defined at the start of the project so that they are consistently assigned and understood by the team. This foresight can help test teams avoid the common disagreements with development teams about the criticality of a defect.

In large projects, it may also be necessary to assign a priority to the defect, which determines the order in which defects should be fixed. The priority assigned to a defect is usually more subjective as it may be based on input from users regarding which defects are most important, resources available, risk, etc.

A Sample Defect Tracking Process

The steps below describe a sample defect tracking process. Depending on the size of the project or project team, this process may be substantially more complex.

1. Execute test and log any discrepancies.

The tester executes the test and compares the actual results to the documented expected results. If a discrepancy exists, the discrepancy is logged as a “defect” with a

status of “open.” Supplementary documentation, such as screen prints or program traces, is attached if available.

2. Determine if discrepancy is a defect.

The Test Manager or tester reviews the defect log with an appropriate member of the development team to determine if the discrepancy is truly a defect, and is repeatable. If it is not a defect, or repeatable, the log should be closed with an explanatory comment.

3. Assign defect to developer.

If a defect exists it is assigned to a developer for correction. This may be handled automatically by the tool, or may be determined as a result of the discussion in step 2.

4. Defect resolution process.

When the developer has acknowledged the defect is valid, the resolution process begins. The four steps of the resolution process are:

- Prioritize the correction.

Three recommended prioritization levels are: “critical”, “major”, and “minor”. “Critical” means there is a serious impact on the organization’s business operation or on further testing. “Major” causes an output of the software to be incorrect or stops or impedes further testing. “Minor” means something is wrong, but it does not directly affect the user of the system or further testing, such as a documentation error or cosmetic GUI error.

The purpose of this step is to initiate any immediate action that may be required after answering the questions: Is this a new or previously reported defect? What priority should be given to correcting this defect? Should steps be taken to minimize the impact of the defect before the correction, such as notifying users, finding a workaround?

- Schedule the correction.

Based on the priority of the defect, the correction should be scheduled. All defects are not created equal from the perspective of how quickly they need to be corrected, although they may all be equal from a defect-prevention perspective. Some organizations actually treat lower priority defects as changes.

- Correct the defect.

The developer corrects the defect, and upon completion, updates the log with a description of the correction and changes the status to “Corrected” or “Retest”. The tester then verifies that the defect has been removed from the system.

Additional regression testing is performed as needed based on the severity and impact of the correction applied. In addition, test data, checklists, etc., should be reviewed and perhaps enhanced, so that in the future this defect will be caught

earlier. If the retest results match the expected results, the tester updates the defect status to “closed.” If the problem remains, the tester changes the status back to “Open” and this step is repeated until closure.

- Report the resolution.

Once the defect has been corrected and the correction verified, appropriate developers, users, etc., need to be notified that the defect has been corrected, the nature of the correction, when the correction will be released, and how the correction will be released.

As in many aspects of defect management, this is an area where an automated process would help. Most defect management tools capture information on who found and reported the problem and therefore provide an initial list of who needs to be notified. Computer forums and electronic mail can help notify users of widely distributed software.

Test Reports are issued periodically throughout the testing process to communicate the test status to the rest of the team and management. These reports usually include a summary of the open defects, by severity or priority. Additional graphs and metrics can also be provided to further describe the status of the application.

Using Defects for Process Improvement

Using defects to improve processes is not done by many organizations today, but it offers one of the greatest areas of payback. NASA emphasizes the point that any defect represents a weakness in the process. Seemingly unimportant defects are, from a process perspective, no different from critical defects. It is only the developer’s good luck that prevents a defect from causing a major failure. Even minor defects, therefore, represent an opportunity to learn how to improve the process and prevent potentially major failures. While the defect itself may not be a big deal, the fact that there was a defect is a big deal.

Based on the research team findings, this activity should include the following:

- Go back to the process that originated the defect to understand what caused the defect
- Go back to the verification and validation process, which should have caught the defect earlier

Not only can valuable insight be gained as to how to strengthen the review process, these steps make everyone involved in these activities take them more seriously. This human factor dimension alone, according to some of the people the research team interviewed, can have a very large impact on the effectiveness of the review process.

NASA takes an additional step of asking the question: “If this defect could have gotten this far into the process before it was captured, what other defects may be present that have not been discovered?” Thus, not only is the process strengthened to prevent defects, it is strengthened to find defects which have been created but not yet discovered. This aggressiveness should be mandatory on life-critical systems.

Metrics and Measurement

A properly established measurement system is used to help achieve missions, visions, goals, and objectives. Measurement data is most reliable when it is generated as a by-product of producing a product or service. The QA analyst must ensure that quantitative data is valued and reliable, and presented to management in a timely and easy-to-use manner. Measurement can be used to gauge the status, effectiveness and efficiency of processes, customer satisfaction, product quality, and as a tool for management to use in their decision-making processes. This category addresses measurement concepts, the use of measurement in a software development environment, variation, process capability, risk management, the ways measurement can be used and how to implement an effective measurement program.

<i>Measurement Concepts</i>	339
<i>Measurement in Software</i>	344
<i>Variation and Process Capability</i>	348
<i>Risk Management</i>	357
<i>Implementing a Measurement Program</i>	367

Measurement Concepts

To effectively measure, one needs to know the basic concepts of measurement. This section provides those basic measurement concepts.

Standard Units of Measure

A measure is a single quantitative attribute of an entity. It is the basic building block for a measurement program. Examples of measures are lines of code (LOC), work effort, or number of defects. Since quantitative measures can be compared, measures should be expressed in numbers.

For example, the measure LOC refers to the “number” of lines and work effort refers to the “number” of hours, days, or months.

Measurement cannot be used effectively until the standard units of measure have been defined. For example, talking about lines of code does not make sense until the measure LOC has been defined. Lines of code may mean LOC written, executable LOC written, or non-compound LOC written. If a line of code contained a compound statement (such as a nested IF statement two levels deep) it could be counted as one or two lines of code. Additionally, organizations may use weighting factors; for example, one verb would be weighted as more complete than other verbs in the same programming language.

Standard units of measure are the base on which all measurement exists. Measurement programs typically have between five and fifty standard units.

Metrics

A metric is a derived (calculated or composite) unit of measurement that cannot be directly observed, but is created by combining or relating two or more measures. A metric normalizes data so that comparison is possible. Since metrics are combinations of measures they can add more value in understanding or evaluating a process than plain measures. Examples of metrics are mean time to failure and actual effort compared to estimated effort.

Objective and Subjective Measurement

Objective measurement uses hard data that can be obtained by counting, stacking, weighing, timing, etc. Examples include number of defects, hours worked, or completed deliverables. An objective measurement should result in identical values for a given measure, when measured by two or more qualified observers.

Subjective data is normally observed or perceived. It is a person's perception of a product or activity, and includes personal attitudes, feelings and opinions, such as how easy a system is to use, or the skill level needed to execute the system. With subjective measurement, even qualified observers may determine different values for a given measure, since their subjective judgment is involved in arriving at the measured value. The reliability of subjective measurement can be improved through the use of guidelines, which define the characteristics that make the measurement result one value or another.

Objective measurement is more reliable than subjective measurement, but as a general rule, subjective measurement is considered more important. The more difficult something is to measure, the more valuable it is. For example, it is more important to know how effective a person is in performing a job (subjective measurement), than knowing they got to work on time (objective measurement). Following are a few other examples of objective and subjective measures:

- The size of a software program measured in LOC is an objective product measure. Any informed person, working from the same definition of LOC, should obtain the same measure value for a given program.

- The classification of software as user-friendly is a subjective product measure. For a scale of 1-5, customers of the software would likely rate the product differently. The reliability of the measure could be improved by providing customers with a guideline that describes how having or not having a particular attribute affects the scale.
- Development time is an objective process measure.
- Level of programmer experience is a subjective process measure.

Types of Measurement Data

Before measurement data is collected and used, the type of information involved must be considered. It should be collected for a specific purpose. Usually the data is used in a process model, used in other calculations, or is subjected to statistical analyses. Statisticians recognize four types of measured data, which are summarized in Table 19 and described below.

Table 19. Types of Measured Data

Operations for a data type also apply to all data types appearing below it.

Data Type	Possible Operations	Description of Data
Nominal	= ≠	Categories
Ordinal	< >	Rankings
Interval	+ -	Differences
Ratio	/	Absolute Zero

Nominal Data

This data can be categorized. For example, a program can be classified as database software, operating system, etc. Nominal data cannot be subjected to arithmetic operations of any type, and the values cannot be ranked in any "natural order." The only possible operation is to determine whether something is the same type as something else. Nominal data can be objective or subjective, depending on the rules for classification.

Ordinal Data

This data can be ranked, but differences or ratios between values are not meaningful. For example, programmer experience level may be measured as low, medium, or high. For ordinal data to be used in an objective measurement the criteria for placement in the various categories must be well defined; otherwise, it is subjective.

Interval Data

This data can be ranked and can exhibit meaningful differences between values. Interval data has no absolute zero, and ratios of values are not necessarily meaningful. For example, a program with a complexity value of 6 is four units more complex than a program with a complexity of 2, but it

is probably not meaningful to say that the first program is three times as complex as the second. T. J. McCabe's complexity metric is an example of an interval scale.

Ratio Data

This data has an absolute zero and meaningful ratios can be calculated. Measuring program size by LOC is an example. A program of 2,000 lines can be considered twice as large as a program of 1,000 lines, and programs can have zero length.

It is important to understand the measurement scale associated with a given measure or metric. Many proposed measurements use values from an interval, ordinal, or nominal scale. If the values are to be used in mathematical equations designed to represent a model of the software process, measurements associated with a ratio scale are preferred, since the ratio scale allows mathematical operations to be meaningfully applied.

Measures of Central Tendency

The measures of central tendency are the mean, medium, and mode. The *mean* is the average of the items in the population; the *medium* is the item at which half the items in the population are below this item and half the items are above this item; and the *mode* represents which items are repeated most frequently.

For example, if a population of numbers are: 1, 2, 2, 3, 4, 5, and 11:

- The mean is "4" because $1 + 2 + 2 + 3 + 4 + 5 + 11 = 28$ and $28 \div 7 = 4$.
- The medium is "3" because there are three values less and three values higher than 3.
- The mode is "2" because that is the item with the most occurrences.

Attributes of Good Measurement

Ideally models should be developed that are capable of predicting process or product parameters, not just describing them. This is facilitated by measures and resulting metrics that are:

- Simple and precisely definable, so it is clear how they can be evaluated
- Objective
- Easily obtainable at reasonable cost
- Valid, measuring what they are intended to measure
- Robust, being relatively insensitive to intuitively small changes in the process or product

Before being approved for use, measures and metrics should be subjected to the following six tests described below.

Reliability

This test refers to the consistency of measurement. If taken by two people, the same results should be obtained. Sometimes measures are unreliable because of the measurement technique. For example, human error could make counting LOC unreliable, but the use of an automated code analyzer would result in the same answer each time it is run against an unchanged program.

Validity

This test indicates the degree to which a measure actually measures what it was intended to measure. If actual project work effort is intended to quantify the total time spent on a software development project, but overtime or time spent on the project by those outside the project team is not included, the measure is invalid for its intended purpose. A measure can be reliable, but invalid. An unreliable measure cannot be valid.

Ease of Use and Simplicity

These two tests are functions of how easy it is to capture and use the measurement data.

Timeliness

This test refers to whether the information can be reported in sufficient time to impact the decisions needed to manage effectively.

Calibration

This test indicates the modification of a measurement so it becomes more valid; for example, modifying a customer survey to better reflect the true opinions of the customer.

Using Quantitative Data to Manage an IT Function

An integral part of an IT function is quantitative management, which contains two aspects: measurement dashboards and statistical process control.

Measurement Dashboards

Measurement dashboards (also called key indicators) are used to monitor progress and initiate change. A measurement dashboard is analogous to the dashboard on a car. Various key indicators are presented in comparison to their own desired target value (i.e., speed, oil temperature). Arrayed together, they provide an overall snapshot of the car's performance, health, and operating quality. Measurement dashboards help ensure that all critical performance areas are analyzed in relation to other areas. Indicators evaluated alone can lead to faulty conclusions and decision-making.

Measurable results defined by an organization can be organized into dashboards for different levels of management. Line managers use process or tactical dashboards to manage the process. Senior management uses strategic dashboards to manage the function or organization and track to

mission, vision, or goals. Using dashboards is known as “management by fact.” Figure 67 depicts strategic and tactical dashboards.

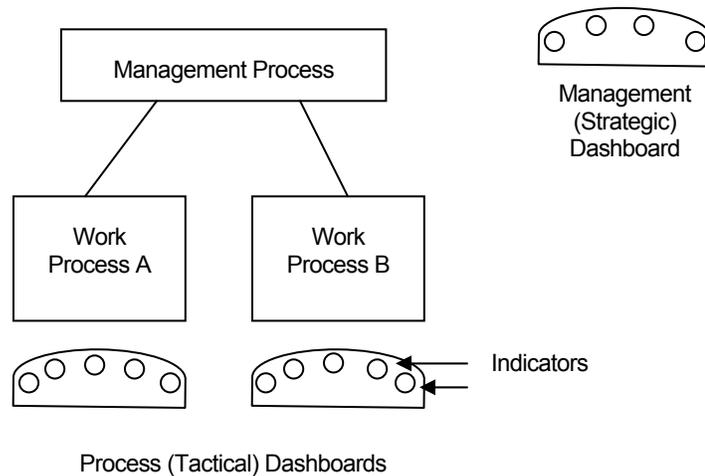


Figure 67. Strategic and Tactical Measurement Dashboards

Statistical Process Control

Statistical process control is used to ensure that the process behaves in a consistent manner. Line managers use the principles of statistical process control to assess consistency of products and services, and as a basis for continuous process improvement.

Key Indicators

Key indicators are the metrics used by management to help them fulfill their management responsibilities. Managers decide what key metrics they need. A dashboard is comprised of the total number of key indicators used by a single manager. The concept of “key” means the metric is considered important by the user in managing job responsibilities. Normally the key indicator is created from many different measures. For example, the Dow Jones Average is created from the combining of stock prices from approximately 40 different stocks.

Measurement in Software

The use of measurement in the software life cycle requires the development and use of software metrics, which are standardized through the use of defined units of measure. This measurement program enables management to control and manage software throughout its entire life.

Both the software product and the process by which it is developed can be measured. The software product should be viewed as an abstract object that evolves from an initial statement of need to a finished software system, including source and object code and the various forms of documentation produced during development. The software metrics are studied and developed for use in modeling the software development process. These metrics and models are used to estimate and predict product costs and schedules, and to measure productivity and product quality.

Information gained from the measurements and models can be used in the management and control of the development process, leading to improved results.

There is no clearly defined, commonly accepted way of measuring software products and services. A large number of measures and metrics exist, but only a few have had widespread use or acceptance. Even with those widely studied, such as LOC or T. J. McCabe's cyclomatic complexity, it is not universally agreed what they mean. Some studies have attempted to correlate the measurements with a number of software properties, including size, complexity, reliability (error rates), and maintainability.

Additionally, many measurements are done subjectively, such as product type or level of programming expertise. They are difficult to evaluate because of the potentially large number of factors involved and the problems associated with assessing or quantifying individual factors.

As for the proposed process models, few have a significant theoretical basis. Most are based upon a combination of intuition, expert judgment, and statistical analysis of empirical data. Many software measures and metrics have been defined and tested but used only in limited environments. There is no single process model that can be applied with a reasonable degree of success to a variety of environments. Generally, significant recalibration is required for each new environment in order to produce useful results. Furthermore, the various models often use a wide variety of basic parameter sets.

The above considerations make it difficult to interpret and compare quoted measurement results, especially with different environments, languages, applications, or development methodologies. Even simple measures, such as LOC, have differences in underlying definitions and counting techniques making it almost impossible to compare quoted results. Metrics involving LOC values across different program languages can lead to incorrect conclusions and thereby conceal the real significance of the data. For example, the productivity metrics LOC per month, and cost per LOC suggest that assembly language programmers are more productive than high-level language programmers (higher LOC per month and lower \$ per LOC), even though the total programming cost is usually lower for high-level languages. Similarly, defects per LOC and cost per defect values have been used as quality or productivity indicators. Again, with different levels of programming languages, using these measurements may obscure overall productivity and quality improvements by systematically yielding lower defect per LOC and cost per defect values for lower-level languages, even though total defects and costs are actually higher.

Despite these problems, applying software metrics and models in limited environments can help improve software quality and productivity. Defect density and McCabe's complexity have been found to be reasonably good predictors of other characteristics, such as defect counts, total effort, and maintainability. There are many useful products for measurement and modeling available on the market today. Additional experience with current models and a better understanding of underlying measurements and their application to the software process will improve the results.

Product Measurement

A product can be measured at any stage of its development. For a software product, the requirements, the complexity of the software design, the size of the final program's source or

object code, or the number of pages of documentation produced for the installed system can be measured.

Most of the initial work in measuring products has dealt with the characteristics of source code. Experience with measurement and models shows that measurement information available earlier in the development cycle can be of greater value in controlling the process and results. Thus, a number of papers have dealt with the size or complexity of the software design.

The following examples show various ways of measuring a product. These were chosen because of their wide use or because they represent a particularly interesting point of view.

Size

LOC is the most common way of quantifying software size; however, this cannot be done until the coding process is complete. Function points have the advantage of being measurable during the design phase of the development process or possibly earlier.

Lines of Code

This is probably the most widely used measure for program size, although there are many different definitions. The differences involve treatment of blank lines, comment lines, non-executable statements, multiple statements per line, multiple lines per statement, and the question of how to count reused lines of code. The most common definition counts any line that is not a blank or a comment, regardless of the number of statements per line. In theory, LOC is a useful predictor of program complexity, total development effort, and programmer performance (debugging, productivity). Numerous studies have attempted to validate these relationships.

Function Points

A. J. Albrecht proposed a metric for software size and the effort required for development that can be determined early in the development process. This approach computes the total function points (FP) value for the project, by totaling the number of external user inputs, inquiries, outputs, and master files, and then applying the following weights: inputs (4), outputs (5), inquiries (4), and master files (10). Each FP contributor can be adjusted within a range of $\pm 35\%$ for a specific project complexity.

Complexity

More metrics have been proposed for measuring program complexity than for any other program characteristic. Two examples of complexity metrics are:

Cyclomatic Complexity -- $v(G)$

Given any computer program, draw its control flow graph, G , where each node corresponds to a block of sequential code and each edge corresponds to a branch or decision point in the program. The cyclomatic complexity of such a graph can be computed by a simple formula from graph theory, as $v(G)=e-n+2$, where e is the number of edges, and n is the number of nodes in the graph.

Knots

Calculate program knots by drawing the program control flow graph with a node for every statement or block of sequential statements. A knot is defined as a necessary crossing of directional lines in the graph. The same phenomenon can be observed by drawing transfer-of-control lines from statement to statement in a program listing.

Quality

There is a long list of quality characteristics for software, such as correctness, efficiency, portability, performance, maintainability, and reliability. Skill Category 1 lists the commonly accepted quality attributes for an information system. While software quality can theoretically be measured at every phase of the software development cycle, the characteristics often overlap and conflict with one another. For example, increased performance or speed of processing (desirable) may result in lowered efficiency (undesirable). Since useful definitions are difficult to devise, most efforts to find any single way to measure overall software quality have been less than successful.

Although much work has been done in this area, there is still less direction or definition than for measuring software size or complexity. Three areas that have received considerable attention are: program correctness, as measured by defect counts; software reliability, as computed from defect data; and software maintainability, as measured by various other metrics, including complexity metrics.

Correctness

The number of defects in a software product should be readily derivable from the product itself. However, since there is no easy and effective procedure to count the number of defects in the program, the four alternative measures listed below have been proposed. These alternative measures depend on both the program and the outcome, or result, of some phase of the development cycle.

- Number of design changes
- Number of errors detected by code inspections
- Number of errors detected in program tests
- Number of code changes required

Reliability

It would be useful to know the probability of a software failure, or the rate at which software errors will occur. Again, although this information is inherent in the software product, it can only be estimated from data collected on software defects as a function of time. If certain assumptions are made, this data can be used to model and compute software reliability metrics. These metrics attempt to indicate and predict the probability of failure during a particular time interval, or the mean time to failure (MTTF) and mean time between failures (MTBF).

Maintainability

Efforts have been made to define ways to measure or predict the maintainability of a software product. An early study by Bill Curtis, and others, investigated the ability of Halstead's effort metric, E , and $v(G)$ to predict the psychological complexity of software maintenance tasks. Assuming such predictions could be made accurately, complexity metrics could then be profitably used to reduce the cost of software maintenance. A carefully detailed experiment indicated that software complexity metrics could effectively explain or predict the maintainability of software in a distributed computer system.

Customer Perception of Product Quality

Determining the customer's perception of quality involves measuring how the customer views the quality of the IT product. It can be measured in a number of ways, such as using customer surveys, service level agreements, loyalty, and recommendations to others.

Process Measurement

A process can be measured by either of the following:

- Attributes of the process, such as overall development time, type of methodology used, or the average level of experience of the development staff.
- Accumulating product measures into a metric so that meaningful information about the process can be provided. For example, function points per person-month or LOC per person-month can measure productivity (which is product per resources), the number of failures per month can indicate the effectiveness of computer operations, and the number of help desk calls per LOC can indicate the effectiveness of a system design methodology.

There is no standardized list of software process metrics currently available. However, in addition to the ones listed above, some others to consider include:

- Number of deliverables completed on time
- Estimated costs vs. actual costs
- Budgeted costs vs. actual costs
- Time spent fixing errors
- Wait time
- Number of contract modifications
- Number of proposals submitted vs. proposals won
- Percentage of time spent performing value-added tasks

Variation and Process Capability

Dr. W. Edwards Deming's quality principles call for statistical evidence of quality. He was a strong proponent of the use of statistics that took into account common and special causes of

variation. Common causes are those that can be controlled by improving the work processes. Special causes are those that must be controlled outside the process; typically they need to be dealt with individually. It is generally not cost-effective or practical to deal with special causes in the day-to-day work processes.

The natural changes occurring in organizations are moving systems and processes toward increasing variation. As a result, it is important for the QA analyst to understand the difference between common and special causes. Since the key to quality is process consistency, variation (the lack of consistency) must be understood before any process can be improved. Statistical tools are the only methods available to objectively quantify variation, and to differentiate between the two types. Control charts are the tools used to monitor variation, and they are discussed in Skill Category 4.

The Measurement Program

A measurement program is defined as the entire set of activities that occur around quantitative data. It can be as simple as measuring whether a system is completed on time or completed within budget, or it can be extensive and complex.

Quantitative measurement occurs at all levels of IT maturity. As organizations mature, their use of measurement changes with the maturation of the management approaches. Immature organizations typically measure for budget, schedule, and project status, and management relies on project teams to determine when requirements are done. When work processes are optimized, management relies on the quantitative data produced from the processes to determine whether or not the requirements are complete, and to prevent problems.

There are four major uses of quantitative data (i.e., measurement):

1. Manage and control the process.

A process is a series of tasks performed to produce deliverables or products. IT processes usually combine a skilled analyst with the tasks defined in the process. In addition, each time a process is executed it normally produces a different product or service from what was built by the same process at another time. For example, the same software development process may be followed to produce two different applications. Management may need to adapt the process for each product or service built, and needs to know that when performed, the process will produce the desired product or service.

2. Manage and control the product.

Quality is an attribute of a product. Quality level must be controlled from the start of the process through the conclusion of the process. Control requires assuring that the specified requirements are implemented, and that the delivered product is what the customer expects and needs.

3. Improve the process.

The most effective method for improving quality and productivity is to improve the processes. Improved processes have a multiplier effect because everyone that uses the improved process gains from the improvement. Quantitative data gathered during process execution can identify process weaknesses, and, therefore, opportunities for improvement.

4. Manage the risks.

Risk is the opportunity for something to go wrong - for example, newly purchased software will not work as stated, projects will be delivered late, or workers assigned to a project do not possess the skills needed to successfully complete it. Management needs to understand each risk, know the probability of the risk occurring, know the potential consequences if the risk occurs, and understand the probability of success based upon different management actions.

The same database of quantitative data is employed for these four uses, but different measures and metrics may be utilized. Table 20 illustrates how the four uses of measurement can be achieved.

Table 20. Achieving the Four Uses of Measurement

Use	Questions Answered	Measurement Category	Examples of Measures/Metrics Used
Manage and Control the Process	- How much have we made? - How much is left to make?	Size	- Lines of code (LOC) - Boxes - Procedures - Units of output
	How much progress have we made?	Status	- Earned Value - Amount of scheduled work that is done - % of each activity completed
	How much effort has been expended?	Effort	Labor hours that differentiate requirements, design, implementation and test
	When will the product be completed?	Schedule	Calendar times (months, weeks) of activity completed
Manage and Control the Product	How good is the product?	Quality	- Number of defects found - Mean time to failure - Mean time to repair
	How effectively does the product perform?	Performance	- Technical performance - Measures specified by customers and management
Improve the Process	- How cost-efficient is the process? - What is the current performance?	Time and effort	- Unit costs - Time to complete
Manage the Risks	What are the risks?	Risks	Probability of exceeding constraints or not meeting requirements

Installing the Measurement Program

Installation of a measurement program is a four-phased approach, with each phase containing multiple steps.

1. Build the Measurement base.

The objective of this phase is to create an environment in which the use of quantitative data is an accepted component of the management process. The four steps for accomplishing this are:

- Define the objectives for the measurement program - how it is to be used. Consider how to implement the four uses of measurement, given the maturity level of the organization. The use of measurement should be tied to the organization's mission, goals and objectives.
- Create an environment receptive to measurement. Begin with the prerequisites listed earlier in this section. Establish service level agreements between IT and the users to define quality and productivity that must be defined before they can be measured. People involved with the measurement should help develop the measure. Establish a quality management environment (see Skill Category 2) and ensure the work processes being used have been implemented.
- Define the measurement hierarchy, which has three levels of quantitative data: measures, metrics, and a strategic results dashboard (also called key indicators). This measurement hierarchy maps to a three-level IT organizational tier: staff, line management and senior management. IT staff collects basic measures, such as product size, cycle time, or defect count. IT line management uses fundamental metrics, such as variance between actual and budgeted cost, user satisfaction or defect rates per LOC to manage a project or part of the IT function. Senior management uses a strategic results dashboard, where the metrics represent the quantitative data needed to manage the IT function and track to the mission, vision, or goals. For example, a mission with a customer focus should have a customer satisfaction metric. A metric of the number of projects completed on time gives insight into the function's ability to meet short and long-term business goals.
- Define the standard units of measurement (discussed in Measurement Concepts).

2. Manage towards results.

In this five-step phase, goals for the desired business results are identified in the form of a strategic dashboard, and the means for measuring those results are determined. The business results need to be prioritized and communicated to the entire IT function so that decisions will be made in a manner that will facilitate achieving those results. This is particularly critical when the third phase is implemented, as the process results should link to the desired business results.

- Identify desired business results, beginning with a mission or vision statement. Turn operative phrases in the mission or vision (such as “deliver on time” or “satisfy customer”) into specific objectives (such as "all software will be delivered to the customer by the date agreed upon with the customer"), and then rank these objectives in order of importance. When objectives are written with a subject, action, target value, and timeframe it is much easier to identify the actual metric that will serve as the results metric or key indicator.

- Identify current baselines by determining the current operational status for each of the desired business results/objectives.
- Select a measure or metric for each desired business result or objective, and determine whether it has been standardized by the IT industry (such as cycle time, which is measured as elapsed calendar days from the project start date to the project end date). If not, explore the attributes of the result or objective and define a measure or metric that is quantitative, valid, reliable, attainable, easy to understand and collect, and a true representation of the intent. Ideally there should be three to five metrics, with no more than seven. Convert the business results metrics into a strategic dashboard of key indicators. Examples of indicators includes productivity, customer satisfaction, motivation, skill sets, and defect rates.
- Consider tradeoffs between the number one ranked business result and the other desired results. For example, the #1 result to complete on time will affect other desired results, such as minimize program size and develop easy-to-read documentation.
- Based on the baseline and desired business result or objective, determine a goal for each result metric. Goals typically specify a subject (such as financial, customer, process or product, or employee) and define an action that is change or control related (such as improve or reduce, increase or decrease or control or track). If a baseline for on time projects is 60%, the goal might be to increase to 80% by next year. Benchmarking (see Skill Category 4) can also be useful prior to setting goals, as it allows an understanding of what is possible given a certain set of circumstances.

3. Manage by process.

Managing by process means to use processes to achieve management's desired results. When results are not achieved, a quality management philosophy tells the organization to look at how the system (i.e., its processes) can be improved rather than reacting, making emotional decisions, and blaming people. Quantitative feedback, which provides indicators of process performance, is needed in order to operate this way. Various processes usually contribute jointly to meeting desired business results, and, therefore, it is important to understand and identify what things contribute to, or influence, desired results. This phase consists of four steps to implement measurement in a process, and to identify the attributes of the contributors, which if met will achieve the desired process results. These steps provide the information to manage a process and to measure its status.

- Develop a matrix of process results and contributors to show which contributors drive which results. The results should come from the process policy statement (see the discussion of the process workbench in Skill Category 6). The contributors can be positive or negative, and involve process, product, or resource attributes. Process attributes include characteristics such as time, schedule, and completion. Product attributes include characteristics such as size, correctness, reliability, usability, and maintainability. Resource attributes include characteristics such as amount, skill, and attitude. A cause-and-effect diagram as

illustrated in Skill Category 4 is often used to graphically illustrate the relationship between results and contributors.

- Assure process results are aligned to business results. Processes should help people accomplish their organization's mission. Alignment is subjective in many organizations, but the more objective it is, the greater the chance that processes will drive the mission.
- Rank the process results and the contributors from a management perspective. This will help workers make tradeoffs and identify where to focus management attention.
- Select metrics for both the process results and contributors, and create two tactical process dashboards: one for process results and one for contributors. These dashboards are used to manage the projects and to control and report project status. Normally results are measured subjectively and contributors are measured objectively. For example, for a result of customer satisfaction, contributors might include competent resources, an available process, and a flexible and correct product. Sometimes, as with customer satisfaction, factors that contribute to achieving the result can actually be used to develop the results metric. In other words, first determine what contributes to customer satisfaction or dissatisfaction and then it can be measured.

4. Management by fact.

Management by fact uses qualitative and quantitative data produced from and about work processes to make informed decisions regarding the operation of those work processes. Quantitative data can be objective (such as the number of defects produced) or subjective (such as the customer's perception of the quality of the products or services produced by the process). Typically the focus of decisions is common cause problems and special cause problems.

The management by fact process contains two components:

- Meeting desired results.
- Managing the processes to drive the results.

Common and Special Causes of Variation

Common Causes of Variation

All processes contain some inherent variation, or common causes of variation. The amount of variation in a process is quantified with summary statistics (the mean and standard deviation).

A process is defined as stable when its mean and standard deviation remain constant over time. Processes containing only common causes of variation are considered stable. As a stable process is predictable, future process values can be predicted within the control limits with a certain amount of belief. A stable process is said to be in a state of statistical control. The control chart in Skill Category 4 depicts a stable process.

In a computer operation, abnormal terminations cause variation. Typical common causes of abnormal terminations include invalid data, no available disk space, and errors in operating or job control instructions.

One researcher¹ provides the following thoughts on common causes of variation:

"Process inputs and conditions that regularly contribute to the variability of process outputs."

"Common causes contribute to output variability because they themselves vary."

"Each common cause typically contributes a small portion to the total variation in process outputs."

"The aggregate variability due to common causes has a 'nonsystematic,' random-looking appearance."

"Because common causes are 'regular contributors,' the 'process' or 'system' variability is defined in terms of them."

Joiner outlined this strategy for reducing common causes of variation:

"Talk to lots of people including local employees, other managers, and staff from various functions."

"Improve the measurement processes if measuring contributes too much to the observed variation."

"Identify and rank categories of problems by Pareto analysis."

"Stratify and desegregate your observations to compare performance of sub-processes."

"Investigate cause-and-effect relations, running experiments (one factor and multifactor)."

Special Causes of Variation

Special causes of variation are not present in a process. They occur because of special or unique circumstances. In the IT example of abnormal terminations in a computer operation, special causes might include operator strikes, citywide power outages, or earthquakes.

If special causes of variation exist, the process may be unpredictable, and therefore unstable. A state of statistical control is established when all special causes of variation have been eliminated (the operator strike ends, citywide power returns or business returns to normal operations after an earthquake). Illustrated in Skill Category 4 is a process that is unstable because it contains a special cause of variation in addition to the common causes.

¹Joiner, Brian, "Stable and Unstable Processes, Appropriate and Inappropriate Managerial Action." From an address given at a Deming User's Group Conference in Cincinnati, OH.

Brian Joiner summarized special causes of variation as follows:

"Process inputs and conditions that sporadically contribute to the variability of process outputs."

"Special causes contribute to output variability because they themselves vary."

"Each special cause may contribute a 'small' or 'large' amount to the total variation in process outputs."

"The variability due to one or more special causes can be identified by the use of control charts."

Because special causes are 'sporadic contributors,' due to some specific circumstances, the 'process' or 'system' variability is defined without them."

Joiner then presented this strategy for eliminating special causes of variation:

"Work to get very timely data so that special causes are signaled quickly - use early warning indicators throughout your operation."

"Immediately search for the cause when the control chart gives a signal that a special cause has occurred. Find out what was different on that occasion from other occasions."

"Do not make fundamental changes in the process."

"Instead, seek ways to change some higher-level systems to prevent that special cause from recurring. Or, if results are good, retrain that lesson."

Variation and Process Improvement

Consistency in all processes from conception through delivery of a product or service is the cornerstone of quality. One of the challenges in implementing quality management is to get process users thinking in terms of sources of variation. Managers must change the way they manage, and use statistical methods when making improvements to processes.

Employees using the process have the lead responsibility for reducing special causes of variation. Management working on the process is responsible for leading the effort to reduce common causes of variation. Improvements to address the common causes of variation usually require process or system changes. It has been widely recognized that at least 85% of problems in any organization are system problems and the responsibility of management to solve. Some sources quote 94%².

The concept of statistical control makes it possible to determine which problems are in a process due to common causes of variation and which are external to the process due to special causes of

²Deming, W. Edwards, *Out of the Crisis*, MIT Press, Cambridge, MA, 1986.

variation. Bringing a process into a state of statistical control is not improving the process - it is bringing it back to its typical operation. Reducing variation due to common causes is process improvement and the real essence of continuous process improvement.

By definition, process variation due to common causes is expected and is not a reason for adjusting or changing the process. Tampering is any adjustment made to a process in response to common cause variation. Deming defines tampering as "Action taken on a stable system in response to variation within statistical control, in an effort to compensate for this variation - the results of which will inevitably increase the variation and will increase cost from here on out." Management that does not understand variation continually asks for explanations or corrective action when confronted with variation due to common causes.

Process Capability

As previously stated, variation due to special causes must be removed to create a stable process. However, a stable process may not be an acceptable process. If the variation due to common causes results in a process operating outside of the customer specifications, the process is called "noncapable." The process must be improved by reducing variation due to common causes, retargeting the process, or both.

Figure 68 illustrates the transition of a process from noncapable to capable. In this figure, LCL and UCL represent lower and upper control limits. LSL and USL represent lower and upper specification limits. In the first picture, the control limits are outside the specification limits, so a value could be within the control limits but outside the specification limits, making the process noncapable. In the last picture the modified process results in different control limits, which have moved within the specification limits, yielding a process that is both stable and capable.

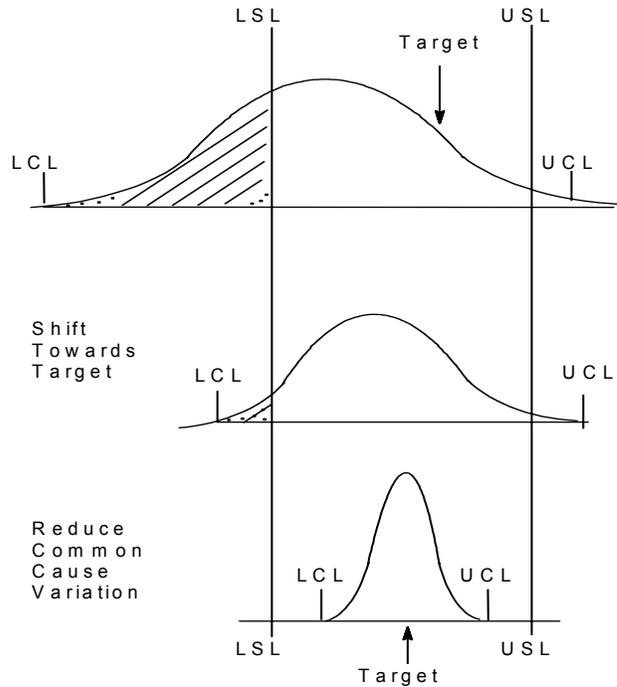


Figure 68. Making a Process Capable

Risk Management

Risk management involves the activities of defining, measuring, prioritizing, and managing risk in order to eliminate or minimize any potential negative effect associated with risk.

Defining Risk

Risk is the possibility that an unfavorable event will occur. It may be predictable or unpredictable. Risk has three components, each of which must be considered separately when determining how to manage the risk.

- The event that could occur – the risk
- The probability that the event will occur- the likelihood
- The impact or consequence of the event if it occurs – the penalty

Risks can be categorized as one of the following:

Technical such as complexity, requirement changes, unproven technology, etc.

Programmatic or Performance such as safety, skills, regulatory changes, material availability, etc.

Supportability or Environment such as people, equipment, reliability, maintainability, etc.

Cost such as sensitivity to technical risk, overhead, estimating errors, etc.

Schedule such as degree of concurrency, number of critical path items, sensitivity to cost, etc.

Characterizing Risk

Risk has five distinguishing characteristics:

Situational

Changes in a situation can result in new risks. Examples include, replacing a team member, undergoing reorganization, or changing a project's scope.

Time-Based

Considering a software development life cycle, the probability of risk occurring at the beginning of the project is very high (due to the unknowns), whereas at the end of the project the probability is very low. In contrast, during the life cycle, the impact (cost) from a risky event occurring is low at the beginning (since not much time and effort have been invested) and higher at the end (as there is more to lose).

Interdependent

Within a project, many tasks and deliverables are intertwined. If one deliverable takes longer to create than expected, other items depending on that deliverable may be affected, and the result could be a domino effect.

Magnitude Dependent

The relationship of probability and impact are not linear, and the magnitude of the risk typically makes a difference. For example, consider the risk of spending \$1 for a 50/50 chance to win \$5, vs. the risk of spending \$1,000 for a 50/50 chance of winning \$5,000 vs. the risk of spending \$100,000 for a 50/50 chance of winning \$500,000. In this example, the probability of loss is all the same (50%) yet the opportunity cost of losing is much greater.

Value-Based

Risk may be affected by personal, corporate or cultural values. For example, completing a project on schedule may be dependent on the time of year and nationalities or religious beliefs of the work team. Projects being developed in international locations where multiple cultures are involved may have a higher risk than those done in one location with a similar work force.

Managing Risk

Risk management is the process used to identify, analyze, and respond to a risk. Identifying, analyzing, and prioritizing risks require knowledge of the business functions, and user involvement. The Project Management Institute's Project Management Body of Knowledge

(PMBOK) defines the following four processes to address risk management. The PMBOK also notes that different application areas may use different names for these four processes.

- Risk Identification
- Risk Quantification
- Risk Response Development
- Risk Response Control

This discussion of risk management addresses six processes, which have the following mapping to the PMBOK processes.

Risk Identification

Risk Identification – this process answers the question "What are the risks?"

Risk Quantification

Risk Analysis - this process answers the question "Which risks do we care about?"

Risk Prioritization - this process answers the question "How are the risks prioritized?"

Risk Response Development

Risk Response Planning - this process answers the question "What should be done about the risk?"

Risk Response Control

Risk Resolution – this process executes the plan that was developed in the prior step.

Risk Monitoring – this process evaluates the action taken, documents the risk results and repeats the cycle of identification, quantification and response.

Risk Identification

For any project, risks should be identified as early as possible in the development life cycle. Care should be taken to not limit the process of identification to internal risks within the project, but to consider possible external risks outside the project, such as management changes, new tools, company mergers, changing strategies, changing market trends, customer inputs, politics, etc. Internal risks can be controlled or influenced by the project team, whereas external risks are outside the control or influence of a project team. For example, major disaster threats, such as storms, fires, terrorism, power and telecom outages, etc., must also be considered and balanced against the costs of full preparedness.

The likelihood of a tornado demolishing a data center is very tiny. The impact, however, could be the loss of the business – perhaps billions of dollars, and even loss of human life. To further complicate the risk, the occurrence of a tornado today does not mean that there cannot be another one tomorrow. The risk potentials are independent, and to a large degree, unpreventable. This

means that organizations must be prepared for major disasters with very low probabilities, but with the possibility of them happening close together. The four 2004 hurricanes in Florida are examples of major disasters happening in close succession.

The risk management process should also address the frequency at which risks should be identified during a project. Typically this occurs not only at the start of a project, but also at regular intervals (e.g., decision point reviews) throughout.

Product and project documentation and historical information should be used as a source of input when considering a list of possible risks. Documentation to examine, includes requirement specifications, compliance matrices, project plans, project schedules, current performance data, contracts, reviews, checklists, and "lessons learned".

Project teams may also opt to use tools such as brainstorming, affinity diagrams, nominal group technique, or checklists to identify risks. These tools are discussed in Skill Category 4.

Risk Analysis

Risk analysis is a method to quantify risk. The objective of risk analysis is to help management strike an economic balance between the impact of risks and the cost of protective measures. Risks that can have a significant impact on the process, product, or organization need to be addressed in a decision-making process; while it may be possible to ignore insignificant risks (this is determined during the risk planning process).

Consider questions such as:

- How big is the risk?
- What exactly is being exposed to the risk?
- Can this be considered an acceptable risk?
- What alternatives are there?
- Will alternatives invoke additional risks?

As it is impossible to be completely certain of the impact or likelihood of many events, these events are estimated using a combination of historical data, knowledge of the event, and experience and judgment. Tools such as simulation, decision trees, or calculating a monetary value are used. Risk can be calculated by structured (associated with data) and unstructured (focus on judgment and experience) methods. Regardless of the method used, two elements must always be considered:

- The probability of such an event occurring
- The resulting impact if the event occurs

Risk rating depends on the individual analyst, and thus is subjective. For a given scenario, each person determining the risk may come up with a different result. This is because the analyst's personality and thought process are important factors. Some people will take more risks than

others, and as a result will view situations differently. Therefore, developing a simple rating system that has documented measurement criteria will help ensure consistency in rating risk.

As shown in Figure 69, a project with low probability and low impact events is a low risk project overall. A project with high probability and high impact events has a high risk overall. On the other hand, a project with only a few high impact events may be considered overall low risk, and a project with a few moderate events may be considered high risk overall. Making these determinations is where the subjectivity of the analyst factors in.

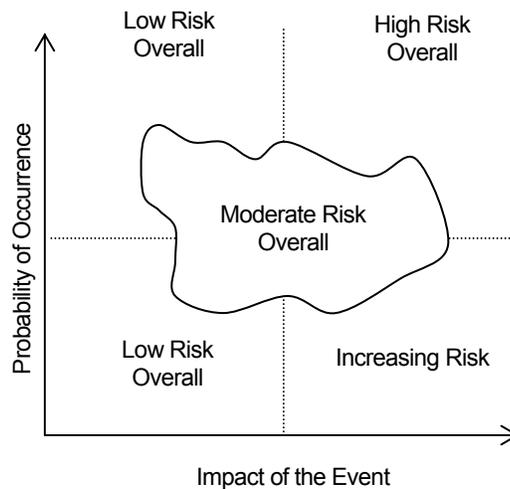


Figure 69. Risk of a Project's Overall Success

The probability of an event's occurrence can be determined:

- Using personal opinion or team consensus
- Using an historical database
- Converting approximations to numbers

The examples below show how different approximation methods can be converted.

Even (50%)

Probable or improbable or likely or unlikely (< 50% or > 50%)

Low (< 33.33%), medium (33.34 - 66.66%), high (66.67 - 100%)

Not likely (< 25%), possible (26 - 40%), likely (41 - 60%), very likely (61 - 75%), certain (76 - 100%)

The impact of an event is usually represented by a monetary value, and considered with respect to schedule, cost, and profitability.

Monetary Value

Monetary value is the best common denominator for quantifying the impact of an adverse circumstance – whether the damage is actual or abstract, whether the victim is a person, a piece of equipment, or a function. It is the recompense used by the courts to redress both physical damage and mental anguish.

Schedules

Schedules are examined to determine any slips in the completion date. One method of analyzing schedules is by looking at each task independently and then multiplying them together. For example, if a project contains 3 independent tasks and each task has a 50% chance of finishing on time, the project has a 12.5% chance of finishing on time (50% * 50% * 50%).

Costs

Costs are calculated over the product's life cycle. The costs for each phase are added together for a total life cycle cost. For example, when producing a software product the cost should reflect not only what it takes to develop the product, but also to fix and maintain it.

Profitability

Profitability is typically calculated using:

- Return-on-sales, which is profit, or return as a percentage of a project's total cost. It does not depend on time. A positive value indicates a profit and a negative value indicates a loss.
- Return-on-investment, which is an organization-wide measure that assesses performance against invested assets (organizations may use different formulas). It measures efficiency, and balances the asset use and the profit margin.
- Economic-value added, which evaluates the cost of capital percent vs. the return of capital percent. The cost of capital is the cost of financing the organization's operations. It takes into account the minimum rate of return that the investors (such as debt holders and shareholders) require.
- Internal rate-of-return, which is a relative measure based on the timing of cash inflow and outflow. It is the rate at which the net present values of cash inflow and outflow become equal.

Using a structured method, risk is calculated using the formula:

$$\text{Expected value} = \text{Probability} * \text{Impact}$$

Where:

- Expected value - is a dollar amount. Considering the best case (all good happens and no bad) and the worst case (all bad happens and no good), the

actual value will most likely be between the best and worst case. The expected value is the estimate of where the actual value is expected to be.

- Probability - is the likelihood that the event will occur.
- Impact - is the gain or loss that is incurred if the event occurs.

A simple, unstructured estimation scheme is to use high, medium, and low categories. The parameters are determined by the organization; for example, a frequency of 1 to 10 may be considered low; 11 to 100, medium; and more than 100, high. Once the categories have been established, they must be used for every risk situation. When frequency and loss are categorized as high (3), medium (2), and low (1), as shown in Table 21, the risk score will be in the range of 2 through 6.

Table 21. Estimating Risk Ratings

Frequency Rating	Impact	Risk
High	High	6
High	Medium	5
Medium	High	5
High	Low	4
Low	High	4
Medium	Medium	4
Medium	Low	3
Low	Medium	3
Low	Low	2

Risk Prioritization

In the prioritization process, risks from the analysis process are ranked from highest to lowest. When possible, they should be ranked quantitatively; otherwise, it is done qualitatively. Items that have similar ranks may need to be ranked separately.

If the high-medium-low method was used to calculate risk, the analyst would need to determine how to rank the two scores of 5, the three scores of 4 and the two scores of 3. In other words, determine whether the value of the frequency or the value of the impact would be more important.

Risks may also be prioritized using a question-and-answer technique to filter out unimportant risks. Four filters are typically used: impact (significant or insignificant), likelihood of occurrence (very likely or not likely), time frame (short-term or long-term), and program control (within control or not within control).

Other factors that may affect how risks are prioritized include the capability or capacity within the organization to do something about the risk, and how much the organization buys into the risk.

Risk Response Planning

In this planning process, a response or strategy is developed for each item in the prioritized risk listing. It may include a primary choice and a backup option.

Responses to risk can be categorized in one of three ways:

- Accepting the consequences if the event occurs
 - Active acceptance would involve developing a contingency plan that would be executed if the risk occurs
 - Passive acceptance would allow the risk event to occur (e.g., making less money if the project is a few weeks late)
- Avoiding the consequences by eliminating possibility of the event occurring
- Mitigating the risk (reducing its expected value) by:
 - Minimizing the probability of occurrence
 - Minimizing the value of the impact
 - Deflecting (or transferring) the risk elsewhere (in a software project this might mean passing the risk onto a subcontractor or to the customer).

Typical responses for risk include: procurement, contingency planning, alternative strategies, and insurance. With procurement, products or services are acquired from outside the project to help mitigate the risk (e.g., hire a consultant who has experience with the new technology being used). As noted above, a contingency plan indicates active acceptance of the risk. Developing alternative strategies is one possible means of avoiding a risk (e.g., a different approach to development may eliminate a risk). Bonding is an example of insurance, which minimizes the impact of a risk.

Once the strategies have been determined, they should be documented in a risk management plan or as part of the project plan.

Risk Resolution

This process implements the planned strategy during the execution of the project if the risk event occurs. A key factor in this process is communication. It is important that the plan be communicated to the project team and other relevant people, and that responsibilities related to the strategy be clear. As part of the communication aspect, the project leader should ensure that all events that occur are documented along with the actions taken.

In addition to the plan, workarounds should be considered in case risks occur that were not planned for, there was an unplanned response to a negative risk event, or there was a crisis.

Risk Monitoring

Risk monitoring includes periodically assessing project status, reassessing the documented risks, examining executed strategies that succeeded or failed, and considering new risks. Questions to consider include the following:

- Have events that occurred affected the status of the project?
- Is the event still possible?
- Have the probability and impact of the event changed?
- Is the tolerance within the project team/organization the same?
- Have there been any changes to the customer base, the technology being used within the organization, or related to resources that would result in new risks?

If an event occurs, it should be a trigger to cycle through the processes of identification, analysis, prioritization and response.

Part of the monitoring process includes documenting risk results in a risk management or project plan (in some literature, this documentation aspect is shown as a separate process). Lessons learned from successes and failures may be documented in a separate "lessons learned" document or database. Documentation should be updated continuously since it can be used for historical purposes, to summarize lessons-learned, and as a communication vehicle. It is also critical that the documentation be disseminated.

Software Risk Management

Within many software development organizations risk management remains ad hoc and incomplete. Where risk management processes exist, they tend to be used only for large projects, or those perceived to be risky.

Incorporating risk management into the software development life cycle includes planning at the following levels:

Long-Term or High-Level

This level includes both long range planning, and optimizing the organization's mix of projects in order to obtain a balance.

Medium-Term or Medium-Level

This level deals with project management strategies, project evaluation, and selection and project portfolio management

Short-Term or Low-Level

This level includes development, integration and testing strategies.

There are several components to consider when incorporating risk management into software project management:

- In a traditional waterfall methodology, most risk management activity occurs close to milestones. In a spiral development model the risk management activity falls in the explicit risk management portion.
- Risk management is not a separate, independent auditing process. It is a part of a project manager's job that should be explicitly performed, either quantitatively or qualitatively. Large projects should follow a formal risk management process; smaller projects may require less effort. Ultimately the project manager must decide based on the project's cost, schedule and performance issues.
- Risk management should be implemented as a series of integral tasks that are inserted into the normal project planning process, not added on to the end of the planning activities.
- The customer, management, development team, and others should be involved with determining project risks. Risks should not be the sole responsibility of the project manager.

Risks of Integrating New Technology

One of the major challenges facing an IT organization is to effectively integrate new technology. This integration needs to be done without compromising quality.

The QA analyst has three roles in integrating new technology:

Determining the Risks

Each new technology poses new risks. These risks need to be identified and prioritized and, if possible, quantified. Although the QA analyst will probably not perform the actual task, the QA analyst needs to ensure that a risk analysis for the new technology is undertaken and effectively performed.

Assuring that the Controls are Adequate to Reduce the Risk

The QA analyst needs to assess whether the controls proposed for the new technology are adequate to reduce the risk to an acceptable level. This may be done by line management and reviewed by the QA analyst.

Assuring that Existing Processes are Appropriately Modified to Incorporate the Use of the New Technology

Work processes that will utilize new technologies normally need to be modified to incorporate those technologies into the step-by-step work procedures. This may be done by the workers responsible for the work processes, but at least needs to be assessed or reviewed by the QA analyst.

Implementing a Measurement Program

The key to a good measurement program is knowing what results are wanted, and what drives those results. Then metrics need to be developed to measure those results and drivers. This section explains how an effective measurement program is implemented.

The Need for Measurement

Current IT management is often ineffective because the IT function is extremely complex, and has few well-defined, reliable process or product measures to guide and evaluate results. Thus, accurate and effective estimating, planning, and control are nearly impossible to achieve. Projects are often characterized by:

- Schedule and cost estimates that are grossly inaccurate
- Poor quality software
- A productivity rate that is increasing more slowly than the demand for software
- Customer dissatisfaction

Addressing these problems requires more accurate schedule and cost estimates, better-quality products, and higher productivity that can be achieved through improved software management. Improvement of the management process depends upon improved ability to identify, measure, and control essential parameters of the IT processes. Measurement is an algorithm connecting the desired result (i.e., the effect wanted) with the contributors or causes that will enable that effect to be achieved. The results are what management wants, and the contributors are attributes of the processes that will be used to achieve those results. By measuring processes and products, information is obtained that helps control schedule, cost, quality, productivity, and customer satisfaction. Consistent measurements provide data for the following:

- Expressing requirements, objectives, and acceptance criteria in a quantitative manner
- Monitoring progress of a project and/or product
- Making tradeoffs in the allocation of resources
- Evaluating process and product quality
- Anticipating problems
- Predicting deadlines of current project
- Estimating future projects of a similar nature

Results indicate that implementation and application of a measurement program can help achieve better management results, both in the short run (for a given project) and in the long run (improving productivity on future projects).

Prerequisites

Implementing a measurement program requires four prerequisite steps:

1. Perceive the need for a measurement program and make a commitment to it.

The lack of timely and usable quantitative information to solve major project problems becomes apparent at the senior management level. Seeing the need for better management information (as discussed in the prior section), the site manager, general manager, or division VP sponsors an organizational commitment to a measurement program. As a senior manager, the sponsor has the authority to ensure understanding at all levels in the organization.

2. Identify a champion and change agent, and assign organizational responsibility.

A champion is an advocate for the program by virtue of his/her technical credibility and influence. Champions are managers at the senior, middle, program, or project level, and are assisted by a change agent.

A change agent is a management leader empowered by the sponsor and champions to plan and implement the program. Change agents are most effective when they are members of working groups that will benefit from the measurement program. They have the operational knowledge needed to schedule, monitor, control, and report the accomplishments of the measurement program.

The project or organization selected for the implementation of the measurement program should have the resources, authority, and responsibility to make the program happen. Unless a very limited program is contemplated, responsibility for implementing the program should not be given to a single individual.

During this step, idea generators, idea exploiters, and information gatekeepers should be identified. Idea generators contribute new ideas about the measurement program. Idea exploiters implement the new ideas in the form of pragmatic programs. Information gatekeepers are experts in measurement, and can provide informed realities of it. These people implement the ideas to form a workable measurement program and ensure developers accept the program.

3. Establish tangible objectives and meaningful measurement program activities.

The change agent guides the planning of the program, including the creation of program objectives and the design of program activities. The planning takes the sponsor's goals for more effective information and defines expected results, needed resources, tasks, and organizations responsible for implementing the program.

4. Facilitate management buy-in at all levels for the measurement program.

The measurement program's sponsor must clearly inform all levels of management of his/her interest in the measurement program and motivate their cooperation. They need to know the implementation team's goals, responsibilities, authority, and interfaces with other

organizations. Also important is to work with affected managers to obtain their buy-in, by tailoring the implementation so that most of their needs are met.

For each of the arguments against measurement that might be raised, there is a counter argument as to its importance.

- Measurement has a high cost; too much investment is required and the return is too low.

Actual experience with measurement suggests that recurring costs of 2 - 3% of direct project costs are adequate for data collection and analysis and for project tracking and monitoring. This small price buys real help in meeting project goals, and in increasing project control through better budgeting, problem anticipation, risk reduction, and incremental process improvement.

- All the data exists to support special studies for the senior management.

Data in many forms is typically scattered throughout an organization, but the data may not be organized, available, or accessible on a timely basis. All levels of management need measurement data in a meaningful form. Lower levels of management may need more detailed, quantitative technical information, but all levels need the information that the measurement function can provide.

- The ability to measure exists if and when it is needed.

Many organizations have the ability to measure their performance, but they only do it when a problem is apparent. At that point, appropriate information, if it exists at all, may not be available in time to solve the problem. System measurement, if practiced in a systematic manner, ensures that information is available at all times, for all projects, over all levels of management, when needed for problem solving and decision-making.

- Our estimates are based on standard industry methods, and our budgeting and planning is sufficient.

To be good enough, estimates, estimating algorithms, metrics, and experience data need to be tailored to an organization's unique environment and processes. Industry standard estimating algorithms, while useful, must have their parameter values calibrated to reflect the organization's unique environment; otherwise, they produce estimates that are not meaningful or reliable in that environment. Experience shows that controllability of system development projects decreases when budgets and the budgeting process bear little relation to the operating environment.

- If data is collected, the prime contractor may want to see it, take it away, or use it to harm your organization.

The customer has access to all customer contract data, and can require access to a central measurement database, including management data not collected as a part of the contract. The measurement database will contain information from past projects as well as ongoing projects. After a contract is satisfactorily completed, it is unlikely that the old data will be requested. Because this database will prove vital to the management of the business, it should be kept under a reasonable level of security.

- There is no use for a measurement program in the organization.

The bottom line is that if a business cannot be measured, it cannot be successfully managed for long without information. Reliable information about a business requires measurements.

Internal Control and Security

Two key issues for quality assurance are internal control and security. Interest in internal control has been highlighted by the passage of the Sarbanes-Oxley Act. Interest in internal control and security has been highlighted by publicized penetrations of security and the increased importance of information systems and the data contained by those systems.

The Sarbanes-Oxley Act, sometimes referred to as *SOX*, was passed in response to the numerous accounting scandals such as Enron and WorldCom. While much of the act relates to financial controls, there is a major section relating to internal controls. For Securities and Exchange Commission (SEC)-regulated corporations, both the CEO and the CFO must personally attest to the adequacy of their organization's system of internal control. Because misleading attestation statements is a criminal offense, top corporate executives take internal control as a very important topic. Many of those controls are incorporated into information systems, and thus the need for:

<i>Principles and Concepts of Internal Control</i>	371
<i>Risk and Internal Control Models</i>	386
<i>Building Internal Controls</i>	392
<i>Building Adequate Security</i>	395

Principles and Concepts of Internal Control

There are many definitions of internal control. Most of those definitions were developed by accountants. Some of those definitions focus more on financial controls, but others take a much broader view of internal control. Note that security is part of the system of internal control.

In the 1990's five major accounting organizations developed a framework for internal control. The five members of the group are: Financial Executives International, American Institute of Certified Public Accountants, American Accounting Association, The Institute of Internal Auditors, and the Institute of Management Accountants. This group is called the Committee of Sponsoring Organizations and is frequently referred to by the acronym *COSO*.

The COSO Internal Control Framework has been widely accepted after the passage of the Sarbanes-Oxley Act. This is because the Act requires organizations to have a "framework for internal control" and the SEC, which oversees the Sarbanes-Oxley Act, only recognizes the COSO Internal Control Framework.

Internal Control and Security Vocabulary and Concepts

There is no one generally accepted definition of internal control. Many have developed definitions, some broad, some very specific. However, it is important to have a clear definition of internal control.

The COSO report defines internal control as:

"...A process, effected by an organization's Board of Directors, management and other personnel, designed to provide reasonable assurance regarding the achievement of objectives in the following categories:

- Effectiveness and efficiency of operations
- Reliability of financial reporting
- Compliance with applicable laws and regulations."

The following four key terms are used extensively in internal control and security:

- Risk – The probability that an undesirable event will occur.
- Exposure – The amount of loss that might occur if an undesirable event occurs.
- Threat – A specific event that might cause an undesirable event to occur.
- Control – Anything that will reduce the impact of risk.

Let's look at an example of these terms using a homeowner's insurance policy. To that policy we will look at one risk, which is the risk of fire. The exposure associated with a risk of fire would be the value of your home. A threat that might cause that risk to turn into a loss might be an improper electrical connection or children playing with matches. Controls that would minimize the loss associated with risk would include such things as fire extinguishers, sprinkler systems, fire alarms and non-combustible material used in construction.

In looking at the same situation in information technology, we might look at the risk of someone penetrating a banking system and improperly transferring funds to the perpetrators personal account. The risk obviously is the loss of funds in the account, which was penetrated. The exposure is the amount of money in the account, or the amount of money that the bank allows to be transferred electronically. The threat is inadequate security systems, which allow the

perpetrator to penetrate the banking system. Controls can include passwords limiting access, limiting the amount that can be transferred at any one time, and unusual transactions such as transferring the monies to an overseas account, a control which limits who can transfer money from the account.

Internal Control Responsibilities

Everyone in an organization has some responsibility for internal control. Management, however, is responsible for an organization's internal control *system*. The chief executive officer is ultimately responsible for the internal control system. Financial and accounting officers are central to the way management exercises control. All management personnel play important roles and are accountable for controlling their units' activities.

Internal auditors contribute to the ongoing evaluation of the internal control system, but they do not have primary responsibility for establishing or maintaining it. The Board of Directors and its audit committee provide important oversight to the internal control system. A number of other parties, such as lawyers and external auditors, contribute to the achievement of the organization's objectives and provide information useful in improving internal control. However, they are not responsible for the effectiveness of, nor are they a part of, the organization's internal control system.

The Internal Auditor's Internal Control Responsibilities

Internal auditors directly examine internal controls and recommend improvements. The Institute of Internal Auditors, the professional association representing internal auditors worldwide, defines internal auditing as:

“... an independent, objective assurance and consulting activity designed to add value and improve an organization's operations. It helps an organization accomplish its objectives by bringing a systematic, disciplined approach to evaluate and improve the effectiveness of risk management, control, and governance processes.”

International Standards for the Professional Practice of Internal Auditing established by the Institute of Internal Auditors, specify that internal auditors should:

- Assist the organization by identifying and evaluating significant exposures to risk and contributing to the improvement of risk management and control systems.
- Monitor and evaluate the effectiveness of the organization's risk management system.
- Evaluate risk exposures relating to the organization's governance, operations, and information systems regarding the:
 - Reliability and integrity of financial and operational information
 - Effectiveness and efficiency of operations
 - Safeguarding of assets

Compliance with laws, regulations, and contracts

- Assist the organization in maintaining effective controls by evaluating their effectiveness and efficiency and by promoting continuous improvement.

All activities within an organization are potentially within the scope of the internal auditors' responsibility. In some entities, the internal audit function is heavily involved with controls over operations. For example, internal auditors may periodically monitor production quality, test the timeliness of shipments to customers or evaluate the efficiency of the plant layout. In other entities, the internal audit function may focus primarily on compliance or financial reporting-related activities.

The Institute of Internal Auditors standards also set forth the internal auditors' responsibility for the roles they may be assigned. Those standards, among other things, state that internal auditors should be independent of the activities they audit. They possess, or should possess, such independence through their position and authority within the organization and through recognition of their objectivity.

Organizational position and authority involve such matters as a reporting line to an individual who has sufficient authority to ensure appropriate audit coverage, consideration and response; selection and dismissal of the director of internal auditing only with Board of Directors or audit committee concurrence; internal auditor access to the Board or audit committee; and internal auditor authority to follow up on findings and recommendations.

Internal auditors are objective when not placed in a position of subordinating their judgment on audit matters to that of others. The primary protection for this objectivity is appropriate internal audit staff assignments. These assignments should be made to avoid potential and actual conflicts of interest and bias. Staff assignments should be rotated periodically and internal auditors should not assume operating responsibilities. Similarly, they should not be assigned to audit activities with which they were involved in connection with prior operating assignments.

It should be recognized that the internal audit function does not – as some people believe – have primary responsibility for establishing or maintaining the internal control system. That, as noted, is the responsibility of the CEO, along with key managers with designated responsibilities. The internal auditors play an important role in evaluating the effectiveness of control systems and thus contribute to the ongoing effectiveness of those systems.

Risk versus Control

From an academic perspective, the sole purpose of control is to reduce risk. Therefore, if there is no risk, there is no need for control. The formula for risk is as follows:

$$\text{Risk} = \text{Frequency} \times \text{Occurrence}$$

To calculate the loss due to risk, one must first determine:

- The frequency with which an unfavorable event will occur; and
- The probable loss associated with that unfavorable occurrence.

Let's look at a simple example. There is a risk that products shipped will not be invoiced. If we were to assume that an average of two products will be shipped per day and not be invoiced and the average billing per invoice is \$500, then the risk associated with not invoicing shipments is \$1,000 per day.

Management has chosen to use a positive concept in addressing risk, rather than a negative concept. In other words, they recognize that there will be a risk that products will be shipped but not invoiced. To address risk such as this, management has chosen to define control objectives rather than risks.

In our shipped but not billed risk example, management would define a control objective of "All products shipped should be invoiced". They would then implement controls to accomplish that positive control objective.

Environmental versus Transaction Processing Controls

Internal control systems have two components. It is important for the quality professional to know that there are two components of controls. The first is *environmental* (sometimes called general controls), and the second is the *transaction processing controls* within an individual business application.

Environmental or General Controls

Environmental controls are the means by which management uses to manage the organization. They include such things as organizational policies, the organizational structure in place to perform work, the method of hiring, training, supervising and evaluating personnel, and the processes provided personnel to perform their day-to-day work activities, such as a system development methodology for building software systems.

Auditors state that without strong environmental controls the transaction processing controls may not be effective. For example, if passwords needed to access computer systems are not adequately protected the password system will not work. Individuals will either protect or not protect their password based on environmental controls such as the attention management pays to password protection, the monitoring of the use of passwords that exist, and management's actions regarding individual workers failure to protect passwords.

Environmental controls are the means by which management uses to manage the organization. They include such things as:

- Organizational policies
- Organizational structure in place to perform work
- Method of hiring, training, supervising and evaluating personnel
- Processes provided to personnel to perform their day-to-day work activities, such as a system development methodology for building software systems.

Two examples of management controls are the review and approval of a new system and limiting computer room access.

Review and Approval of a New System

This control should be exercised to ensure management properly reviews and approves new IT systems and conversion plans. This review team examines requests for action, arrives at decisions, resolves conflicts, and monitors the development and implementation of system projects. It also oversees user performance to determine whether objectives and benefits agreed to at the beginning of a system development project are realized.

The team should establish guidelines for developing and implementing system projects and define appropriate documentation for management summaries. They should review procedures at important decision points in the development and implementation process.

Limiting Access to Computer Resources

Management controls involve limiting access to computer resources. It is necessary to segregate the functions of systems analysts, programmers, and computer operators. Systems analysts and programmers should not have physical access to the operating programs, and the computer files. Use of production files should be restricted to computer operating personnel. Such a restriction safeguards assets by making the manipulation of files and programs difficult. For example, assume a bank's programmer has programmed the demand deposit application for the bank. With his knowledge of the program, access to the files in the computer room on which information about the demand depositors is contained may allow him to manipulate the account balances of the bank's depositors (including his own balance if he is a depositor).

Transaction Processing Controls

The object of a system of internal control in a business application is to minimize business risks. Risks are the probability that some unfavorable event may occur during processing. Controls are the totality of means used to minimize those business risks.

There are two systems in every business application. As illustrated in Figure 70, the first is the system that processes business transactions, and the second is the system that controls the processing of business transactions. From the perspective of the system designer, these two are designed and implemented as one system. For example, edits that determine the validity of input are included in the part of the system in which transactions are entered. However, those edits are part of the system that controls the processing of business transactions.

Because these two systems are designed as a single system, most software quality analysts do not conceptualize the two systems. Adding to the difficulty is that the system documentation is not divided into the system that processes transactions and the system that controls the processing of transactions.

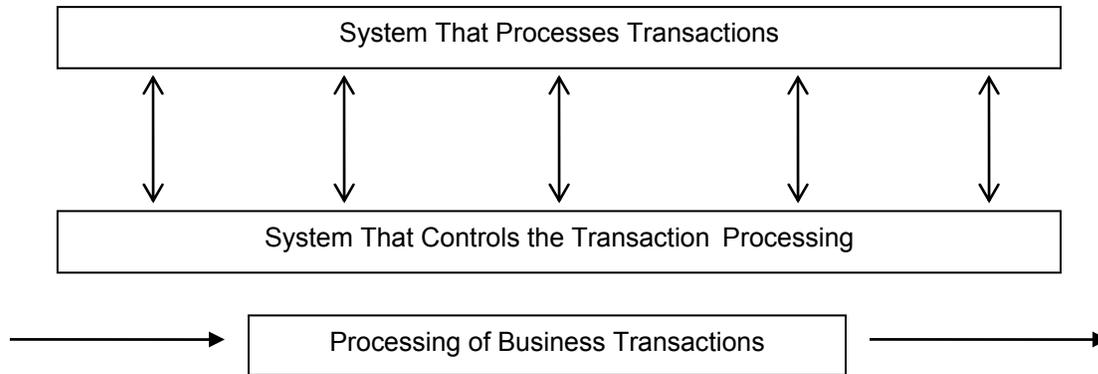


Figure 70. The Two Systems in Every Business Application

When one visualizes a single system, one has difficulty in visualizing the total system of internal control. For example, if one looks at edits of input data by themselves, it is difficult to see how the totality of control over the processing of a transaction is controlled. For example, there is a risk that invalid transactions will be processed. This risk occurs throughout the system and not just during the editing of data. When the system of internal controls is viewed it must address all of the risks of invalid processing from the point that a transaction is entered into the system to the point that the output deliverable is used for business purposes.

A point to keep in mind when designing transaction processing controls is that some input errors may be acceptable if they do not cause an interruption in the processing run. A simple example of this would be a misspelled description of an item. In deciding on controls, it is necessary to compare the cost of correcting an error to the consequences of accepting it. Such trade-offs must be determined for each application. Unfortunately there are no universal guidelines available.

It is important that the responsibility for control over transaction processing be separated as follows:

- Initiation and authorization of a transaction
- Recording of the transaction
- Custody of the resultant asset

In addition to safeguarding assets, this division of responsibilities provides for the efficiencies derived from specialization, makes possible a cross-check that promotes accuracy without duplication or wasted effort, and enhances the effectiveness of a management control system.

Preventive, Detective and Corrective Controls

This skill category describes three different categories of controls, preventive, detective, and corrective and provides examples of those types of controls. Also provided is a detailed process to follow when building controls within an information system.

The objectives of transaction processing controls are to prevent, detect, or correct incorrect processing. Preventive controls will stop incorrect processing from occurring; detective controls

identify incorrect processing; and corrective controls correct incorrect processing. Since the potential for errors is always assumed to exist, the objectives of transaction processing controls will be summarized in five positive statements:

- Assure that all authorized transactions are completely processed once and only once.
- Assure that transaction data is complete and accurate.
- Assure that transaction processing is correct and appropriate to the circumstances.
- Assure that processing results are utilized for the intended benefits.
- Assure that the application can continue to function.

In most instances controls can be related to multiple exposures. A single control can also fulfill multiple control objectives. For these reasons transaction processing controls have been classified according to whether they prevent, detect, or correct causes of exposure. The controls listed in the next sections are not meant to be exhaustive but, rather, representative of preventive, detective, and corrective controls.

Preventive Controls

Preventive controls act as a guide to help things happen as they should. This type of control is most desirable because it stops problems from occurring. Computer application systems designers should put their control emphasis on preventive controls. It is more economical and better for human relations to prevent a problem from occurring than to detect and correct the problem after it has occurred.

Preventive controls include standards, training, segregation of duties, authorization, forms design, pre-numbered forms, documentation, passwords, consistency of operations, etc.

One question that may be raised is, “At what point in the processing flow is it most desirable to exercise computer data edits?” The answer to this question is simply, “As soon as possible, in order to uncover problems early and avoid unnecessary computer processing.” Some input controls depend on access to master files and so must be timed to coincide with file availability. However, many input validation tests may be performed independently of the master files. Preferably, these tests should be performed in a separate edit run at the beginning of the computer processing. Normally, the input validation tests are included in programs to perform data-conversion operations such as transferring cards to tape. By including the tests in programs performing such operations, the controls may be employed without significantly increasing the computer run time.

Preventive controls are located throughout the entire IT system. Many of these controls are executed prior to the data entering the computer programs. The following preventive controls will be discussed in this chapter:

- Source-data authorization
- Data input

- Source-data preparation
- Turn-around documents
- Pre-numbered forms
- Input validation
- Computer updating of files
- Controls over processing

Source-Data Authorization

Once data has been recorded properly, there should be control techniques to ensure that the source data has been authorized. Typically, authorization should be given for source data such as credit terms, prices, discounts, commission rates, overtime hours, and so forth.

The input documents, where possible, should have evidence of authorization and should be reviewed by the internal control group in data processing. To the extent practical, the computer should be utilized as much as possible to authorize input. This may be done through programmed controls.

Data Input

Data input is the process of converting data in non-machine-readable form (such as hard-copy source documents) into a machine-readable form so that the computer can update files with the transactions. Since the data input process is typically a manual operation, control is needed to ensure that the data input has been performed accurately.

Source-Data Preparation

In many automated systems, conventional source documents are still used and, therefore, no new control problems are presented prior to the conversion of source documents into machine-readable form. Specially designed forms promote the accuracy of the initial recording of the data. A pre-audit of the source documents by knowledgeable personnel to detect misspellings, invalid codes, unreasonable amounts, and other improper data helps to promote the accuracy of input preparation.

In IT systems where the source document is eliminated or is in a form which does not permit human review, control over source-data preparation should be such that access to, and use of, the recording and transmitting equipment is properly controlled to exclude unauthorized or improper use.

Turn-Around Document

Other control techniques to promote the accuracy of input preparation include the use of turn-around documents which are designed to eliminate all or part of the data to be recorded at the source. A good example of a turn-around document is the bill which you may receive from an oil company. Normally the bill has two parts: one part is torn off and included with the remittance you send back to the oil company as payment for your bill; the other you keep for your records. The part you send back normally includes pre-recorded data for your account number and the

amount billed so that this returned part can be used as the input medium for computer processing of the cash receipts for the oil company.

Pre-Numbered Forms

Sequential numbering of the input transaction form with full accountability at the point of document origin is another traditional control technique. This can be done by using pre-numbered forms or by having the computer issue sequential numbers.

Input Validation

An important segment of input processing is the validation of the input itself. This is an extremely important process because it is really the last point in the input preparation where errors can be detected before files are updated. The primary control techniques used to validate the data are associated with the editing capabilities of the computer. Because of the characteristics of the computer, an IT system has unusual capabilities to examine or edit each element of information processed by it. This editing involves the ability to inspect and accept (or reject) transactions according to validity or reasonableness of quantities, amounts, codes, and other data contained in input records. The editing ability of the computer can be used to detect errors in input preparation that have not been detected by other control techniques discussed previously.

The editing ability of the computer is achieved by installing checks in the program of instructions, hence the term program checks. They include:

- Validity tests
- Completeness tests
- Logical tests
- Limit tests
- Self-checking digits
- Control totals

Validity tests are used to ensure that transactions contain valid transaction codes, valid characters, and valid field size. For example, in an accounts receivable system, if only input coded PB through PL were valid transaction codes, then input with other codes would be rejected by the computer. In a labor data collection system, all time transactions and job transactions could be checked by the computer against the random-access file of active job numbers, and non-matches indicated on a report to the shop foreman.

Completeness checks are made to ensure that the input has the prescribed amount of data in all data fields. For example, a particular payroll application requires that each new employee hired have two input cards in proper sequence and with all necessary information punched. A check may also be included to see that all characters in a field are either numeric or alphabetic.

Logical checks are used in transactions where various portions, or fields, of the record bear some logical relationship to one another. A computer program can check these logical relationships to reject combinations that are erroneous even though the individual values are acceptable.

Limit tests are used to test record fields to see whether certain predetermined limits have been exceeded. Generally, reasonable time, price, and volume conditions can be associated with a business event. For example, on one payroll application, the computer is programmed to reject all payroll rate changes greater than 15 percent of the old rate. The labor hours field is checked to see if the number of hours worked exceeds 44. In another application, an exception report is generated when a customer's receivable balance plus the total of his unfilled orders exceeds his credit limit.

Self-checking digits are used to ensure the accuracy of identification numbers such as account numbers. A check digit is determined by performing some arithmetic operation on the identification number itself. The arithmetic operation is formed in such a way that typical errors encountered in transcribing a number (such as transferring two digits) will be detected.

Computer Updating of Files

The updating phase of the processing cycle is the computer updating of files with the validated transactions. Normally computer updating involves sequencing transactions, comparing transaction records with master-file records, computations, and manipulating and reformatting data, for the purpose of updating master files and producing output data for distribution to user departments for subsequent computerized processing.

The accuracy of the file updating depends upon controls to ensure the programming, hardware checks designed and built into the equipment by the manufacturer, and programmed controls included in the computer programs themselves.

Another control technique for the proper updating of files is file maintenance. File maintenance consists of those procedures involved in making changes to the permanent-type information contained in master files, information such as name, address, employee number, and pay rate, for example, in a payroll file. Since this data is so important to the proper computerized processing of files, formalized procedures are required to make changes to this type of permanent information. All master file changes should be authorized in writing by the department initiating the change. A notice or register of all changes should be furnished to the initiating department to verify that the changes were made.

Controls over Processing

When we discussed input validation, we saw that programmed controls are a very important part of application control. Programmed controls in computer updating of files are also very important since they are designed to detect loss of data, check arithmetic computation, and ensure the proper posting of transactions. Let us examine some of these programmed controls.

Programmed checks to detect loss or non-processing of data are record counts, control totals and hash totals. A record count is the number of records processed by the computer. The resulting total can then be compared with a predetermined count. Normally a record count is established when the file is assembled, and the record count is carried as a control total at the end of the file or reel and is adjusted whenever records are added or deleted. For example, a record count may be established for all new hiring's or terminations processed. This record count can then be compared internally (if a control card is included with the input transactions) or manually to predetermined totals of new hiring's or terminations. Each time the file is processed, the records are recounted

and the quantity is balanced to the original or adjusted total. Although the record count is useful as a proof of processing accuracy, it is difficult to determine the cause of error if the counts are out of balance.

Three examples of processing controls are:

- A control total is made from amount or quantity fields in a group of records and is used to check against a control established in previous or subsequent manual or computer processing.
- A HASH total is another form of control total made from data in a non-quantity field (such as vendor number or customer number) in a group of records.
- Programmed checks of arithmetic calculations include limit checks, cross-footing balance checks, and overflow tests.

Some calculations produce illogical results such as million-dollar payroll checks or negative payroll checks. Such calculations can be highlighted in exception reports with the use of limit checks, which test the results of a calculation against predetermined limits. For example, a payroll system may include limit checks to exclude, from machine payroll check preparation, all employees with payroll amounts greater than \$500 or less than \$0.

Cross-footing balance checks can be programmed so that totals can be printed out and compared manually or totals can be compared internally during processing. For example, the computer-audit program (to be discussed in Chapter 8) is used in testing accounts receivable and in selecting accounts for confirmation. Each account is aged according to the following categories: current, 30, 60, and 90 days. The aged amounts for each account are temporarily stored in accumulators in the central processing unit. When all open items for the account have been aged, the aged totals for the account are compared to the account balance stored elsewhere in the central processing unit. Any difference results in an error indication. The program also includes for all accounts the accumulation and printout of aged amounts for manual comparison with the total accounts receivable balance.

The overflow test is a widely used test to determine whether the size of a result of a computation exceeds the registered size allocated to hold it. If so, there must be a means of saving the overflow portion of the results which would otherwise be lost. Overflow control may be programmed or may be available as a hardware or software control provided by the equipment manufacturer.

Programmed checks for proper postings may be classified as file checks. Basically, these are controls used to ensure that the correct files and records are processed together. The problem of using the correct file is a significant one in IT systems because of the absence of visible records and because of the ease with which wrong information can be written on magnetic tapes and disks. The increase in the size and complexity of modern data processing systems has resulted in the growth of large system libraries containing data that can cost thousands of dollars to generate. For the purpose of preserving the integrity of data, various labeling techniques have been devised to provide maximum protection for a file to prevent accidental destruction or erasure and to ensure proper posting, updating, and maintenance. Two types of labels are used, external and internal.

External labels are a physical safeguard which properly falls under the category of documentation and operating practices. They are attached to the exterior data processing media.

Detective Controls

Detective controls alert individuals involved in a process so that they are aware of a problem. Detective controls should bring potential problems to the attention of individuals so that action can be taken. One example of a detective control is a listing of all paychecks for individuals who worked over 80 hours in a week. Such a transaction may be correct, or it may be a systems error, or even fraud.

Detective controls will not prevent problems from occurring, but rather will point out a problem once it has occurred. Examples of detective controls are batch control documents, batch serial numbers, clearing accounts, labeling, and so forth.

The following detective controls will be discussed here:

- Data transmission
- Control register
- Control totals
- Documentation and testing
- Output Checks

Data Transmission

Once the source data has been prepared, properly authorized, and converted to machine-processable form, the data usually is transmitted from the source department to the data processing center. Data transmission can be made by conventional means (such as messenger and mail) or by data transmission devices which allow data transmission from remote locations on a much timelier basis.

One important control technique in data transmission is batching, the grouping of a large number of transactions into small groups. Batching typically is related more to sequential-processing systems where transactions have to be put into the same order as the master files; however, batching may also apply to many direct-access systems where it may be desirable to batch input for control purposes.

Let us examine a payroll example as an illustration of batching. In such an example, the source document may include time cards (source-data preparation) which should have been approved by a foreman (data authorization). For batching, these data time cards could be divided into groups of 25, with a control total for hours worked developed for each batch along with the total for all batches. Each batch transaction and its control totals could then be sent (data transmission) to the internal control group in the IT department for reconciliation with their batch control totals. Thus batching and control totals are useful techniques for the control of both data conversion and data transmission. These control totals could also be used during the computer-processing phase where the payroll files would be updated. We shall discuss this phase later in the chapter.

Control totals should be developed on important fields of data on each record to ensure that all records have been transmitted properly from the source to the data processing center. Controls might be developed on the number of records in each batch or could be based on some quantitative field of data such as invoice amount or hours worked, etc. Such controls serve as a check on the completeness of the transaction being processed and ensure that all transactions have been received in the data processing center.

Control Register

Another technique to ensure the transmission of data is the recording of control totals in a log so that the input processing control group can reconcile the input controls with any control totals generated in subsequent computer processing.

Control Totals

Control totals are normally obtained from batches of input data. These control totals are prepared manually, prior to processing, and then are incorporated as input to the computer-processing phase. The computer can be programmed to accumulate control totals internally and make a comparison with those provided as input. A message confirming the comparison should be printed out, even if the comparison did not disclose an error. These messages are then reviewed by the internal processing control group.

Documentation and Testing

Accuracy of programming is ensured by proper documentation and extensive program testing procedures. Good documentation will aid in locating programming errors and will facilitate correction even in the absence of the original designer or programmer. Extensive program test procedure under real-life conditions, testing all possible exceptions without actual programmer involvement, will minimize possibilities of hidden program bugs and facilitate smooth running of the system.

Output Checks

The output checks consist of procedures and control techniques to:

- Reconcile output data, particularly control totals, with previously established control totals developed in the input phase of the processing cycle.
- Review output data for reasonableness and proper format.
- Control input data rejected by the computer during processing and distribute the rejected data to appropriate personnel.
- Distribute output reports to user departments on a timely basis.

Proper input controls and file-updating controls should give a high degree of assurance that the computer output generated by the processing is correct. However, it is still useful to have certain output controls to achieve the control objectives associated with the processing cycle. Basically, the function of output controls is to determine that the processing does not include any unauthorized alterations by the computer operations section and that the data is substantially correct and reasonable. The most basic output control is the comparison of control totals on the final output with original input control totals such as record counts or financial totals. Systematic

sampling of individual items affords another output control. The testing can be done by the originating group or the control group.

One of the biggest controls in any system occurs when the originating group reviews reports and output data and takes corrective action. Review normally consists of a search for unusual or abnormal items. The programmed controls discussed above, coupled with exception reporting, actually enhance the ability of responsible personnel to take necessary corrective action.

Another form of output control in some organizations is the periodic and systematic review of reports and output data by internal audit staff. This group normally has the responsibility to evaluate operating activities of the company, including computer operations, to determine that internal policies and procedures are being followed.

Corrective Controls

Corrective controls assist individuals in the investigation and correction of causes of exposures that have been detected. These controls primarily collect evidence that can be utilized in determining why a particular problem has occurred. Corrective action is often a difficult and time-consuming process; however, it is important because it is the prime means of isolating system problems. Many systems improvements are initiated by individuals taking corrective actions on problems.

It should be noted that the corrective process itself is subject to error. Many major problems have occurred in organizations because corrective action was not taken on detected problems. Therefore detective control should be applied to corrective controls.

Examples of corrective controls are audit trails, discrepancy reports, error statistics, backup and recovery, etc. The following two corrective controls will be discussed in this chapter: error detection and resubmission, and audit trails.

Error Detection and Resubmission

Until now we have talked about data control techniques designed to screen the incoming data in order to reject any transactions that do not appear valid, reasonable, complete, etc. Once these errors have been detected, we need to establish specific control techniques to ensure that all corrections are made to the transactions in error and that these corrected transactions are reentered into the system. Such control techniques should include:

- Having the control group enter all data rejected from the processing cycle in an error log by marking off corrections in this log when these transactions are reentered; open items should be investigated periodically.
- Preparing an error input record or report explaining the reason for each rejected item. This error report should be returned to the source department for correction and resubmission. This means that the personnel in the originating or source department should have instructions on the handling of any errors that might occur.

- Submitting the corrected transactions through the same error detection and input validation process as the original transaction.

Audit Trails

Another important aspect of the processing cycle is the audit trail. The audit trail consists of documents, journals, ledgers, and worksheets that enable an interested party (e.g., the auditor) to trail an original transaction forward to a summarized total or from a summarized total backward to the original transaction. Only in this way can they determine whether the summary accurately reflects the business's transactions.

Cost versus Benefit of Controls

In information systems there is a cost associated with each control. No control should cost more than the potential errors it is established to detect, prevent, or correct, the cost of controls needs to be evaluated. To the extent that controls are poorly designed or excessive, they become burdensome and may not be used. This failure to use controls is a key element leading to major exposures.

Preventive controls are generally the lowest in cost. Detective controls usually require some moderate operating expense. On the other hand, corrective controls are almost always quite expensive. Prior to installing any control, some cost/benefit analysis should be made.

Controls need to be reviewed continually. This is a prime function of the auditor. The auditor should determine if controls are effective. As the result of such a review an auditor will recommend adding, eliminating, or modifying system controls.

The Quality Professionals Responsibility for Internal Control and Security

The quality professional is the organization's advocate for quality. The role of the quality professional involves identifying opportunities to improve quality and facilitating solutions. Because internal control and security are important responsibilities of management, the quality professional should be involved in those areas.

The quality professional should not be responsible for building or assessing the adequacy of internal control and security systems. However, the quality professional can assist in building work processes for building and assessing internal control and security. The quality professional can also evaluate the effectiveness and efficiency of those work processes.

The quality professional should support the importance of environmental controls in creating an environment conducive to effective internal control and security. The following section on control models will emphasize the importance of a strong control environment.

Risk and Internal Control Models

There are three generally accepted models for risk and internal control which are the COSO Enterprise Risk Management Model, the COSO Internal Control Model, and the CobiT Model.

COSO Enterprise Risk Management (ERM) Model

The ERM Process

In Fall 2001, the Committee of Sponsoring Organizations of the Treadway Commission (COSO) launched a landmark study designed to provide guidance in helping organizations manage risk. Despite an abundance of literature on the subject, COSO concluded there was a need for this study to design and build a framework and application guidance. Pricewaterhouse Coopers was engaged to lead this project.

The framework defines risk and enterprise risk management, and provides a foundational definition, conceptualizations, objectives categories, components, principles and other elements of a comprehensive risk management framework. It provides direction for companies and other organizations in determining how to enhance their risk management architectures, providing context for and facilitating application in the real world. This document is also designed to provide criteria for companies' use in determining whether their enterprise risk management is effective and, if not, what is needed to make it so.

Components of ERM

ERM consists of eight interrelated components. These are derived from the way management runs a business, and are integrated with the management process. The following components are illustrated in Figure 71:

- **Internal Environment**
Management sets a philosophy regarding risk and establishes a risk appetite. The internal environment sets the foundation for how risk and control are viewed and addressed by an organization's people.
- **Objective Setting**
Objectives must exist before management can identify events potentially affecting their achievement. ERM ensures that management has a process in place to set objectives and that the chosen objectives support and align with the organization's mission/vision and are consistent with the organization's risk appetite.
- **Event Identification**
Potential events that might have an impact on the organization must be identified. Event identification includes identifying factors – internal and external – that influence how potential events may affect strategy implementation and achievement of objectives.

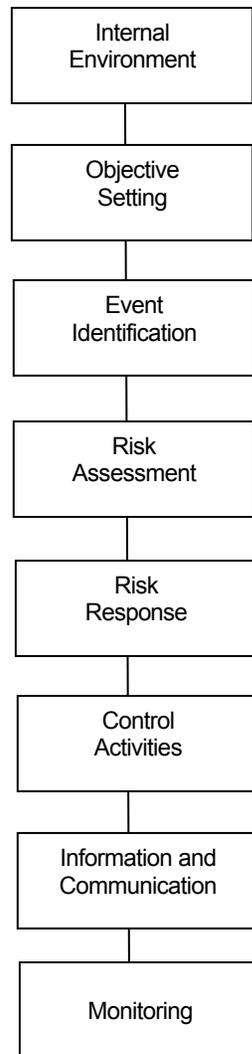


Figure 71. The Eight ERM Components

- Risk Assessment

Identified risks are analyzed in order to form a basis for determining how they should be managed. Risks are associated with related objectives that may be affected.

- Risk Response

Management selects an approach or set of actions to align assessed risks with the organization's risk appetite, in the context of the strategy and objectives.

- Control Activities

Policies and procedures are established and executed to help ensure that the risk responses management selected are effectively carried out.

- **Information and Communication**
Relevant information is identified, captured and communicated in a form and timeframe that enable people to carry out their responsibilities.
- **Monitoring**
The entire enterprise risk management process must be monitored, and modifications made as necessary.

COSO Internal Control Framework Model

In the COSO internal control framework, those developing the framework chose to use “control objectives” as opposed to defining risk. However, it is important to recognize that in accomplishing the control objectives, the control designers may have to go through a risk assessment process.

In understanding and using COSO to evaluate internal control, the internal auditor must evaluate whether controls are adequate to achieve the defined control objectives. Throughout the internal control framework only control objective will be defined. Even in the category that COSO defines as “risk”, positive control objectives will be stated rather than defining risks.

COSO uses the term “framework” to indicate an integrated system of internal controls. While the COSO framework defines specific control objectives, the framework also indicates that these control objectives are integrated vertically and horizontally.

Internal auditors are normally involved in auditing what COSO refers to as “control activity.” For example, payroll is a control activity. Within the payroll system there are procedures, which produce paychecks and the appropriate records, associated with payroll. In conjunction with this process is the system of controls that commences as transactions are initiated and concludes when those transactions are completed and incorporated into the appropriate organizational financial records. Thus, there are single controls and systems of controls.

COSO’s internal control framework consists of five interrelated components. These are derived from the way management runs a business, and are integrated with the management process. The components are:

- **Control Environment** – The core of any business is its people – their individual attributes, including integrity, ethical values and competence – and the environment in which they operate. They are the engine that drives the organization and the foundation on which everything rests.
- **Risk Assessment** – The organization must be aware of, and deal with, the risks it faces. It must set objectives, integrated with the sales, production, marketing, financial and other activities so that the organization is operating in concert. It also must establish mechanisms to identify, analyze and manage the related risks.
- **Control Activities** – Control policies and procedures must be established and executed to help ensure that the actions identified by management as necessary to

address risks to achievement of the organization's objectives are effectively carried out. The control activities component controls transaction processing.

- Information and Communication – Surrounding these activities are information and communication systems. These enable the organization's people to capture and exchange the information needed to conduct, manage and control its operations.
- Monitoring – The entire process must be monitored, and modifications made as necessary. In this way, the system can react dynamically, changing as conditions warrant.

These internal control components and their linkages are depicted in a model, presented in Figure 72. The model depicts the dynamics of internal control systems. Internal control is not a serial process, where one component affects only the next. It is a multidirectional interactive process in which almost any component can and will influence another.

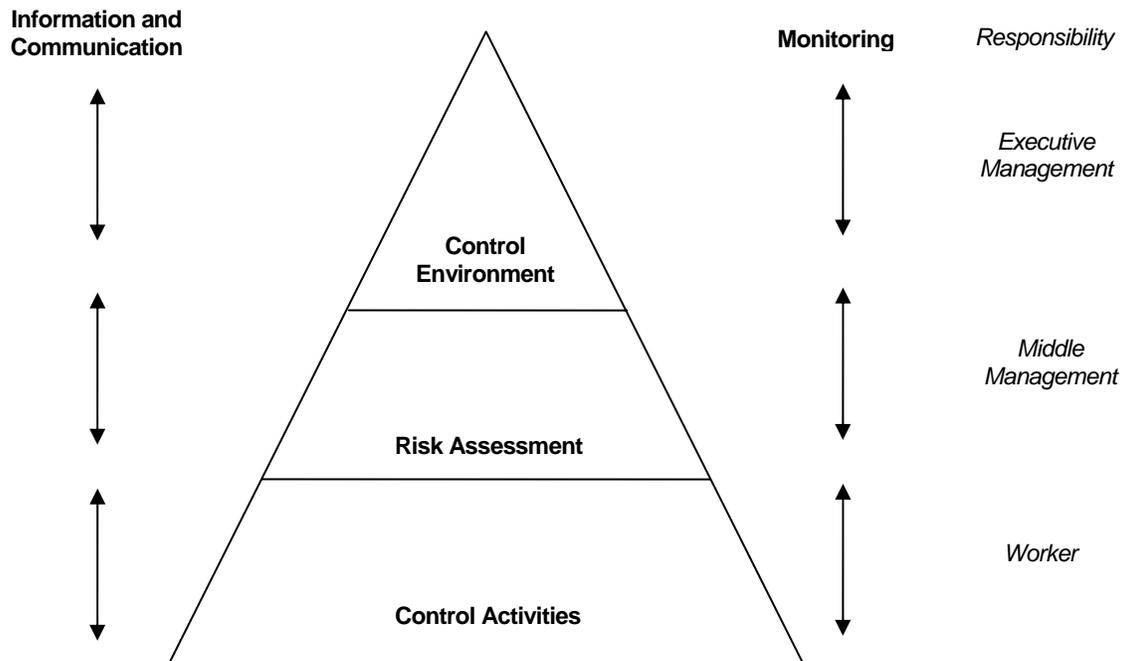


Figure 72. COSO Internal Control Framework Components

No two entities will, or should, have the same internal control system. Companies and their internal control needs differ dramatically by industry, size, culture, and management philosophy. Thus, while all entities need each of the components to maintain control over their activities, one company's internal control system often will look very different from another's.

Example of a Transaction Processing Internal Control System

Control objectives are defined to minimize risk. Many controls may be implemented to achieve a control objective. All the controls used to accomplish a control objective must be viewed as integrated, that is, a system of internal controls.

Figure 73 is an example of a cause/effect diagram. In viewing this diagram from an internal control perspective the effect is the achievement of a control objective. A previously discussed control example was “All products shipped are invoiced”. This is the effect that is wanted. Figure 73 lists four causes, which if effectively implemented, should achieve the control objective. These causes are that pre-numbered invoices will be used, pre-numbered shipping documents will be used, invoices will be prepared prior to shipment and invoices will be matched to shipping documents.

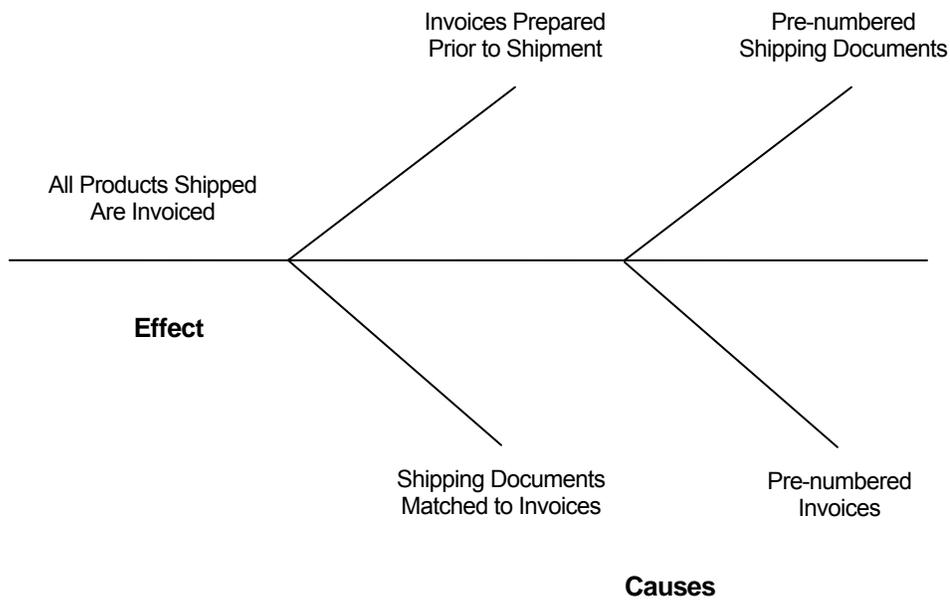


Figure 73. Cause and Effect Diagram Example

In understanding the COSO framework, and the interrelationship of control objectives, you may find it helpful to visualize the framework as a cause/effect diagram. The desired product of the COSO framework is the accomplishment of these three control objectives:

- Effective and efficient use of organization’s resources
- Preparation of reliable public financial statements
- Organization’s compliance to applicable laws and regulations

Using the COSO internal control framework to evaluate internal control is a two-step process as follows:

- Evaluate the organization's system of controls to assure that each control objective is achieved; and
- Assure that for all five components there is an effective integration into the organization's system of controls.

CobiT Model

CobiT is generally applicable and an accepted standard for IT security and control practices that provides a reference framework for management, users, and IS audit, control and security practitioners. CobiT enables an enterprise to implement effective governance over IT that is pervasive and intrinsic throughout the enterprise.

The CobiT Model is comprised of the following four-part cycle. The components of the four parts of the CobiT cycle can best be explained by listing the tasks within each component as follows:

- Part 1: Plan and Organize – The tasks in this part include:
Define strategic IT plan.
- Part 2: Acquire and Implement – The tasks in this part include:
Identify automated solutions.
- Part 3 – Deliver and Support
Defining and managing service levels, performance, problems and incidences.
- Part 4 – Monitor
Managing the processes and internal control practices.

Building Internal Controls

The system of internal control is designed to minimize risk. The control models emphasize the importance of the control environment. Normally quality assurance does not establish the control environment, but can review the control environmental practices in place in the IT organization using the COSO model for guidance. This section will focus on building transaction processing control in software systems.

Perform Risk Assessment

Building controls starts with risk assessment because reduction in risk is the requirement for a control. Risk assessment allows an organization to consider the extent to which potential events might have an impact on achievement of objectives. Management should assess events from two perspectives; first, the likelihood of an event occurring and second, the impact of that event. The assessment normally uses a combination of qualitative and quantitative methods.

The positive and negative impacts of potential events should be examined, individually or by category, across the organization. Potentially negative events are assessed on both an inherent and residual basis.

In risk assessment, management considers the mix of potential future events relevant to the organization and its activities. This entails examining factors – including organization size, complexity of operations and degree of regulation over its activities, that shape the organization's risk profile and influence the methodology it uses to assess risks.

The risk assessment component of Enterprise Risk Management or ERM is comprised of these sub-components:

- Inherent and Residual Risk

Management considers both inherent and residual risk. Inherent risk is the risk to an organization in the absence of any actions management might take to alter either the risk's likelihood or impact. Residual risk is the risk that remains after management responds to the risk.

- Estimating Likelihood on Impact

Likelihood represents the possibility that a given event will occur, while impact represents its affect. Estimating the likelihood and impact will determine how much attention the organization should give to a specific event.

- Qualitative and Quantitative Methodology and Techniques

Qualitative techniques such as categorizing an event into high, medium and low are used where risks do not lend themselves to quantification or when sufficient data is not available for quantification. Quantitative assessment techniques usually require a higher degree of effort and rigor.

- Correlation of Events

Management may assess how events correlate, where sequences of events combine and interact to create significantly different probabilities or impacts. While the impact of a single event might be slight, a sequence of events might have more significant impact.

Risk assessment is important because it is the process that enables management to determine both the likelihood and potential impact from the materialization of risk. Until the potential impact of an event is known, management may provide too much attention to an event or not enough attention.

Let's look at an example of the risk of customers leaving a Web site because it is too difficult to navigate. If few customers leave because of navigation problems and their purchase potential is minimal, no navigation improvements are needed. On the other hand, if there is a likelihood that many customers will leave with a potentially large loss of orders, resources should be allocated for navigation improvement.

Model for Building Transaction Processing Controls

System controls for computer applications involve automated and manual procedures. Automated procedures may include data entry performed in user areas, as well as the control of the data flow within a computer system. Manual procedures in user areas are developed to ensure that the transactions processed by IT are correctly prepared, authorized, and submitted to IT.

Manual application control procedures are also required within IT. For example, the IT input/output control section frequently balances and reconciles input to output. File retention and security procedures may be required and specified for individual computer applications. Such controls are unique to the requirements of the application and complement management controls that govern input/output controls and the media library.

Figure 74 shows the six steps of a transaction flow through a computer application system. Transaction flow is used as a basis for classifying transaction processing controls, because, based on field interviews, it provides a common framework for user and system development, personnel, and others interested in computer application systems control.

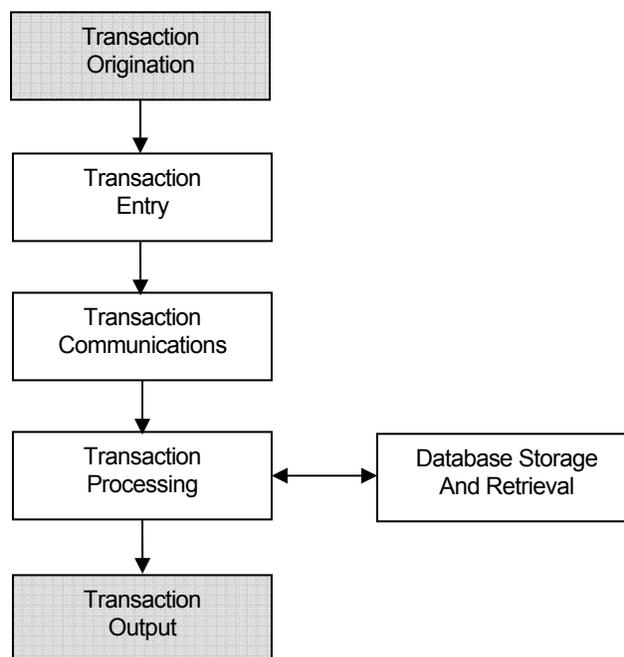


Figure 74. Model for Building Transaction Processing Controls

The two shaded boxes on the figure involve mostly the user organization. Each box is described below.

Transaction Origination

Transaction processing controls govern the origination, approval, and processing of source documents and the preparation of data processing input transactions and associated error detection and correction procedures.

Transaction Entry

Transaction processing controls govern the data entry via remote terminal or batch, data validation, transaction or batch proofing and balancing, error identification and reporting, and error correction and reentry.

Transaction Communications

Transaction processing controls govern the accuracy and completeness of data communications, including message accountability, data protection hardware and software, security and privacy, and error identification and reporting.

Transaction Processing

Transaction processing controls govern the accuracy and completeness of transaction processing, including the appropriateness of machine-generated transactions, validation against master files, and error identification and reporting.

Database Storage and Retrieval

Transaction processing controls govern the accuracy and completeness of database storage, data security and privacy, error handling, backup, recovery, and retention.

Transaction Output

Transaction processing controls govern the manual balancing and reconciling of input and output (within the input/output control section and at user locations), distribution of data processing output, control over negotiable documents (within data processing and user areas), and output data retention.

As a general rule, if risks are significant, controls should be strong. If the quality assurance analysts and/or the individual developing the adequacy opinion can match the risks with controls, the opinion can be based on that documentation.

Building Adequate Security

Security is heavily dependent on management establishing a strong control environment that encourages compliance to security practices. Good security practices, such as protecting passwords, will not be effective unless employees are diligent complying with password protection practices.

Security can be divided into two parts. First is the security management controls, and second is the security technical controls. Normally, security experts are needed to identify, install and monitor

the technical controls such as anti-virus software. Quality assurance should focus on the security management controls.

To build good security management controls quality assurance needs to build a security baseline. However, prior to building the baseline the team needs to understand the vulnerabilities that allow security penetrations. The next step is normally security awareness training. This section also identifies some of the more widely used security practices.

Where Vulnerabilities in Security Occur

Vulnerability is a weakness in an information system. It is the point at which software systems are easiest to penetrate. Understanding the vulnerabilities helps in designing security for information systems.

This section describes vulnerabilities that exist in the functional attributes of an information system. This section identifies the location of those vulnerabilities and distinguishes accidental from intentional losses.

Functional Vulnerabilities

The primary functional vulnerabilities result from weak or nonexistent controls in the following eight categories, listed in order of historic frequency of abuse:

1. Input/Output Data

The greatest vulnerability in this category occurs when access is most open. Data is subject to human interference both *before* it has been entered into a computer and *after* it has been output from the computer. Manual controls offer weaker resistance to people intent on interfering with data than do programs that must be manipulated to achieve unauthorized access. Input/output data controls include separation of data handling and conversion tasks, dual control of tasks, document counts, batch total checking, audit trails, protective storage, access restrictions, and labeling.

2. Physical Access

When physical access is the primary vulnerability, non-employees can gain access to computer facilities, and employees can gain access at unauthorized times and in unauthorized areas. Perpetrators' access motives may include political, competitive, and financial gain. Financial gain can accrue through burglary, larceny, and the unauthorized sale of computer services. In some cases, disgruntled employees pose a risk. Physical access controls include door locks, intrusion alarms, and physical-access line of sight, secure perimeter identification/establishment, badge systems, guard and automated monitoring functions (e.g., closed-circuit television), inspection of transported equipment and supplies, and staff sensitivity to intrusion. Violations often occur during nonworking hours when safeguards and staff are not present.

3. IT Operations

In this category of functional vulnerability, losses result from sabotage, espionage, sale of services and data extracted from computer systems, unauthorized use of facilities for personal advantage, and direct financial gain from negotiable instruments in IT areas. Controls in this category include separation of operational staff tasks, dual control over sensitive functions, staff accountability, accounting of resources and services, threat monitoring, close supervision of operating staff, sensitivity briefings of staff, documentation of operational procedures, backup capabilities and resources, and recovery and contingency plans. The most common abuse problem in this functional category is the unauthorized use or sale of services and data. The next most common problem is sabotage perpetrated by disgruntled IT staff.

4. Test Processes

A weakness or breakdown in a business test process can result in computer abuse perpetrated in the name of a business or government organization. The principal act is related more to corporate test processes or management decisions than to identifiable unauthorized acts of individuals using computers. These test processes and decisions result in deception, intimidation, unauthorized use of services or products, financial fraud, espionage, and sabotage in competitive situations. Controls include review of business test processes by company boards of directors or other senior-level management, audits, and effective regulatory and law enforcement.

5. Computer Programs

Computer programs are subject to abuse. They can also be used as tools in the perpetration of abuse and are subject to unauthorized changes to perpetrate abusive acts. The abuses from unauthorized changes are the most common. Controls include labeling programs to identify ownership, formal development methods (including testing and quality assurance), separation of programming responsibilities in large program developments, dual control over sensitive parts of programs, accountability of programmers for the programs they produce, safe storage of programs and documentation, audit comparisons of operational programs with master copies, formal update and maintenance procedures, and establishment of program ownership.

6. Operating System Access and Integrity

These abuses involve the use of time-sharing services. Frauds can occur as a result of discovering design weaknesses or by taking advantage of bugs or shortcuts introduced by programmers in the implementation of operating systems. The acts involve intentional searches for weaknesses in operating systems, unauthorized exploitation of weaknesses in operating systems, or the unauthorized exploitation of weaknesses discovered accidentally. Students committing vandalism, malicious mischief, or attempting to obtain free computer time have perpetrated most of the acts in university-run time-sharing services. Controls to eliminate weaknesses in operating system access include ensuring the integrity and security of the design of operating systems, imposing sufficient implementation methods and discipline, proving the integrity of implemented systems relative to complete and consistent specifications, and adopting rigorous maintenance procedures.

7. Impersonation

Unauthorized access to time-sharing services through impersonation can most easily be gained by obtaining secret passwords. Perpetrators learn passwords that are exposed accidentally through carelessness or administrative error, or they learn them by conning people into revealing their passwords or by guessing obvious combinations of characters and digits. It is suspected that this type of abuse is so common that few victims bother to report cases. Controls include effective password administration, periodic password changes, user protection of their passwords, policies that require hard-to-guess passwords, threat monitoring or password-use analysis in time-sharing systems and rules that forbid the printing/display of passwords.

8. Media

Theft and destruction of digital data are acts attributed to weaknesses in the control of magnetic/optical media. Many other cases, identified as operational procedure problems, involve the manipulation or copying of data. Controls include limited access to data libraries, safe storage of magnetic/optical media, data labeling, location controls, number accounting, controls of degasser equipment, and backup capabilities. (Most computer centers have a demagnetizing device for the purpose of erasing magnetic tapes.)

IT Areas Where Security is Penetrated

Data and report preparation areas and computer operations facilities with the highest concentration of manual functions are areas most vulnerable to having security penetrated. Nine primary functional locations are listed, described, and ranked according to vulnerability:

	Vulnerable areas	Rank
Functional Locations	Data and report preparation facilities	1
	Computer operations	2
	Non-IT areas	3
	Online terminal storage	4
	Programming offices	5
	Online data preparation and report generation	6
	Digital media storage facilities	7 (tie)
	Online operations	7 (tie)
	Central processors	9

1. Data and Report Preparation Facilities

Vulnerable areas include key-to-disk, computer job setup, output control and distribution, data collection, and data transportation. Input and output areas associated with online remote terminals are excluded here.

2. Computer Operations

All locations with computers in the immediate vicinity and rooms housing central computer systems are included in this category. Detached areas that contain peripheral equipment connected to computers by cable and computer hardware maintenance areas or offices are also included. Online remote terminals (connected by telephone circuits to computers) are excluded here.

3. Non-IT Areas

Security risks also derive from business decisions in such non-IT areas as management, marketing, sales, and business offices; and primary abusive acts may originate from these areas.

4. Online Terminal Systems

The vulnerable functional areas are within online systems, where acts occur by execution of programmed instructions as generated by terminal commands.

5. Programming Offices

This area includes office areas in which programmers produce and store program listings and documentation.

6. Online Data Preparation and Report Generation

This category includes the functions for preparing online scripts.

7. Digital Media Storage Facilities

This area includes data libraries and any storage place containing usable data.

7. Online Operations

This category is the equivalent of the computer operations discussed previously, but involves the online terminal areas. This category tied in ranking with number 7 above.

9. Central Processors

These functional areas are within computer systems themselves, and abusive acts may originate from within the computer operating system (not from terminals).

Accidental versus Intentional Losses

Errors generally during labor-intensive detailed work and errors lead to vulnerabilities. The errors are usually data errors, computer program errors (bugs), and damage to equipment or supplies. Such errors often require rerunning of jobs, error correction, and replacement/repair of equipment/supplies.

Nevertheless, it is often difficult to distinguish between accidental loss and intentional loss. In fact, some reported intentional loss is due to perpetrators discovering and making use of errors that result in their favor. When loss occurs, employees and managers tend to blame the computer hardware first (to absolve themselves from blame and to pass the problem along to the vendor to solve). The problem is rarely a hardware error, but proof of this is usually required before searching for elsewhere the cause. The next most common area of suspicion is users or the source of data generation because, again, the IT department can blame another organization. Blame is usually next placed on the computer programming staff. Finally, when all other targets of blame have been exonerated, IT employees suspect their own work.

It is not uncommon to see informal meetings between computer operators, programmers, maintenance engineers, and users arguing over who should start looking for the cause of a loss. The thought that the loss was intentional is remote because they generally assume they function in a benign environment.

In many computer centers, employees do not understand the significant difference between accidental loss from errors and intentionally caused losses. Organizations using computers have been fighting accidental loss for 40 years, since the beginning of automated data processing. Solutions are well known and usually well applied relative to the degree of motivation and cost-effectiveness of controls. They anticipate, however, that the same controls used in similar ways also have an effect on people engaged in intentional acts that result in losses. They frequently fail to understand that they are dealing with an intelligent enemy who is using every skill, experience, and access capability to solve the problem or reach a goal. This presents a different kind of vulnerability, one that is much more challenging and that requires adequate safeguards and controls not yet fully developed or realized, let alone adequately applied.

Establishing a Security Baseline

A baseline is a snapshot of the organization's security program at a certain time. The baseline is designed to answer two questions:

- What are we doing about computer security?
- How effective is our computer security program?

Baseline information should be collected by an independent assessment team; as much as possible, bias for or against a security program should be removed from the process. The process itself should measure both factual information about the program and the attitudes of the people involved in the program.

Creating Baselines

The establishment of a security baseline need not be time-consuming. The objective is to collect what is easy to collect, and ignore the information that is difficult to collect. In many instances, the needed information may be already available.

The three key aspects of collecting computer security baseline information are as follows:

- What to collect.
A determination must be made about what specific pieces of information would be helpful in analyzing the current security program and in building a more effective computer security program
- From whom will the information be collected?
Determining the source of information may be a more difficult task than determining what information should be collected. In some instances, the source will be current data collection mechanisms (if used by the organization). In other instances, individuals will be asked to provide information that has not previously been recorded.
- The precision of the information collected.
There is a tendency to want highly precise information, but in many instances it is not necessary. The desired precision should be both reasonable and economical. If people are being asked to identify past costs, high precision is unreasonable; and if the cost is large, it must be carefully weighed against the benefit of having highly precise information. In many instances, the same decisions would be made regardless of whether the precision was within plus or minus 1 percent, or within plus or minus 50 percent.

These six steps are commonly used to build a security baseline:

1. Establish baseline team.
2. Set baseline requirements and objectives.
3. Design baseline data collection methods.
4. Train baseline participants.
5. Collect baseline data.
6. Analyze and report computer security status.

The baseline procedures described in this chapter are general in nature. They do not take into account any unique features of an organization or the fact that data may already be available. Therefore, the six-step procedure may need to be customized to ensure that the correct information is collected at the least cost.

If customizing the six-step procedure, keep the following in mind:

- Availability of information.
If data is already available, it should not be collected in the baseline study. Those pieces of information can be excluded from the six-step process and incorporated into the process in Step 5 (data collection step).
- Need for information.

The baseline team must establish the objectives of the baseline study. The information collected should support these baseline objectives. If recommended information is not needed to support an objective, it should be deleted (and vice versa).

- Adjust for corporate language and nomenclature.

Generalized terms may have been used in baseline collection forms. If so, these should be adapted to organizational terminology wherever possible. Customized terminology provides the appearance of a baseline process designed specifically for the organization. The more closely people identify with the questions, the greater the reliability of the data collected.

The baseline is presented as a one-time data collection procedure. Nevertheless, the data collected must be updated periodically to measure changes from the baseline. This follow-up data collection should be integrated into the security program, and not separated as a special data collection process in the future.

All processes need to be continually updated as business conditions change. The proper time to change processes depends on the collection and analysis of feedback information. This feedback information is the same information that was collected in the baseline study. Without this continual feedback and analysis, even the most effective security programs will fall into disrepair.

Step 1: Establish the Team

The selection of the baseline team is a critical step in the baselining process. Team members must exhibit the following characteristics:

- Be representative of the groups involved in computer security
- Believe they are responsible for the performance of the baseline study
- Believe that the baseline study is a worthwhile exercise
- Be responsible for using the results of the baseline study to improve security

The baseline study must belong to the individuals responsible for computer security, and not to senior management. This does not mean that senior management does not participate in the baseline study or use the results of the study. It means that as much as possible the baseline will be owned by the people responsible for revising the computer security program.

This principle of ownership cannot be overemphasized. Most security programs fail because employees do not believe it is their program. They believe it is the program of the security officer, data processing management, senior management, or anybody but themselves. The entire emphasis in this book is that ownership and responsibility of the computer security belongs to all employees of an organization.

The concept of ownership begins with the selection of the computer security team. The recommended process for building the computer security team is relatively simple. Senior management convenes a group representative of all parties involved in computer security. In general, these should be the “movers and shakers” in the organization – the people who have the respect of the majority of the employees. These individuals may or may not be supervisors, but all must have the respect of the employees in the area from which they come.

Senior management then describes the importance and objectives of the baseline study. This presentation should be primarily a testimonial by senior management on the importance of computer security in the organization. The baseline study should be presented as a first step in establishing an effective security program for the organization. The presentation should not discuss the existing program, or identify or imply that the individuals associated with the current program have done less than an admirable job. The emphasis should be on changing requirements and the need to change and update the program.

Senior management should then ask for volunteers to work on the baseline study. Nobody should be appointed to this study group. If the individuals in the group do not believe in computer security, the results of the study will probably reflect that disbelief. On the other hand, if people volunteer, they must have an interest that can be nurtured by senior management to produce the kind of results desired. If senior management's testimonial is believable, there will be sufficient volunteers for the study group. If there are no volunteers, then senior management must seriously reconsider its attitudes and practices on computer security. In that instance, it may be better to hire a consultant to perform the study and then begin the new security program at the senior management level.

The desired size of the baseline group is a team of three to seven people. Three is the minimum to get an adequate synergistic effect between the team members, and a team with more than seven is difficult to manage. It is generally wise to have an odd number so that there will be no ties in the case of votes.

Step 2: Set Requirements and Objectives

The goal of the initial meeting of the baseline team should be to establish the requirements and objectives of the baseline study. To a large degree, these requirements and objectives will have been established by senior management when the study team was formed. Nevertheless, for the requirements and objectives to be "owned" by the study team, the team must adopt those requirements and objectives as their own and not as orders dictated by management.

The objectives should be twofold: first, to collect information about the computer security process; and second, to collect information about the effectiveness of that process in detecting and preventing penetration.

These two objectives must be converted into baseline requirements. The requirements should be defined in sufficient detail so that at the end of the baseline study it can be determined whether the baseline requirements have been met. It is these requirements that will motivate the remaining steps of the baseline process.

The baseline requirements must answer the why, what, who, when, where, and how of the baseline study; this information is then supplemented by the precision desired or achieved in the data collection process. The precision can be part of the requirements, or the respondents can be asked to state the level of precision they believe is in their responses. A Baseline Requirements Worksheet is illustrated in Figure 75.

Baseline Question	Security Process						Security Violations			
	Resource Protection	Resource Used for Security	Method	Training	Awareness	Support	Effectiveness	Penetration		Loss
								Prevented	Detected	
What										
Why										
Who										
Where										
Precision										
How										

Figure 75. Baseline Requirements Worksheet

Use this worksheet to record information regarding six focuses of concern, as follows:

- **Resource Protection**

Whenever possible, place a value on the resource that warrants security measures. The resources can be grouped; for example, the computer media library can be treated as one resource. It is not necessary to identify each reel of tape or each disk.

- **Resource used for security**

Evaluate the number of people and the amount of computer time, security equipment, and other expenditures made for the sole purpose of providing security over the resources.

- **Method**

Describe in detail the tools, techniques, and processes used to protect the identified resources so that they are readily understandable by management. For example, a single reference to guards is inadequate; the methods to be described should explain the number of guards and their specific responsibilities.

- **Training**

Define the programs used to train individuals in how to fulfill their security responsibilities, as well as the objectives, the methods used for training, and any procedures used to ensure that the necessary skills have been mastered.

- **Awareness**

Record employee perception regarding the need for security, management's security intent and specific responsibilities, and attitudes about the performance of security responsibilities.

- Support

The support received from supervision and peers in performing security responsibilities includes being given adequate time, training, direction, and assistance where necessary.

The security violations should be available through existing reporting systems; if not, information in the following four areas must be collected:

- Effectiveness

This involves the judgment of the individuals participating in the security program about the ability of the program to prevent and detect violations.

- Penetration prevented

Automated security systems log attempted violations, thus making this information easy to obtain. Manual security systems such as barriers may prevent a lot of people from penetrating, but unless the suspected offenders actually try and are observed, the information collected may not accurately reflect potential risk and efficacy of prevention.

- Penetration detected

This information should be recorded and reported by the individuals who detected the penetration (either through a formal reporting system or based on the recollection of those involved in the detection).

- Loss

These are the losses to the organization due to ineffective security systems. Computer security experts estimate that only about 10 percent of all losses are identified; therefore, this category might be divided into identified and estimated losses.

For all categories in the Baseline Requirements Worksheet, the baseline team should provide guidance as to the following:

- What should be collected
- Why the information is necessary
- Who has the information
- Where the information is available (For instance, the source of the information might not realize that he/she has the information within a database)
- Precision of information wanted
- How the information is to be provided

The more complete the information at this point in the data collection process, the easier the remaining steps are to execute.

Step 3: Design Data Collection Methods

Ideally, the collection of feedback information about computer security should come from the process itself. Feedback information should be a by-product of the process, and not an extra task given to the individuals who perform security-related activities. For example, the scheduling or conducting of training should generate the information about training; forms required to record security violations should provide the information about violations; and job-accounting systems should indicate the amount of resources utilized to provide computer security.

The baseline study recognizes that the appropriate feedback mechanisms are not yet in place. The performance of the study is usually a one-time task, and as such special data collection techniques need to be developed. Even so, these techniques should be commensurate with the value received from collecting the information. An easy-to-use data collection form as illustrated in Figure 76 is provided for the baseline study.

This Baseline Factual Data Collection form as shown in Figure 76 is designed to collect information that has a factual basis, as opposed to attitudinal information. This does not mean that the factual information is 100 percent accurate, but rather that it is factual in nature as opposed to an individual's opinion. However, both types of information are important because both affect the integrity of the security program.

Baseline Factual Data Collection Form	
Area of Security	_____ (Where) _____
Responsible Individuals	_____ (Who) _____
Resources Protected	_____ (Why) _____ _____ _____
Security Methods Used	_____ (What) _____
Training Provided on Security Methods	_____ (What) _____ _____
Cost to Develop	_____ (What) _____
Cost to Use	_____ (How) _____
Number of Penetrations during Last 12 Months (Why)	
Prevented	_____
Detected	_____
Losses	_____
Please Describe:	_____ (How) _____ _____ _____
Rate the accuracy of the above information: (Precision)	
_____ ± 5%	_____ ± 15%
_____ ± 25%	_____ ± 50%
_____ ± > 50%	
Additional Comments:	_____

Figure 76. Baseline Factual Data Collection Form

The factual information represents the areas of information defined in requirements (which are factual in nature). The information is collected according to areas of security. These can be organizational areas or areas of responsibility. The determination of what areas to survey will be based on an analysis of where computer security is conducted. At a minimum, it would include the following:

- Central computer site(s)
- Remote computer site(s)
- Storage areas for computer media and resources
- Points of physical protection
- Communication lines or networks
- Office equipment with security concerns
- Origination of security concerns

This data collection form provides general categories of information wanted. The prototype has been designed to assist baseline study teams in developing a customized form for their organization. The end of the form provides space for the respondent to indicate the precision of the information included on the form. This may need to be qualified to indicate which piece of information is being rated. As stated earlier, the precision can be given by the baseline study team or provided by the respondent. If the baseline study team indicates the desired level of precision, this last item should be omitted from the form.

People's attitudes about security are as important as the factual information gathered. People are the penetrations of security systems, and people are the guardians of the computer resources. If people are dedicated to computer security, it will happen; if they are not, the opportunity for penetration will increase.

Step 4: Train Participants

The individuals administering the baseline study, as well as the individuals involved in providing that data, should be trained in what is expected from them. At a minimum, this training should include the following:

- Baseline awareness
All of the individuals who will be participating in the program should be alerted to the fact that there is such a program in place and that they will be involved in it and are an important part of the success of the program.
- Collection methods
The participants in the program should be trained in the sources of security information, how to get it, and the attributes of information that are wanted. This primarily will be an explanation of the information recorded on the Baseline Requirements Worksheet as illustrated in Figure 76.
- Forms completion
If forms are distributed or used, the respondents to those forms should be instructed in the intent of the request for information, the type and extent of responses desired, and the type of precision needed.

The training is the responsibility of the chairman of the baseline study group. That individual should first train the study group, and then the study group should train the other participants.

Step 5: Collect Data

The collection process should be performed as quickly as possible. When the requests are made, include a response due date on the request (generally, within three to five working days). This provides enough time to work the request into a normal schedule, and yet does not give the appearance that the request is unimportant.

It is recommended, but not required, that the requests be made at a meeting of the participants. This is an opportunity to train the people in the forms and at the same time to make the request to have the data collected. It is the responsibility of the baseline study group to follow up and ensure that the needed data is provided by the prescribed date. In the event that the data is collected from computer files, the baseline team may either personally perform those analyses or contract them out to other parties.

Step 6: Analyze and Report Security Status

The analysis of the security self-assessment exercise resulted in a profile indicating the elements of security that presented the greatest threat. The analysis of the baseline will result in two more profiles:

- Factual information collected about security practices
- Attitudinal information from individuals involved in security

These analyses can be presented individually or as a single report. The approach selected should be based on experience as to how best to influence management decisions.

Let us consider a sample report of an analysis of the computer security baseline as illustrated in Figure 77. This analysis shows the individual areas and selected analyses based on both the factual and attitudinal data collection processes. The areas of security listed are general, such as the central computer site. In actual practice, these should be specific organizational units.

Area of Security		Totals	Central Computer Site(s)	Remote Computer Site(s)	Storage Area for Computer Media/Resources	Points for Physical Protection	Communication Lines/Networks	Office Equipment
Baseline Security Analysis								
Value of Computer Resources								
Cost to Install Security								
Cost to Operate Security								
Number of Penetrations	Prevented							
	Detected							
Value of Security Issues								
Effectiveness of Security								
Management Support for Security								

Figure 77. Analysis of the Computer Security Baseline Sample Report

The analyses selected show five factual areas (value of computer resources, cost to install security, cost to operate security, number of penetrations, and value of security losses) and two attitudinal areas, (effectiveness of security, and management support for security).

The translation from the data collection worksheets to this report may require some interpretation and further analysis. For example, the report suggests that a value be placed on the computer resources protected so that the cost of security can be shown in relation to what is being protected. If the individual reports cannot quantify this, the baseline team must do it. The attitudinal analyses, such as effectiveness of security, are taken from the attitudinal data collection worksheets. In our worksheet, we had a rating classification of one to six. These normally would have to be converted to terms that are more easily understandable. For example, if on the Likert scale the rating one represented ineffective security and six represented very effective security, the numbers might be converted to the following rating system:

- Likert scale values of 1 and 2 are rated “poor.”
- Likert scale values of 3 and 4 are rated “effective.”
- Likert scale values of 5 and 6 are rated “excellent.”

A report of this type obviously requires explanatory material (in writing or orally). The purpose of showing a recommended analysis is to suggest a way of presenting the information. It is expected that the baseline team will use its creativity in putting together a baseline report.

Using Baselines

The baseline is the starting point to a better security program. It reports the status of the current program and provides a basic standard against which improvements can be measured.

The baseline study serves two primary objectives. First, it reduces computer security discussions from opinion to fact. Even though some of the facts are based on attitude, they are a statistical base of data on which analyses and discussion can be focused, as opposed to people's opinion and prejudices. The baseline helps answer the question of whether the expenditure was worthwhile. For example, if a security software package is acquired, but there is no way to determine whether the environment has been improved, management will wonder whether that expenditure was worthwhile. When the next computer security request is made, the uncertainty about the last expenditure may eliminate the probability of a new improvement.

Security Awareness Training

IT organizations cannot protect the confidentiality, integrity, and availability of information in today's highly networked systems environment without ensuring that all the people involved in using and managing IT:

- Understand their roles and responsibilities related to the organizational mission
- Understand the organization's IT security policy, procedures, and practices
- Have at least adequate knowledge of the various management, operational, and technical controls required and available to protect the IT resources for which they are responsible
- Fulfill their security responsibilities.

As cited in audit reports, periodicals, and conference presentations, it is generally agreed by the IT security professional community that people are the weakest link in attempts to secure systems and networks.

The "people factor" – not technology – is the key to providing an adequate and appropriate level of security. If people are the key, but are also a weak link, more and better attention must be paid to this "component of security". A robust and enterprise-wide awareness and training program is paramount to ensuring that people understand their IT security responsibilities, organizational policies, and how to properly use them to protect the IT resources entrusted to them.

This practice provides a strategy for building and maintaining a comprehensive awareness and training program, as part of an organization's IT security program. The strategy is presented in a life cycle approach, ranging from designing, developing, and implementing an awareness and training program, through post-implementation evaluation of the program. The document includes guidance on how IT security professionals can identify awareness and training needs, develop a training plan, and get organizational buy-in for the funding of awareness and training program efforts.

While there is no one best way to develop a security awareness program, the process that follows is an all inclusive process of the best security awareness training program. This example includes these three steps:

1. IT management creates a security awareness policy.
2. Develop the strategy that will be used to implement that policy. (Note that this practice focuses on that strategy. Other practices will explain how to implement the steps in that strategy.)
3. Assign the roles for security and awareness to the appropriate individuals.

Step 1 – Create a Security Awareness Policy

The CIO and/or the IT Director need to establish a security awareness policy. The policy needs to state management's intension regarding security awareness. Experience has shown that unless senior management actively supports security awareness, there will be a lack of emphasis on security among the staff involved in using information technology and information.

Management support for security awareness begins with the development and distribution of a security awareness policy. Once that policy has been established, management makes security awareness happen through supporting the development of a strategy and tactics for security awareness, appropriately funding those activities, and then becoming personally involved in ensuring the staff knows of management's support for security awareness.

A security awareness policy can be as simple as ensuring that all stakeholders involved in the use of information technology and the information controlled by that technology, be made aware of their role and responsibility in assuring the security over that technology and information. Generally that policy would be clarified and expanded with a statement such as:

“Ensure that all individuals are appropriately trained in how to fulfill their security responsibilities before allowing them access to the system. Such training shall ensure that employees are versed in the rules of the system and apprise them about available technical assistance and technical security products and techniques. Behavior consistent with the rules of the system and periodic refresher training shall be required for continued access to the system. Before allowing individuals access to the application, ensure that all individuals receive specialized training focused on their responsibilities and the application rules. This may be in addition to the training required for access to a system. Such training may vary from a notification at the time of access (e.g., for members of the public usage of an information retrieval application) to formal training (e.g., for an employee that works with a high-risk application).”

“Ensure that the organization has trained personnel sufficient to assist the agency in complying with these requirements and related policies, procedures, standards, and guidelines. Delegate to the agency CIO the authority to ensure compliance with the responsibilities for information security. The required agency-wide information security program shall include security awareness training to inform personnel, including contractors and other users of information systems that support the

operations and assets of the agency, or information security risks associated with their activities.”

Step 2 – Develop a Security Awareness Strategy

A successful IT security program consists of: 1) developing an IT security policy that reflects business needs tempered by known risks; 2) informing users of their IT security responsibilities, as documented in the security policy and procedures; and 3) establishing processes for monitoring and reviewing the program.

Security awareness and training should be focused on the organization’s entire user population. Management should set the example for proper IT security behavior within an organization. An awareness program should begin with an effort that can be deployed and implemented in various ways and is aimed at all levels of the organization including senior and executive managers. The effectiveness of this effort will usually determine the effectiveness of the awareness and training program. This is also true for a successful IT security program.

An effective IT security awareness and training program explains proper rules of behavior for the use of an organization’s IT systems and information. The program communicates IT security policies and procedures that need to be followed. This must precede and lay the basis for any sanctions imposed due to noncompliance. Users first should be informed of the expectations. Accountability must be derived from a fully informed, well-trained, and aware workforce.

This step describes the relationship between awareness, training, and education – the awareness-training-education continuum. An effective IT security awareness and training program can succeed only if the material used in the program is firmly based on the IT security awareness policy and IT issue-specific policies. If policies are written clearly and concisely, then the awareness and training material – based on the policies – will be built on a firm foundation. The continuum is mentioned here and shown in Figure 78 to show the conceptual relationship between awareness, training, and education. For the purpose of this practice, clear boundaries are established between the three methods of learning.

Learning is a continuum; it starts with awareness, builds to training, and evolves into education. The continuum is illustrated in Figure 78.

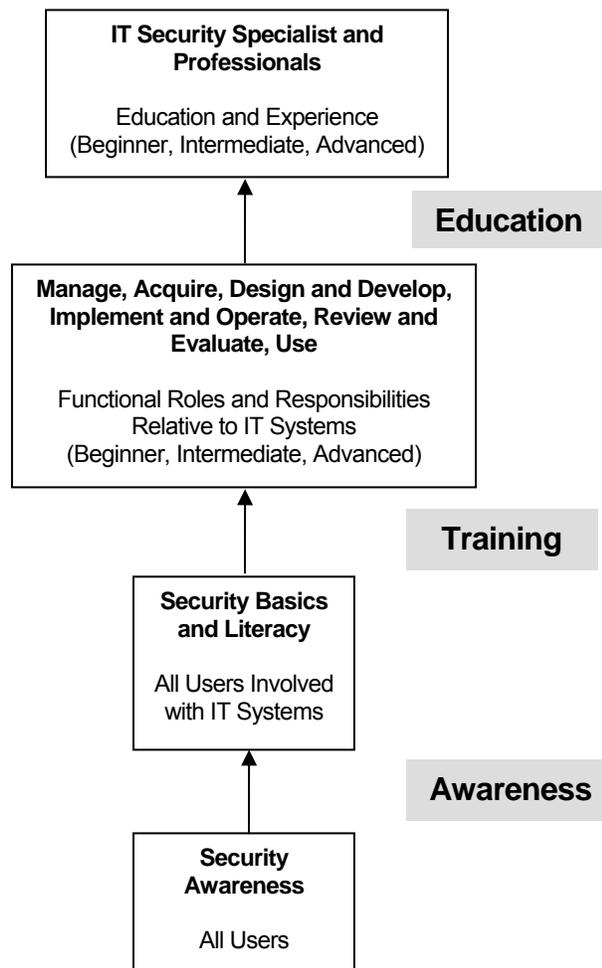


Figure 78. The IT Security Learning Continuum

Awareness

Security awareness efforts are designed to change behavior or reinforce good security practices.

Awareness is not training. The purpose of awareness presentations is simply to focus attention on security. Awareness presentations are intended to allow individuals to recognize IT security concerns and respond accordingly. In awareness activities, the learner is the recipient of information, whereas the learner in a training environment has a more active role. Awareness relies on reaching broad audiences with attractive packaging techniques. Training is more formal, having a goal of building knowledge and skills to facilitate the job performance.

An example topic for an awareness session (or awareness material to be distributed) is virus protection. The subject can simply and briefly be addressed by describing what a virus is, what can happen if a virus infects a user's system, what the user should do to protect the system, and what the user should do if a virus is discovered.

Training

Training strives to produce relevant and needed security skills and competencies. Training is defined as follows: The ‘training’ level of the learning continuum strives to produce relevant and needed security skills and competencies by practitioners of functional specialties other than IT security (e.g., management, systems design and development, acquisition, auditing). The most significant difference between training and awareness is that training seeks to teach skills, which allow a person to perform a specific function, while awareness seeks to focus an individual’s attention on an issue or set of issues. The skills acquired during training are built upon the awareness foundation, in particular, upon the security basics and literacy material.

An example of training is an IT security course for system administrators, which should address in detail the management controls, operational controls, and technical controls that should be implemented. Management controls include policy, IT security program management, risk management, and life cycle awareness and training, computer support and operations, and physical and environmental security issues. Technical controls include identification and authentication, logical access controls, audit trails, and cryptography.

Education

Education integrates all of the security skills and competencies of the various functional specialties into a common body of knowledge and strives to produce IT security specialists and professionals capable of vision and proactive response.

Education is defined as follows: The ‘education’ level integrates all of the security skills and competencies of the various functional specialties into a common body of knowledge, adds a multi-disciplinary study of concepts, issues, and principles (technological and social), and strives to produce IT security specialists and professionals capable of vision and pro-active response.

An example of education is a degree program at a college or university. Some people take a course or several courses to develop or enhance their skills in a particular discipline. This is training as opposed to education. Many colleges and universities offer certificate programs, wherein a student may take two, six, or eight classes, for example, in a related discipline, and is awarded a certificate upon completion. Often, these certificate programs are conducted as a joint effort between schools and software or hardware vendors. These programs are more characteristic of training than education. Those responsible for security training need to assess both types of programs and decide which one better addresses identified needs.

Professional Development

Professional development is intended to ensure that users, from beginner to the career security professional, possess a required level of knowledge and competence necessary for their roles. Professional development validates skills through certification. Such development and successful certification can be termed “professionalization.” The preparatory work to test such a certification normally includes study of a prescribed body of knowledge or technical curriculum, and may be supplemented by on-the-job experience.

The movement toward professionalization within the IT security field can be seen among IT security officers, IT security auditors, IT contractors, and system/network administrators, and is evolving. There are two types of certification: general and technical. The general certification focuses on establishing a foundation of knowledge on the many aspects of the IT security profession. The technical certification focuses primarily on the technical security issues related to specific platforms, operating systems, vendor products, etc.

Some organizations focus on IT security professionals with certifications as part of their recruitment efforts. Other organizations offer pay raises and bonuses to retain users with certifications and encourage others in the IT security field to seek certification.

Step 3 – Assign the Roles for Security Awareness

While it is important to have a policy that requires the development and implementation of security and training, it is crucial that IT organizations understand who has responsibility for IT security awareness and training. This step identifies and describes those within an organization that have responsibility for IT security awareness and training.

Some organizations have a mature IT security program, while other organizations may be struggling to achieve basic staffing, funding, and support. The form that an awareness and training program takes can vary greatly from organization to organization. This is due, in part, to the maturity of that program. One way to help ensure that a program matures is to develop and document IT security awareness and training responsibilities for those key positions upon which the success of the program depends.

IT Director/CIO

IT Director and/or the CIO must ensure that high priority is given to effective security awareness and training for the workforce. This includes implementation of a viable IT security program with a strong awareness and training component. The IT Director should:

- Assign responsibility for IT security
- Ensure that an organization-wide IT security program is implemented, is well-supported by resources and budget, and is effective
- Ensure that the organization has enough sufficiently trained personnel to protect its IT resources
- Establish overall strategy for the IT security awareness and training program
- Ensure that senior managers, system and data owners, and others understand the concepts and strategy of the security awareness and training program, and are informed of the progress of the program's implementation
- Ensure that the IT security awareness and training program is funded
- Ensure the training of personnel with significant security responsibilities
- Ensure that all users are sufficiently trained in their security responsibilities
- Ensure that effective tracking and reporting mechanisms are in place.

Information Technology Security Program Manager

The IT Security Program Manager has tactical-level responsibility for the awareness and training program. In this role, the program manager should:

- Ensure that awareness and training material developed is appropriate and timely for the intended audiences
- Ensure that awareness and training material is effectively deployed to reach the intended audience
- Ensure that users and managers have an effective way to provide feedback on the awareness and training material and its presentation
- Ensure that awareness and training material is reviewed periodically and updated when necessary
- Assist in establishing a tracking and reporting strategy

IT Managers

Managers have responsibility for complying with IT security awareness and training requirements established for their users. Managers should:

- Work with the CIO and IT security program manager to meet shared responsibilities
- Serve in the role of system owner and/or data owner, where applicable
- Consider developing individual development plans (IDPs) for users in roles with significant security responsibilities
- Promote the professional development and certification of the IT security program staff, full-time or part-time security officers, and others with significant security responsibilities
- Ensure that all users (including contractors) of their system (i.e., general support systems and major applications) are appropriately trained in how to fulfill their security responsibilities before allowing them access
- Ensure that users (including contractors) understand specific rules of each system and application
- Work to reduce errors and omissions by users due to lack of awareness and/or training.

Users

Users are the largest audience in any organization and are the single most important group of people who can help to reduce unintentional errors and IT vulnerabilities. Users may include employees, contractors, foreign or domestic guest researchers, other agency personnel, visitors, guests, and other collaborators or associates requiring access. Users must:

- Understand and comply with the IT security policies and procedures

- Be appropriately trained in the rules of behavior for the systems and applications to which they have access
- Work with management to meet training needs
- Keep software/applications updated with security patches
- Be aware of actions they can take to better protect their information. These actions include, but are not limited to: proper password usage, data backup, proper anti-virus protection, reporting any suspected incidents or violations of security policy, and following rules established to avoid social engineering attacks and rules to deter the spread of spam or viruses and worms.

Security Practices

When addressing security issues, some general information security principles should be kept in mind, as follows:

- **Simplicity**
Security mechanisms (and information systems in general) should be as simple as possible. Complexity is at the root of many security issues.
- **Fail Safe**
If a failure occurs, the system should fail in a secure manner. That is, if a failure occurs, security should still be enforced. It is better to lose functionality than lose security.
- **Complete Mediation**
Rather than providing direct access to information, mediators that enforce access policy should be employed. Common examples include file system permissions, web proxies and mail gateways.
- **Open Design**
System security should not depend on the secrecy of the implementation or its components. “Security through obscurity” does not work.
- **Separation of Privilege**
Functions, to the degree possible, should be separate and provide as much granularity as possible. The concept can apply to both systems and operators and users. In the case of system operators and users, roles should be as separate as possible. For example, if resources allow, the role of system administrator should be separate from that of the security administrator.
- **Psychological Acceptability**
Users should understand the necessity of security. This can be provided through training and education. In addition, the security mechanisms in place should present

users with sensible options that will give them the usability they require on a daily basis. If users find the security mechanisms too cumbersome, they find ways to work around or compromise them. An example of this is using random passwords that are very strong but difficult to remember; users may write them down or look for methods to circumvent the policy.

- Layered Defense

Organizations should understand that any single security mechanism is generally insufficient. Security mechanisms (defenses) need to be layered so that compromise of a single security mechanism is insufficient to compromise a host or network. There is no “magic bullet” for information system security.

- Compromise Recording

When systems and networks are compromised, records or logs of that compromise should be created. This information can assist in security of the network and host after the compromise and assist in identifying the methods and exploits used by the attacker. This information can be used to better secure the host or network in the future. In addition, the records and logs can assist organizations in identification and prosecution.

Outsourcing, COTS and Contracting Quality

Organizations can assign software development work responsibilities to outside organizations through purchasing software or contracting services; but they cannot assign the responsibility for quality. Quality of software remains an internal IT responsibility regardless of who builds the software. The quality professionals need to assure that those quality responsibilities are fulfilled through appropriate processes for acquiring purchased software and contracting for software services.

<i>Quality and Outside Software</i>	421
<i>Selecting COTS Software</i>	425
<i>Selecting Software Developed by Outside Organizations</i>	432
<i>Contracting for Software Developed by Outside Organization</i>	439
<i>Operating for Software Developed by Outside Organization</i>	444

Quality and Outside Software

There is a trend in the software industry for organizations to move from in-house developed software to commercial off-the-shelf (COTS) software and software developed by contractors. Software developed by contractors who are not part of the organization is referred to as *outsourcing organizations*. Contractors working in another country are referred to as *offshore software developers*.

There are some common differences between any software developed by an outside organization, and then differences specific to COTS and contractor developed software. Quality professionals should be familiar with these differences as they impact their quality responsibilities.

Two differences between software developed in-house and software developed by an outside organization are:

- Relinquishment of control
The software is developed by individuals who are not employees of the organization, and thus it is difficult to oversee the development process. The contracting organization cannot direct the employees of the other organization, nor have control over the many day-to-day decisions that are made in developing software.
- Loss of control over reallocation of resources
If work needs to be done to correct problems and/or speed up development, the contractor cannot take workers off one project and assign them to another project.

Purchased COTS software

COTS software is normally developed prior to an organization selecting that software for its use. For smaller, less expensive software packages the software is normally “shrink wrapped” and is purchased as-is. As the COTS software becomes larger and more expensive, the contractor may be able to specify modifications to the software.

Differences or challenges faced with COTS software include:

- Task or items missing
- Software fails to perform
- Extra features
- Does not meet business needs
- Does not meet operational needs
- Does not meet people needs

Evaluation versus Assessment

Many organizations select COTS software on evaluation which is a static analysis of the documentation and benefits of the software, versus performing an assessment which the software will be tested in a dynamic mode before use.

Outsourced Software

The differences in contracted software developed by an outsourcer include:

- **Quality factors may not be specified**
There are many factors such as reliability and ease of use which are frequently not included as part of the contractual criteria. Thus when the software is delivered it may not be as easy to use or as reliable as desired by the contractor.
- **Non-testable requirements and criteria**
If the requirements or contractual criteria are in measurable and testable terms then the delivered result may not meet the intent of the contractor.
- **Customer's standards may not be met**
Unless the contract specifies the operational standards and documentation standards the delivered product may be more complex to use than desired by the contractor.
- **Missing requirements**
Unless detailed analysis and contractual specifications work is complete the contractor may realize during the development of the software that requirements are missing and thus the cost of the contract could escalate significantly.
- **Overlooked changes in standards in technology**
If changes in standards that the organization must meet, or the introduction of new desirable technology is incorporated into the contract there may be significant cost to modify the software for those new standards in technology.
- **Training and deployment may be difficult**
If software is developed by another organization there may be inadequate knowledge in the contracted organization to provide the appropriate training for staff and to ensure that deployment is effective and efficient.

Additional differences if the contract is with an offshore organization

Experience has shown that over 50% of the software developed by offshore organizations fails to meet the expectations of the contractor. Since many of the decisions to have software developed offshore are economic decisions, the differences associated with having the software developed offshore negate the economic advantages in many cases. These offshore differences are:

- **Cultural differences**
There may be a significant difference in the culture and values between the contractor and the offshore organization.
- **Communication barriers**
The language of the offshore organization may be different or difficult to comprehend which causes difficulty in communicating the needs and desires of the contractor.

- Loss of employee morale and support
Employees who would like to have developed the software may resent the software being developed offshore and thus make it difficult for the offshore developed software to be successful.
- Root cause of the contractor IT organization not addressed
Frequently, offshoring is chosen because there are problems in the contractor organization that executives do not want to address. For example, the problems might include a lack of training for the employees in the contractor organization or other perhaps better options for software development were not explored.

The above discussions are not meant to be an exhaustive list of the differences between in-house developed software and software developed by outside organizations. The objective is so the quality assurance professional recognizes some potential root causes of software quality. If those differences are not adequately addressed in the contract, and with employees of the organization, the probability of the contracted or offshore-developed software increases.

Quality Professionals Responsibility for Outside Software

While the software may be developed by an outside organization, the responsibility for quality of that software cannot be contracted. The contractor is still responsible for the quality of the organization. There must be a process to monitor the development and validate the correct functioning of the software when it is developed by an outside organization.

The quality professional is the individual who should accept the quality responsibility for software developed by an outside organization. This may mean that the quality professional needs to visit periodically or during the entire developmental period of the software to ensure the quality. Many of the same practices used to assure quality of in-house developed software are applicable to software developed by outside organizations. For example, conducting reviews at specific checkpoints should occur on contracted software. Acceptance testing should be conducted on all software regardless of how developed.

The quality professional's specific responsibility for software developed by outside organizations is to assure that the process for selecting COTS software and contracting with an outside organization for software are adequate.

One of the major responsibilities of the quality assurance activity is to oversee the development and deployment of work processes, and then to assure that those work processes are continuously improved.

Without a process for selecting COTS software and a process for contracting for software those processes would be subject to great variability. One contract may work well, one acquisition of COTS software may work well, while other acquisitions may result in a failure.

The quality professional needs to look at these two processes in the same way that they view the SEI CMMI® Capability Maturity Model. If contracting is done at a Level 1 maturity there will be great variability and thus many disappointments in the delivered product and services. On the

other hand, as those processes move to a Level 5 maturity, the probability of getting exactly what is wanted from COTS software and contracted software is very high.

This category contains a prototype process for selecting COTS software and a prototype process for contracting for software with outside organizations. The two processes incorporate many of the best practices for acquiring software developed by outside organizations.

Selecting COTS Software

There is no generally accepted best practice for acquiring COTS software. However, there are many practices in place by organizations that have a process for selecting COTS software. The process proposed includes many of those best practices.

It is important for the quality professional to understand first that a process is needed for acquiring COTS software, and second understanding the key components of that process. Thus, the purpose of presenting a process for selecting COTS software is to facilitate the quality professional understanding the type of criteria that should be included in a COTS software selection process.

The following seven-step process includes those activities which many organizations follow in assuring that the COTS software selected is appropriate for the business needs. Each of the processes is discussed below:

- Assure Completeness of Needs Requirements
- Define Critical Success Factor
- Determine Compatibility with Hardware, Operating System, and other COTS Software
- Assure the Software can be Integrated into Your Business System Work Flow
- Demonstrate the Software in Operation
- Evaluate People Fit
- Acceptance Test the Software Process

Assure Completeness of Needs Requirements

This step determines whether you have adequately defined your requirements. Your requirements should be defined in terms of the following two categories of outputs:

1. Output Products and Output Reports.

Output products and output reports are specific documents that you want produced by the computer system. In many instances, such as the previous payroll check example, the style and format of these output products is important. This does not mean that the specific location of the check has to be defined but, rather, the categories of information to be included on the check. Computer-produced reports may also be important for tax information (e.g., employee withholding forms sent to governmental units), financial

statements where specific statements are wanted (e.g., balance sheets or statements of income and expense), or customer invoice and billing forms which you might want preprinted to include your logo and conditions of payment.

2. Management decision information. This category tries to define the information needed for decision-making purposes. In the computer product/report category you were looking for a document; in this case you are looking for information. How that information is provided is unimportant. Thus, the structure of the document, what the documents are, or their size, frequency, or volume are not significant. All you need is information.

Define Critical Success Factor

This step tells whether the COTS software will be successful in meeting your requirements.

Critical success factors (CSFs) are those criteria or factors that must be present in the acquired software for it to be successful. You might ask whether the needs are the same as the critical success factors. They are, but they are not defined in a manner that makes them testable, and they may be incomplete. Often the needs do not take into account some of the intangible criteria that make the difference between success and failure. In other words, the needs define what we are looking for, and the critical success factors tell us how we will evaluate that product after we get it. They are closely related and complementary, but different in scope and purpose.

The following list indicates the needs or requirements for the automobile, and is then followed by the CSFs on which the automobile will be evaluated:

- Seats six people
- Four doors
- Five-year guarantee on motor
- Gets 20 miles per gallon or greater
- Costs under \$12,000
- Critical success factors:
 - Operates at 20.5 cents or less per mile
 - Experiences no more than one failure per year
 - Maintains its appearance without showing signs of wear for two years

Some of the more common critical success factors for COTS you may want to use are:

- Ease of use
 - The software is understandable and usable by the average person.
- Expandability
 - The contractor plans to add additional features in the future.

- **Maintainability**
The contractor will provide support/assistance to help utilize the package in the event of problems.
- **Cost-effectiveness**
The software package makes money for your business by reducing costs, and so on.
- **Transferability**
If you change your computer equipment the contractor indicates that they will support new models or hardware.
- **Reliability**
In computer language, the system is friendly, meaning that it will help you get your transactions entered into the system so that you can produce your results readily.
- **Security**
The system has adequate safeguards to protect the data against damage (for example, power failures, operator errors, or other goofs that could cause you to lose your data).

Determine Compatibility with Hardware, Operating System, and Other COTS Software

This is not a complex step. It involves a simple matching between your processing capabilities and limitations, and what the contractor of the software says is necessary to run the software package. The most difficult part of this evaluation is ensuring the multiple software packages can properly interface.

This step is best performed by preparing a checklist defining your compatibility needs. Software contractors are generally good about identifying the needed hardware and operating system compatibility. They are generally not good in identifying compatibility with other software packages.

In addition to the hardware on which the software runs, and the operating system with which it must interact to run, there are two other important compatibilities:

- Compatibility with other software packages
- Compatibility with available data.

If you have no other software packages that you want to have interact with this one, or no data on computer-readable media, you need not worry about these aspects of compatibility. However, as you do more with your computer these aspects of compatibility will become more important, and the hardware and operating compatibility will become routine and easy to verify.

Systems compatibility is defined in data processing jargon as “interoperability.” This term refers to the amount of effort required to inter-couple or interconnect computer systems. In other words,

how do you tie two or more programs together so that they will work and pass data between them? For example, if you have a payroll system it may be desirable to pass that payroll summary information to your general-ledger system. The ability to pass information from system to system is an extremely important part of data processing. Much of the success of the Lotus Corporation was based on its ability to inter-couple five office software functions so that information could be readily passed from one to another.

To help prepare a compatibility list for the purpose of assuring compatibility, the information that needs to be included is described below. The list is divided into hardware, operating systems, programs, and data.

Hardware Compatibility

List the following characteristics for your computer hardware:

- Hardware contractor
- Amount of main storage
- Disk storage unit identifier
- Disk storage unit capacity
- Type of printer
- Number of print columns
- Type of terminal
- Maximum terminal display size
- Keyboard restrictions

Operating Systems Compatibility

For the operating system used by your computer hardware, list:

- Name of operating system
- Version of operating system in use

Program Compatibility

List all of the programs that you expect or would like to interact with this specific application. Be sure that you have the name of the contractor and, if applicable, the version of the program. Note that as discussed earlier, this linkage may only be verifiable by actually attempting to interact two or more systems using common data.

Data Compatibility

In many cases, program compatibility will answer the questions on data compatibility. However, if you created special files you may need descriptions of the individual data elements and files.

Again, as with program compatibility, you may have to actually verify through trial whether the data can be read and used by other programs.

Assure the Software can be Integrated into Your Business System Work Flow

Each computer system makes certain assumptions. Unfortunately, these assumptions are rarely stated in the contractor literature. The danger is that you may be required to do some manual processing functions that you may not want to do in order to utilize the software.

The objective of this step is to determine whether you can plug the COTS into your existing manual system without disrupting your entire operation. Remember that:

- Your manual system is based on a certain set of assumptions.
- Your manual system uses existing forms, existing data, and existing procedures.
- The computer system is based on a set of assumptions.
- The computer system uses a predetermined set of forms and procedures.
- Your current manual system and the new computer system may be incompatible.
- If they are incompatible, the computer system is not going to change—you will have to.
- You may not want to change—then what?

The objective of this process is to illustrate the type and frequency of work flow changes that will be occurring. You can see graphically illustrated what will happen when the computer system is brought into your organization. For example, there might be tasks performed now that weren't performed before, or tasks that were previously performed but are no longer necessary, or tasks which had been performed by people which will now be performed by the computer. Having the computer perform those tasks might mean that the oversight that people had been giving will not be available any more.

At the end of this test, you will need to decide whether you are pleased with the revised work flow. If you feel the changes can be effectively integrated into your work flow, the potential computer system has passed the test. If you feel the changes in work flow will be disruptive, you may want to fail the software in this test and either look for other software or continue manual processing.

If the testing is to continue, you should prepare a clean data flow diagram indicating what actions need to be taken to integrate the computer system into your organization's work flow. This new data flow diagram becomes your installation plan of action. It will tell you what changes need to be made, who is involved in them, what training might be necessary, and areas of potential work flow problems.

Demonstrate the Software in Operation

This step analyzes the many facets of software. Software developers are always excited when their program goes to what they call “end of job.” This means that it executes and concludes without abnormally terminating (i.e., stops after doing all the desired tasks). While this is one aspect of the demonstration, observing the functioning of software is like taking an automobile for a test drive. The more rigorous the test, the greater the assurance you are getting what you expect.

Demonstrations can be performed in either of the following ways:

- Computer store – controlled demonstration

In this mode, the demonstration is conducted at the computer store, by computer store personnel, using their data. The objective is to show you various aspects of the computer software, but not to let you get too involved in the process. This is done primarily to limit the time involved in the demonstration.

- Customer site demonstration

In this mode, the demonstration takes place at your site, under your control, by your personnel, using your information. It is by far the most desirable of all demonstrations, but many software COTS computer stores may not permit it unless you purchase the COTS.

These aspects of computer software should be observed during the demonstration:

- Understandability

As you watch and listen to the demonstration, you need to evaluate the ease with which the operating process can be learned. If the commands and processes appear more like magic than logical steps, you should be concerned about implementing the concept in your organization. If you have trouble figuring out how to do it, think about how difficult it may be for some of your clerical personnel who understand neither the business application nor the computer.

- Clarity of communication

Much of the computer process is communication between man and machine. That is, you must learn the language of the computer software programs in order to communicate with the computer. Communication occurs through a series of questions and responses. If you do not understand the communications, you will have difficulty using the routine.

- Ease of use of instruction manual

While monitoring the use of the equipment, the tasks being demonstrated should be cross-referenced to the instruction manual. Can you identify the steps performed during the demonstration with the same steps included in the manual? In other words, does the operator have to know more than is included in the manual, or are the steps to use the process laid out so clearly in the manual that they appear easy to follow?

- **Functionality of the software**

Ask to observe the more common functions included in the software: Are these functions described in the manual? Are these the functions that the salesperson described to you? Are they the functions that you expected? Concentrate extensively on the applicability of those functions to your business problem.
- **Knowledge to execute**

An earlier test has already determined the extent of the salesperson's knowledge. During the demonstration, you should evaluate whether a lesser-skilled person could as easily operate the system with some minimal training. Probe the demonstrator about how frequently they run the demonstration and how knowledgeable they are about the software.
- **Effectiveness of help routines**

Help routines are designed to get you out of trouble when you get into it. For example, if you are not sure how something works you can type the word "help" or an equivalent and the screen should provide you additional information. Even without typing "help" it should be easy to work through the routines from the information displayed on the screen. Examine the instructions and evaluate whether you believe you could have operated the system based on the normal instructions. Then ask the operator periodically to call the help routines to determine their clarity.
- **Evaluate program compatibility**

If you have programs you need to interact with, attempt to have that interaction demonstrated. If you purchased other software from the same store where you are now getting the demonstration, they should be able to show you how data is passed between the programs.
- **Data compatibility**

Take one of your data files with you. Ask the demonstrator to use your file as part of the software demonstration. This will determine the ease with which existing business data can be used with the new software.
- **Smell test**

While watching the demonstration, let part of your mind be a casual overseer of the entire process. Attempt to get a feel for what is happening and how that might impact your business. You want to end up being able to assess whether you feel good about the software. If you have concerns, attempt to articulate them to the demonstrator as well as possible to determine how the demonstrator responds and addresses those concerns.

To determine whether an individual has the appropriate skill level to use the software it is recommended to involve one or more typical users of the software in software demonstrations.

Evaluate People Fit

The objective of this step is to determine whether your employees can use the software. This testing consists of ensuring that your employees have or can be taught the necessary skills.

This step evaluates whether people possess the skills necessary to effectively use computers in their day-to-day work. The evaluation can be of current skills, or the program that will be put into place to teach individuals the necessary skills. Note that this includes the owner-president of the organization as well as the lowest-level employee in the organization.

The test is performed by selecting a representative sample of the people who will use the software. The sample need not be large. This group is given training that may only involve handing someone the manuals and software. The users will then attempt to use the software for the purpose for which it was intended. The results of this test will show:

- The software can be used as is.
- Additional training and support is necessary.
- The software is not usable with the skill sets of the proposed users.

Acceptance Test the Software Process

The objective of this step is to validate that the software will in fact meet the functional and structural needs of the user of the software.

We have divided testing into functional and structural testing, which also could be called correctness and reliability testing. “Correctness” means that the functions produce the desired results. “Reliability” means that the correct results will be produced under actual business conditions.

Selecting Software Developed by Outside Organizations

Normally contracting for software is a more significant activity than acquiring COTS software. COTS software has been developed and the acquisition challenge is to assure that the developed software will meet the organization’s needs. On the other hand the contracted software may not be developed or may be only partially developed and thus it incorporates many of the aspects of in-house developed software except the actual implementation of the requirements/criteria.

There is no one single process generally accepted for contracting for software. Many large organizations have a purchasing section which specifies many of the components of contracting for software. In addition, large organization’s legal departments may also be involved in writing and approving contracts for software development.

If an off-shore organization develops the software, even more attention should be given to the contract because of cultural and communication differences. Off-shore contracts may involve the laws of the country in which the software is developed.

The following contracting life cycle incorporates many of the best practices used by organizations in selecting, contracting, and operating software developed by an outside organization. This process commences when the need for software developed by an outside organization is defined, and continues throughout the operation of the contracted software.

Contracting Life Cycle

In order to participate in the various aspects of contracting for software development, it is necessary to establish an acquisition life cycle for contracted software. This life cycle contains the following three activities:

- **Selecting an Outside Organization**

Selecting a contractor is similar to systems design. It is a time of studying alternative contractors, costs, schedules, detailed implementation design specifications, and the specification of all the deliverables, such as documentation. The selection of an outside organization may involve the following individuals in addition to the quality control reviewer:

- Systems analysts
- User personnel
- Internal auditor
- Purchasing agent
- Legal counsel

- **Contract Negotiations**

In some organizations, the purchasing agent conducts all the negotiations with the contractor; the other parties are involved only in the needs specification. In other organizations, there is no purchasing agent, so the data processing department deals directly with contractors for the acquisition of application systems.

- **Operations and Maintenance**

The maintenance and operations of purchased applications may be subject to contractual constraints. For example, some contracts limit the frequency with which an application can be run without paying extra charges, and limit an organization's ability to duplicate the application system in order to run it in a second location. Some purchased applications can be maintained by in-house personnel, but others do not come with source code and thus are not maintainable by in-house personnel. There may also be problems in connecting a purchased application from one contractor with purchased or rented software from another contractor. It is important that software

from multi contractors be evaluated as to its capability to work in the same operating environment.

The contract lists the obligations assumed by both parties. The contractor may be obligated to meet contractual requirements for such things as updating the application system to be usable with new versions of operating systems, etc. The contractor may have obligations for protecting the application system from compromise, paying for extensive use of the application, etc. This provision should be monitored and enforced during the life of the contract. Provisions which are violated and not enforced may be unenforceable at a later point in time due to the implied agreement by one party not to enforce a provision of the contract. Therefore, it is important that contracts be reviewed regularly to determine that all the provisions of the contract are being enforced.

Selecting an Outside Organization

The activities that can be included in this step are discussed below.

Feasibility Study

Feasibility studies serve the purpose of identifying and evaluating the alternative solutions to satisfy a need. Feasibility studies normally do not get into the detailed methods of implementing the solution. However, the availability of an application may instigate a feasibility study, or the solution to the need may dictate that the application should be purchased.

There are several concerns that need to be addressed when one of the methods of implementation is contracting for software development. These review concerns are:

- Control specification

The controls desired in the application should be specified in enough detail during the feasibility study so that the individual negotiating for an application can include those control specifications in the contract negotiation.

- Needed installation date

The date on which the application is to go into production should be specified. Many applications lose a large portion of their benefits if the application is not installed when needed. For example, an application designed to provide special promotions for Christmas shopping would be of little value if it is not completed in time for Christmas purchasing. Specifying this date may indicate whether an application needs to be purchased, or whether it can be developed in-house.

- Value of applications

The benefits to be obtained from an application should be specified. These are the benefits above and beyond what is currently being obtained from the existing method of performing the tasks. The benefits need to be quantified. For example, benefits like improved customer service are of limited aid in making business decisions on installing or purchasing new applications. The objective of having a stated dollar value

from an application is to provide guidance for the cost-benefit calculation for that application. The final benefits and costs should be estimated by those individuals conducting the contract negotiations. This does not mean that the benefits will not change based on suggestions or methods incorporated into the purchased application.

- Systems specifications defined

The feasibility study should extensively define the desired specifications. These specifications should include:

- The input needed by the system, including its attributes and desired reliability.
- Required processing.
- The output to be produced by the system, showing report layouts if possible.
- Operational response times, such as providing terminal response within five seconds.
- Information to be retained, the data and the length of retention, as well as the media of retention if known.
- Types of documentation, forms, etc., needed to be developed.
- Volumes of data processed.
- Unique hardware characteristics.

- Useful life

The expected life of the application should be stated. For example, if it is expected that the requirements and technology needed will satisfy the needs of the organization for an eight-year period, that should be stated. Where it is difficult to estimate a specific useful life, a range such as 5-10 years is often sufficient.

- Confidentiality of application

Feasibility studies should state the confidentiality of the application to the organization. This importance would deal with the need to keep the processing rules confidential. For example, the application may include the formula used in creating products for production. Since the application contains those formulas, it contains the trade secrets of the organization which must be adequately protected.

- Confidentiality of data

Many applications are not confidential until data is processed by those applications. For example, an application that maintains a customer list is a simple file update system not requiring any special type of protection. However, once the customers are entered into that system, the customer file is a valuable asset of the organization and thus requires protection. The type of data and the degree of protection it requires needs to be stated as this can significantly affect the way the application is constructed and operated.

- Legal implications

The processes or data included in the application may be governed by laws and regulations. For example, payroll processing is governed by the Federal Wage and Hour Law, among others. The implementation of those applications must produce processing and information that is in compliance with those laws and regulations. The legal requirements should be stated as part of the systems specifications.

- Contract implications

There may be contractor, customer, application, or data characteristics that affect contractor selection and contract specification. For example, an organization may decide that certain contractors are also competitors, and thus does not want to enter into contract negotiations with them to purchase an application. In other instances, the organization may feel that certain contractors would not adequately protect the confidentiality of processing needs and data requirements given to a contractor. These types of contractual restrictions or recommendations should be included in the feasibility study.

Selection of an Outside Organization

The most important step of the acquisition life cycle is contractor selection. This step has the same importance in the acquisition life cycle as the systems design step has in the system development life cycle. It is determined during this step whether or not the purchased application meets the needs of the organization. During the systems design phase, the needs are translated into systems design specifications. During selection those needs are translated into contractual requirements.

Selecting the right organization to develop software is critical to successfully outsourcing development. The selection process must find a competent and compatible organization to develop all or part of the software development. What cannot be outsourced is the responsibility for software quality. The following concerns need to be addressed when selecting an outsourced organization.

Assure That Requirements and Contract Criteria are Testable

A common term used for contracting today is *performance-based contracting*. This means that the performance criteria of the contractor will be defined and once defined, can be monitored and measured.

It is important that all the requirements that will be given to the contractor are testable. That means as much as possible, an objective measurement can be made as to whether or not that requirement or criteria has or has not been met. For example, easy-to-use criteria might specify the type and number of help screens included within the software.

As a general rule, if a requirement or criteria is not testable the contractor has significant discretion on how that requirement or criteria is implemented. It also limits the contractee's ability to have corrections made without additional cost for a requirement or criteria that does not meet the customer's need. If it is a testable requirement, it is easy to demonstrate that the contractor has failed to meet the component of the contract.

Assure That the Contractor Has an Adequate Software Development Process

The contractor should provide a certificate indicating what SEI CMMI level they have achieved or equivalent; for example, ISO certified.

Assure That the Contractor Has an Effective Test Process

This responsibility is primarily for contracted developed software. It is important for the quality professional to develop an opinion on the contractor's ability to deliver software that meets the requirements or criteria in the contract. It can also provide insight on the ability of the contractor to develop software at a reasonable cost. For example, if the test plan indicates that the defect removal efficiency at the requirements phase is 95%, then the quality professional knows that only 5% of the requirement defects will move to the design phase. On the other hand, if the contractor does not begin testing until after the unit is compiled, then there may be extensive rework and potential delays in getting the software on the scheduled date.

Define Acceptance Testing Criteria

The quality professional should not allow COTS or contracted software to be placed into operation without some acceptance testing. If the COTS software is acquired by the internet or from a business college its reliability is questionable, and it may contain viruses. Many organizations do not permit their employees to acquire software from unapproved sources such as the internet.

The quality professional may or may not be involved in the actual acceptance testing. However, the quality professional must determine that acceptance criteria are developed. It has been demonstrated extensively that the cost of not acceptance testing is normally much greater than the cost of acceptance testing. It only takes a few problems in acquired software to far exceed the cost of acceptance testing.

Acceptance testing of software will be dependant upon the risk associated with the use of that software. As the use and importance diminishes so does the amount of acceptance testing. Likewise, the greater assurance the quality professional has in the ability of a contractor of software to produce high quality, defect-free software, the fewer acceptance testing that needs to be conducted.

At a minimum the acceptance testing should validate:

- The documentation is consistent with the software execution.
- The documentation is understandable.
- Users will be adequately trained in the software prior to use of the software.
- It is operable within the operational constraints of the organization. For example, all the features and the platform that are needed to operate the software are in fact there.

Contractor's Status Reporting

The quality professional should assure that reports will be issued regularly by the contractor on the software being developed.

Ensure Knowledge Transfer Occurs

There are two concerns that the quality professional must assure about software developed by outside organizations. The first is that there is adequate knowledge transfer about the software from the developer. The second is that the intellectual property rights of both the contractor and contracted are protected.

The amount of knowledge transfer will be dependant upon the purchase or contractual arrangements for acquiring the software. For example, contractors may not release source code to the contracted. The importance of this issue can change based on relationship with the developer and/or whether the developer stays in business. For example, if the contractor goes out of business, does the contracted then have the right to obtain the source code so the software can be maintained even though the contractor of the software is no longer in business?

Among the items that may be important in knowledge transfer are:

- Training programs for contracted staff.
- Being advised of defects uncovered by other organizations using the software.
- Ability to contact a contractor help desk to resolve problems/get additional information.

There are legal aspects of protecting the intellectual property acquired from an outside organization. The contracted may not have the right to reproduce and distribute the software without additional compensation to the developing organization.

The contracted may have to exercise reasonable care to protect the rights of the developing organization, such as securing and/or providing adequate protection over the software and associated documentation.

Ensure Protection of Intellectual Property Rights of Both Organizations

The contracting organization may also have intellectual property that they want protected. For example, they may share with the developer of the software proprietary material that they do not want the developer to use for any other purpose.

In some instances, the contracting organization wants access to the software, or developer materials prior to acquiring or contracting for software. This and other aspects of protecting intellectual property may be covered in a nondisclosure agreement between the outside developer of the software and the acquiring organization.

Developing Selection Criteria

The selecting organization needs to develop the criteria they will use to determine who will develop their software. The criteria previously described, such as the maturity of the contractor's software development and test process, the testability of the requirements, and knowledge transfer are typical selection criteria. Other criteria include cost, reputation of the contractor, compatibility of cultures, references, and delivery dates are all equally important.

The selection should be made on both the importance of the selection criteria and the contractor's ability to meet the selection criteria. Some criteria may be classified as essential, meaning if a contractor cannot meet those criteria they are eliminated from consideration.

Contracting for Software Developed by Outside Organizations

What Contracts Should Contain

Contracts are legal documents. To fully understand the impact of provisions being included in, or excluded from, the contract may require legal training. However, the following information should be included in all contracts:

- What is done.

The contract should specify the deliverables to be obtained as a result of execution of the contract. Deliverables should be specified in sufficient detail so that it can be determined whether or not the desired product has been received.

- Who does it.

The obligations of both contractual parties should be spelled out in detail in the contract.

- When it is done.

The dates on which the contractual obligations need to be filled should be specified in the contract.

- How it is done.

The contract should specify the methods by which the deliverables are to be prepared if that is important in achieving the contractual needs. For example, the organization may not want certain types of source instructions used in developing an application system because they plan to perform the maintenance with the in-house personnel.

- Where it is done.

The location where the deliverables are to be prepared, delivered, operated, and maintained should be specified as part of the contract.

- Penalties for nonperformance.

If the contractual agreements are not followed, the penalties should be specified in the contract. For example, if the contractor is late in delivering the work products, the contract may specify a penalty of x dollars per day.

Contract Negotiations

The concerns that need to be addressed during contract negotiations include the following factors.

- Warranty

The guarantees provided by the contractor that the deliverables will meet the specifications. This segment of the contract should state specifically what the contractor warrants, and what the contractor will do if the deliverables fail to meet the warranty guarantees. For example, if the contractor guarantees the application will provide a one-second response, and the implemented application fails to meet those specifications, this contract defines recourse against the contractor.

- Deliverables

The application system, documentation, and other products to be provided by the contractor should be described in great detail. For example, a phrase like “provide for adequate controls” is a meaningless phrase in that the deliverables are not measurable. The contractor’s definition of “adequate controls” may be entirely different than that of the customer, but loose wording such as this, except in cases of gross negligence, eliminates recourse. The product specifications should include as much detail as practical, and as much as necessary to ensure that the organization gets the product they want.

- Delivery date

The date on which the product is to be delivered should be specified in the contract. This may be multiple dates, in that a product may be delivered for testing and then another date specified for when those corrections will be made, etc.

- Commencement date

The date at which the contract becomes effective should be specified in the contract. This is particularly important if delivery dates are keyed to the commencement date, such as; a deliverable will be available sixty days after the contract is signed.

- Installation

The contractor’s and customer’s commitment for installing the application should be specified. If the contractor is to have personnel present to help during the installation, that should be specified. If the contractor is to provide machine time on certain days, that, too, should be specified in the contract.

- Updates

The type of continual maintenance provided by the contractor for the application system should be specified. This is particularly important if the customer operates in an environment where operating systems are regularly changed. The contract might specify that the contractor will provide necessary updates so that the system will operate on new versions of the operating system being used by the customer.

- Contractor support

The types, quantity, and location of contractor support should be specified. In addition, some organizations specify the caliber of people that should provide that support. For example, systems analysts should have a minimum of five years' programming and systems experience, and at least one year experience with the application system. It is also important to specify where the support will occur. For example, is contractor support to be provided at the customer's place of business, or must the customer's personnel go to the contractor's place of business to get that support?

- Costs

The amounts to be paid to the contractor by the customer should be specified, including payment terms. If there are penalty clauses for late payments, that should be specified, as well as the rights of the customer to withhold payments if the contractor fails to provide individual deliverables or other support. Ideally, the value for each deliverable should be specified so that the amounts withheld are determined by contract for failure to perform. The more precise the description, the fewer the problems.

- Foreign attachments

If the application system is interconnected with applications and/or other software from different contractors, that interrelationship should be specified. It is important to state in the contract that has primary responsibility for tracing and correcting errors. In a multi-contractor environment, when no contractor accepts primary responsibility for error tracking, the customer may need to expend large amounts of funds to trace errors because of the unwillingness of contractors to accept this responsibility.

- Penalties

Penalties assessed in the event of failure on the part of either the contractor or the customer to meet contractual obligations should be covered in the contract. Where dollar penalties are to be assessed, the amount should be specified in the contract. For example, if the contractor is late in delivering the product, a per-day dollar penalty can be assessed, as well as a dollar penalty for failure of the customer to make computer time available, etc.

- Life of contract

The duration of the contract should be specified, including any rights to continue or discontinue the contract at the end of its life. For example, the customer may have the right to extend the contract another year for X dollars.

- Modification capability

The contract should specify what type of modifications (to the application system) the contractor is willing to make. This should include how promptly modifications will be made, and the costs and other terms associated with making those modifications. The contract should also state what type of modifications can be made by the customer, and which ones must be made by the contractor.

- Service discontinuance

If the contractor decides to discontinue service on the application system, the customer's rights in those instances should be specified. For example, if any of the training material or application system is copyrighted then those copyrights should be passed to the customer in the event that service on the purchased products is discontinued.

- Manual/training discontinuance

If the contractor decides to discontinue training manuals, service manuals, or training courses, the rights to that material should revert to the customer. If this is not included in the contract, the customer may be prevented from using material and training courses copyrighted by the contractor, without making additional payments.

- Acceptance test criteria

The contract should specify the criteria which determine that the delivered product is acceptable. The acceptance test criteria should not only cover the delivered application system, but any documentation and training material to be included with the application system. The contract should also state where and by whom the acceptance test is to be performed.

- Purchase versus lease

The options of the customer to either purchase or lease the application system should be outlined in the contract. If it is a lease contract, the rights of the customer to purchase that application, if available, should be specified in the contract.

- Fairness of contract

Both the customer and the contractor should want a contract that is fair to both parties. If the contract is extremely harsh, one or the other parties may find it more desirable to terminate than to continue the contract. For example, if the penalty clauses to the contractor are extremely harsh, and the contractor finds the effort to prepare the deliverables is greater than anticipated, it may be more advantageous to the contractor to terminate the contract than to be late and pay the unrealistic penalty amounts. Thus, an unfair contract may not achieve the desired objectives on the part of either or both parties.

- Performance of maintenance

The location, method, and timing of the performance of maintenance should be specified in the contract. If it is important for the customer that the maintenance be performed at the customer's place of business, that needs to be specified in the contract. If not, the contractor can provide maintenance at the contractor's convenience as opposed to the customer's convenience.

- Contractor training

The type, frequency, caliber of instructors, and location of contractor training for the application system should be included in the contract.

- Contractor manuals

The manuals needed to operate and maintain the application system should be specified in the contract. These normally include computer operator's manuals, user manuals, learning manuals, and systems documentation manuals. The contract should be as specific as possible regarding the size, content, method of presentation, and continued maintenance of the material in the manual.

- Supplies

The types of supplies provided by the contractor should be specified in the contract. This may include input forms, printer forms, and computer media. The cost, availability, and rights of the customer to reproduce any of the supplies should the contractor be unable to make delivery, should be included in the contract.

- Transportability

The rights of the customer to move the application system from location to location should be stated in the contract. Transportability should also include the rights of the customer to run the application system in more than one location. There may or may not be fees associated with the rights of the customer to move, duplicate, and run the application system in multiple locations.

- Termination

In the event the contractor or customer wishes to terminate the contract, the methods of terminating the contract should be specified. The termination clause should cover both cost and the return of deliverables provided under the contract. Providing for the termination can avoid a lot of bad feelings if it becomes necessary to end the contract. It also lets the organization know in advance the type of financial commitments that are associated with the termination.

- Contractor employee rights

Contractor personnel may need to visit the premises of the customer to perform service on the application system. In providing this service, the contractor needs to know if they have free access to the customer's place of business, if they can use the customer's equipment for testing (with or without charge), whether or not they can store books, manuals, supplies, in the contractor's place of business, etc. Also, if the contract is tied to usage, the contractor may wish the right to examine logs and other evidence of usage.

- Governing law

The state under which the rules of the contract are binding should be defined. It is also the laws of that state under which any dispute would be tried in court.

- Contractor inspection

The right of the contractor to look at the records of the customer should be stated. This would be necessary only in a lease arrangement if the rental is tied to usage, revenue, or other criteria based on records maintained by the customer. In order to be assured that the contractor receives full and fair rental, the contractor may wish the right to examine the customer's records. Contracts usually indicate where these are available, and the procedures necessary to obtain them.

Operating for Software Developed by Outside Organizations

Three activities occur once the software is ready for delivery. These are acceptance testing, operation and maintenance of the software, and contractual relations.

Acceptance Testing

The acceptance testing phase is necessary to verify that the contractual requirements have been met. During this phase, the customer's personnel verify that the delivered products are what the organization needs. This requires examination and testing of the deliverables.

The contractor has a responsibility to test the application to determine that it meets contractual requirements. However, the contractor will be testing against what the contractor believes to be the contractual requirements. This may or may not be what the customer needs. The customer tests against what is actually wanted.

In some cases, there will be a difference between what the user wants and what the contractor delivers. If the contract has specified the deliverables in sufficient detail, the customer will have adequate recourse for correction. However, if the contract is loosely worded, the customer may be obligated to pay all or part of additional costs necessary to meet the customer's specifications.

Ideally, the acceptance test phase occurs throughout the development process if the application is being developed for the customer. If differences can be uncovered during the development phase, correction is not costly and will usually be made by the contractor. However, problems uncovered after the application system has been developed can be costly, and may result in resistance by the contractor making the change.

If the application systems have been developed prior to contract negotiations, the user can perform acceptance testing prior to signing the contract. This is ideal from a user perspective because they know exactly what they are getting, or what modifications are needed prior to contract signing. The customer is always in a better position to negotiate changes prior to contract signing than after contract signing.

Acceptance Testing Concerns

The primary concern of the user during acceptance testing is that the deliverables meet the user requirements. The specific concerns during this phase are:

- **Meets specifications**

All of the deliverables provided by the contractor meet the user requirements as specified in the contract.
- **On time**

The delivered products will be in the hands of the user on the date specified on the contract.
- **Adequate test data**

The customer should prepare sufficient test data so that the deliverables can be adequately tested. For application systems, this may be test transactions to verify the processing performed by the application. The acceptance test criteria for training courses and manuals may be review and examination. For example, the contractor may be asked to put on a training class in order to determine the adequacy of the material.
- **Preparation for operation**

Any supporting activities should be completed in time for acceptance testing. This may involve the ordering of forms, making changes to existing operating systems, and other application systems, developing procedures to use the application, etc. The customer should have identified these needs during the feasibility phase, and worked on meeting those requirements while the contractor was preparing the application system.
- **User training**

The users of the application should be trained prior to the acceptance test phase. This training may be provided by the organization itself, or it may be done by the contractor.
- **When can it begin**

Acceptance testing should occur on the date which was specified in the contract.
- **Conversion to production**

The procedures and steps necessary to put the application into production should be tested during the acceptance testing phase. These are normally customer obligations to prepare files, schedule production, etc. These procedures should be tested just as vigorously as the contractor's application system.

Operation and Maintenance of the Software

The contractual arrangements determine the ongoing relationship between the contractor and the customer. This relationship may continue as long as the customer continues to use the application system. It encompasses continued service and maintenance. However, the ongoing relationship may only involve guarantee and warranty of the product.

Frequently organizations overlook contractual agreements after the application system has gone operational. This is because problems may not occur initially, and when they do occur the organization neglects to go back to the contract to determine the obligation of the contractor for these problems.

The major concern during the operation and maintenance of a purchased or leased application system is that both parties to the agreement comply with the contractual requirements. Contracts should be periodically reviewed to verify that the contractual requirements are being met.

Reviewers should evaluate negotiations over time to determine that the contractor fulfills their part of the contract. There are also instances where the customer is obligated to meet ongoing contractual obligations and compliance to those obligations should be verified.

Operation and Maintenance Concerns

The major concerns during the operation and maintenance of a purchased application include:

- **Adequacy of control**

Controls provided in the purchased application should be sufficient to assure that the data processed by the application is accurate, complete, and authorized. Controls should provide sufficient preventive, detective, and corrective controls to enable the organization to be assured of processing integrity. Available review checklists, such as the ones provided in previous sections of this manual, provide guidance for reviewers in making this determination.
- **Adequacy of documentation**

The application system should be maintainable and operable with the documentation provided by the contractor. If the organization is dependent upon the contractor for help in the day-to-day operations, the documentation is probably inadequate. When the user finds they cannot adequately operate or maintain the system, they should request more documentation from the contractor.
- **Speed of service**

Service should be provided by the contractor on a basis such that the operations of the organization are not seriously curtailed. In some instances, this may mean service within a few hours, while in other instances several days may be adequate.
- **Nearness of service**

The contractor should have people located such that they can service the application system in a reasonable period of time. Remote service that will be made available quickly may be adequate to satisfy this need.
- **Competency of service**

The service personnel of the contractor should be sufficiently skilled to perform the tasks for which they are assigned.

- **Adequacy of hardware**

The customer should provide sufficient hardware so that the purchased application can run in an efficient and economical mode.
- **Skilled personnel**

The customer personnel should be adequately trained in the operation of the application so that they are self-sufficient.
- **Multi-contractor problem resolution**

If application systems are provided by more than one contractor, procedures should be established and agreed upon by all of the contractors as to who has the primary problem resolution responsibility.
- **Cost of services**

The services to be provided by the contractor during the life of the contract should be specified in terms of cost. The customer should be able to anticipate the approximate cost of required service.
- **Cost of operations**

If the application system is leased, and the cost of the lease is tied to usage, the cost associated with usage should be easily measurable.
- **Error diagnosis**

The responsibility to diagnose problems should be documented. Those responsible should do the error diagnosis. If customer personnel have that responsibility, they should be sufficiently trained and have sufficient aids provided by the contractor so that they can perform this function. If the contractor personnel accept that responsibility, they must be responsive to the needs of the customer in error detection and correction.
- **Error documentation**

Procedures should be established to specify the type of documentation collected at the time of errors. This should be collected for two purposes: first, to aid contractor personnel in further diagnosis and correction of the problem; and second, as possible recourse against the contractor for recovery of fees due to extra expenses incurred. The contractor should agree that the type of documentation being collected is sufficient for their error diagnosis and correction purposes.

Contractual Relations

The relationship between the contractor and the customer is an ongoing relationship. Time and effort must be expended to keep that a viable and healthy relationship. The relationship should not be considered fixed at the point in time the contract was signed but, rather, a continual evolving relationship in which both the interest of the contractor and the customer are protected.

The contractor is anxious to sell more applications and service to the customer. Therefore, special needs and interests of the customer are normally handled even if they are above and beyond the contractual negotiations. These are normally performed in an effort to continually improve the relationship in hopes of ongoing business.

The customer hopefully has received a valuable product from the contractor. In most instances, the customer either could not produce this product, or produce it at an equivalent cost or time span. Thus, it is normally within the best interest of the customer to gain more products of a similar nature.

Contractual Relation Concerns

The concerns that arise in maintaining a relationship of harmony and good will include:

- Contractor obligations met
The contractor should meet their requirements as specified in the contract.
- Customer obligations met
The customer should meet their requirements as specified in the contract.
- Need met
It is to the benefit of both parties to have the customer satisfied with the application system. Even when the initial deliverables meet the customer's need, there will be ongoing maintenance required to meet the continually evolving needs of the customer. The methods of doing this should be specified in the contract, and those requirements should form the basis for both parties specifying and delivering new contractual obligations.
- Limits on cost increases
The cost specified in the contract should include provisions for ongoing costs. In an inflationary economy, it may be advantageous to have limits placed on cost increases. For example, if service is provided at an hourly rate, the increases in that rate might be specified in the contract.
- Exercising options (e.g., added features)
Many contracts contain options for additional features or work. When new requirements are needed, it should first be determined if they can be obtained by exercising some of the options already available in the contract.
- Renegotiation
Many contracts contain provisions to renegotiate in the event of some specified circumstances. For example, if the customer wants to extend the contract, that extension may involve a renegotiation of the terms of the contract. The renegotiation process should be conducted in accordance with the contractual specifications.
- Compensation for error

If the contractor agrees to compensate for problems due to contractor causes, the penalties should be specified in the contract.

- Returns on termination

If the contract is terminated, the contractual termination procedures should be performed in accordance with the contract requirements.

How to Take the CSQA Examination

The Introduction of this preparation guide explained a process for you to follow to prepare for the examination. It emphasized familiarizing yourself with a Common Body of Knowledge (CBOK), the vocabulary of software quality, the activities performed by quality professionals, and reviewing several other different references that are included on the software certifications Web site. Other references you might look at include articles in professional IT publications.

If you feel you are ready to take the examination, you need to schedule the examination. Be sure to visit www.softwarecertifications.org for up-to-date CSQA examination dates and places. Once scheduled, the remaining event is to take the examination.

<i>CSQA Examination Overview</i>	413
<i>Guidelines to Answer Questions</i>	414
<i>Sample CSQA Examination</i>	417

CSQA Examination Overview

The four and a half hour examination consists of four written parts, including multiple-choice and essay questions. A typical CSQA examination is comprised of two parts for Quality Assurance Theory and two parts for Quality Assurance Practice:

Quality Assurance Theory

- | | | |
|--------|--|------------------------------------|
| Part 1 | Approximately 50 multiple-choice questions | Complete within 45 minutes |
| Part 2 | Approximately 10 essay questions | Complete within 1 hour, 15 minutes |

Quality assurance theory evaluates your understanding of quality principles, practices, vocabulary and concepts. In other words, do you have a solid foundation in quality basics? For example, can you differentiate between quality assurance and quality control?

Quality Assurance Practice

Part 3 Approximately 50 multiple-choice questions Complete within 45 minutes

Part 4 Approximately 10 essay questions Complete within 1 hour, 15 minutes

Quality assurance practice evaluates whether you can apply quality basics to real-world situations. For example, a question may be: “What methods of quality control would you use to reduce defects for a software system under development? During which phase of development would you use those methods?”

You cannot bring any study or supporting materials to the examination site other than a pencil. Each individual will have an assigned seat. You may take a 10-minute break between each examination part, but not during the examination.

Proctors for the examination are not required to be certified. If not certified, they are ineligible to take the examination for at least two years after proctoring. Proctors follow specific instructions to conduct the examination. Software Certifications’ policies and procedures enforce confidentiality and security of the examination instrument, prior to and after the examination.

Guidelines to Answer Questions

The examination proctor will give you one part of the examination at a time. As you receive each part, use the following steps to answer the questions:

1. Read the entire examination part before answering any questions.
 - For multiple-choice parts, only read each question’s stem, not the four to five responses.
 - For essay parts, read all of the essay questions thoroughly.
2. As you read through each question (multiple-choice and essay) determine whether:
 - You absolutely know the answer to this question.
 - You believe you know the answer to this question.
 - You are not sure you know the answer, or it would take time to develop an answer.
3. For both multiple-choice and essay questions, answer the questions that you know the answers; they should not take you much time to complete. This will give you the majority of the examination time for the other questions you may need more time to answer.
4. Answer the questions that you believe you know the answer.
 - For multiple-choice questions, answer all of the questions.

- For essay questions, answer the questions worth the most points first, followed by those worth less points. Note that the points equal the percentage of score allocated to that essay question.
5. Answer the questions that you do not know the answer.
- For multiple-choice questions, answer all of the questions.
 - For essay questions, answer the questions worth the most points first, followed by those worth less points.

Follow these recommended guidelines for **answering essay questions**. Remember that an individual grades your examination. Make your thoughts and ideas clear and concise. Also, be sure to write legibly.

- Those questions worth the most points should be the longest essay response. For example, if an essay question is worth 25 points, the response should be at least twice as long as an essay response for a question worth 10 points.
- You need not write complete sentences. Those grading your examination are looking for key phrases and concepts. You can highlight them in a bulleted form, underlined or capitalized. This will help ensure that the individual scoring your examination can readily understand your knowledge of the correct response to that essay question.
- Charts, graphs, and examples enhance your responses and enable the individual grading your examination to evaluate your understanding of the question. For example, use a control chart example to clearly indicate your knowledge of statistical process control.

Follow these recommended guidelines for **answering multiple-choice questions**.

- Each multiple-choice question is comprised of a stem statement (or stem) and multiple responses to the stem. Read the stem carefully to assure you understand what is being asked. Then without reading the given responses to the stem, create in your mind what you believe would be the correct response. Look for that response in the list of responses.
- You will be given four or five responses to each stem.
- If you cannot create a response to the stem prior to reading the responses, attempt to eliminate those responses which are obviously wrong. In most cases, after that you will only have two responses remaining.
- To select between what appears to be correct responses, rethink the subject matter in the stem and document what you know about that subject. This type of analysis should help you to eliminate a response or select the correct response among what should be two responses.

Follow these recommended guidelines when you **do not know the answer to a question**. Usually you can take a few minutes to look over the choices or write down some key points to help you answer the questions.

- For multiple-choice questions, answer all questions – there is no reduction to your score for wrong answers. If you do not know the answer, try to rule out some of the potential responses. Then select the response you believe might be correct.

- For essay questions, indicate any information that might be helpful. For example, you have a question that deals with how quality analysts should provide criticism to their staff. You are not sure how to do this, but do remember that one of Dr. Deming's 14 points is "drive out fear." Write in your response the following: "In providing criticism you should not create a situation of "fear" with a subordinate."

Sample CSQA Examination

The following CSQA examination is a sample of some of the questions, both multiple-choice and essay, which may appear on the actual examination. Use this sample examination to help you study. The multiple-choice questions are listed by skill category. If you miss a question in a particular skill category, simply refer back to that skill category in this guide for additional study. Use the essay question responses in the sample examination as a guide for responding to the actual CSQA essay questions.

Part 1 and Part 3 Multiple-Choice Questions

This sample examination contains 20 multiple-choice questions – two questions for each category. While taking this sample examination, follow the steps described in “Guidelines to Answer Questions” on page 414 to practice the recommended techniques.

Circle the correct answer. Compare your answers to the answer key that follows on page 424.

Skill Category 1 – Quality Principles and Concepts

1. The quality attribute “interoperability” is defined as:
 - a. The effort required to couple one system to another
 - b. The extent to which a program satisfies its specifications
 - c. The amount of computing resources and code required by a program to perform a function
 - d. The effort required for learning, operating, preparing input and interpreting output of a program

2. Monies spent to train IT professionals in the concepts of quality is included in which category of the cost of quality:
 - a. Cost of production
 - b. Prevention cost
 - c. Appraisal cost
 - d. Failure cost

Skill Category 2 – Quality Leadership

3. Which of the following is considered a quality management philosophy as opposed to the more traditional management philosophies:
 - a. Competition between organizations
 - b. Motivation from fear of failure
 - c. Motivation from within (self)
 - d. Correct the error
 - e. Accomplishment from meeting quotas, the monthly or quarterly bottom line
4. In a quality management infrastructure, which of the following is a responsibility of the quality council:
 - a. Commits resources
 - b. Forms teams
 - c. Develops plans
 - d. Oversees practices
 - e. Develops processes

Skill Category 3 – Quality Baselines

5. Which of the following groups normally does not conduct an IT baseline study:
 - a. Quality assurance groups
 - b. Quality task forces
 - c. IT management
 - d. Internal auditors

6. Which of the following is not one of the five maturity levels in the SEI CMM framework:
- a. Repeatable
 - b. Testable
 - c. Defined
 - d. Managed
 - e. Optimized

Skill Category 4 – Quality Assurance

7. Which of the following is considered the most desirable skill for a successful quality assurance professional:
- a. Systems knowledge
 - b. Business system design knowledge
 - c. Project management knowledge
 - d. Verbal communications
 - e. Knowledge of computer operations
8. The quality control tool that is a technique used to quickly generate a quantity of creative or original ideas on or about a process, problem, product, or service is called:
- a. Force field analysis
 - b. Benchmarking
 - c. Quality function deployment
 - d. Playscript
 - e. Brainstorming

Skill Category 5 – Quality Planning

9. What is the first question that needs to be answered when doing quality planning:
 - a. Where do we want to go?
 - b. Where are we?
 - c. How are we going to get there?
 - d. Who is responsible for what?

10. Which is the preferred approach to quality planning:
 - a. Develop a quality plan – no updating needed during execution
 - b. Develop a quality plan – updating may be needed during execution
 - c. Ask the development team to do quality planning
 - d. Perform quality planning only when there is a quality problem
 - e. Develop a quality plan as a guideline for performing quality initiatives

Skill Category 6 – Define, Build, Implement, and Improve Work Processes

11. In the PDCA Cycle, the letter “C” stands for:
 - a. Control
 - b. Compare
 - c. Check
 - d. Continuous
 - e. Custom

12. Who within the IT organization has the responsibility for developing IT policies:
 - a. IT management
 - b. Quality assurance function
 - c. Human resource function
 - d. Internal audit function
 - e. Quality control function

Skill Category 7 – Quality Control Practices

13. Which of the following types of testing assumes that a path of logic in a unit or program is known:
- a. White-box testing
 - b. Black-box testing
 - c. Incremental testing
 - d. Thread testing
 - e. Regression testing
14. The “V” model of testing shows two development paths, which are the two sides of the “V.” These two processes are:
- a. Developmental process and test process
 - b. Test planning process and test execution process
 - c. Requirements development process and programming process
 - d. Test planning process and acceptance test planning process
 - e. System design process and system programming process

Skill Category 8 – Metrics and Measurement

15. Which of the following data types would be used for ranking data:
- a. Nominal
 - b. Ordinal
 - c. Interval
 - d. Racial

16. Which of the following metrics are used to indicate the size of a program:
- a. Number of programmers needed to build a program
 - b. Cost to build a program
 - c. Function points
 - d. Number of paths
 - e. Complexity of logic

Skill Category 9 – Internal Control and Security

17. A system of internal control in an IT organization is the responsibility of:
- a. IT management
 - b. Quality assurance function
 - c. Quality control function
 - d. Internal audit function
 - e. Project managers
18. Routines in a computer system that check the validity of input data are referred to as what type of control:
- a. Environmental control
 - b. SOX control
 - c. Preventive control
 - d. Detective control
 - e. Corrective control

Skill Category 10 – Outsourcing, COTS, and Contracting Quality

19. What is the first step an organization should do prior to acquiring COTS software:
- a. Define their requirements
 - b. Identify the cost
 - c. Develop an acceptance test plan
 - d. Evaluate ease of use
 - e. Determine reputation of COTS developer
20. In a contract to develop software, what contractual section explains the guarantees provided by the contractor:
- a. Deliverables
 - b. Vendor support
 - c. Warranty
 - d. Foreign attachment
 - e. Governing law

Part 1 and Part 3 Multiple-Choice Answers

The answers to the sample examination for Part 1 and Part 3 are as follows. If you missed a question, study that material in the relative skill category.

1. b The effort required to couple one system to another
2. b Prevention cost
3. c Motivation from within (self)
4. a Commits resources
5. d Internal auditors
6. b Testable
7. d Verbal communications
8. e Brainstorming
9. b Where are we?
10. b Develop a quality plan – updating needed during execution
11. c Check
12. a IT management
13. a White-box testing
14. a Development process and test process
15. b Ordinal
16. c Function points
17. a IT management
18. d Detective control
19. a Define their requirements
20. c Warranty

Part 2 and Part 4 Essay Questions and Answers

Theory essay questions focus on “what to do” and practice essay questions focus on “how to do it.” Together Part 2 and Part 4 have ten essay questions, one from each skill category. Five of the following essay questions are questions that could be included in Part 2, Theory Essay Questions of the CSQA examination. The other five questions are questions that could be included in Part 4, Practice Essay Questions.

Part 2 – Quality Assurance Theory Essay Questions

Answer these five essay questions following the guidelines in Guidelines to Answer Questions on page 414. Note that on the actual examination, each page has just one essay question to give you plenty of space to write your response.

Skill Category 1 – Quality Principles and Concepts

1. The producer of a product views the quality of the product differently from the customer that uses that product. First, define the producer’s view of quality, and the customer’s view of quality. Second, list the characteristics of the product produced from both the producers view and customer’s view.

Skill Category 2 – Quality Baselines

2. Compare and contrast the different models embodied in the SEI Capability Maturity Model for software and the ISO 9000 standards.

Skill Category 4 – Quality Assurance

3. The method used in your IT organization to acquire tools is two-fold. First, tool vendors send you information about new tools and/or call to explain the tools to you. Second, requests from IT staff members are an important input in determining whether or not to acquire tools. Once acquired, the determination for tool usage is left to the project and/or individual.

Your IT management has determined that many of the tools they have acquired have become “shelf-ware.” Shelf-ware meaning that after acquisition, the tools are rarely used. Your management wants a better process for tool acquisition, implementation and usage. Indicate below what you believe would be a reasonable step-by-step process for tool acquisition, implementation and use. List the steps in the sequence in which you believe they should be performed. Only a brief, but clear explanation in each step is needed.

Skill Category 6 – Define, Build, Implement, and Improve Work Processes

4. Your IT manager has been receiving a lot of complaints from the department's users/customers about late delivery of batched process computer applications. You have been assigned to identify the cause of the problem and propose a recommendation to increase the percent of on-time delivery. Your investigation shows that most of the application processing results delivered late, involved abnormal terminations. The cause of those abnormal terminations is retained on an abnormal database. The causes include:

- Allocated file space exceeded
- Operator error
- Program logic error
- Data errors
- Incorrect data entry

Following good quality principles, list below the steps you would undertake to improve on-time delivery and a description of the action taken during each step.

Skill Category 10 – Outsourcing, COTS, and Contracting Quality

5. It has been stated that approximately 50% of outsourced contracts fail to meet the customer's objectives. List and explain the differences between in-house developed software and contracted software that may be the causes for the high rate of failure of software whose development is outsourced.

Part 2 – Quality Assurance Theory Essay Questions

The following responses are examples of responses expected to receive a good grade. Review these examples as responses that adequately answer the essay question, not as the only correct response.

Essay 1.

Meeting requirements is a producer's view of quality. This is the view of the organization responsible for the project and processes, and the products and services acquired, developed, and maintained by those processes. Meeting requirements means that the person building the product does so in accordance with the requirements. Requirements can be very complete or they can be simple, but they must be defined in a measurable format, so it can be determined whether they have been met. The producer's view of quality has these four characteristics:

- *Doing the right thing*
- *Doing it the right way*
- *Doing it right the first time*
- *Doing it on time without exceeding cost*

Being fit for use is the customer's definition. The customer is the end user of the products or services. Fit for use means that the product or service meets the customer's needs regardless of the product requirements. Of the two definitions of quality, fit for use, is the more important. The customer's view of quality has these characteristics:

- *Receiving the right product for their use*
- *Being satisfied that their needs have been met*
- *Meeting their expectations*
- *Being treated with integrity, courtesy and respect*

Essay 2.*SEI CMM Model:*

- *It is a staging model.*
- *All processes have to be followed in order.*
- *Designed specifically for software organizations.*
- *It has 5 stages:*
 - *Initial – Level 1*
 - *Repeatable – Level 2*
 - *Defined – Level 3*
 - *Managed – Level 4*
 - *Optimizing – Level 5*

ISO 9000:

- *It is a continuous model.*
 - *Processes can be followed independently.*
 - *Not specific to software organizations.*
-

Essay 3.

- *Identify the need for the tool.*
 - *Identify currently available tools in the organization.*
 - *Map the need to the available tools.*
 - *If the tool satisfies the need, select, else*
 - *Identify the existing tools in the market.*
 - *Select the best possible tool to fit the need.*
 - *Introduce the tool to the team.*
 - *Conduct the awareness sessions about the tool.*
 - *Explain how it can be effectively used.*
 - *Come to a consensus that the tool should be used.*
-

Essay 4.

- *Review the current quality principles.*
- *Identify where the process went wrong by conducting the following: brainstorming session, cause-and-effect diagram.*
- *Find out the causes of the problem.*
- *Rank the causes using the nominal group technique.*
- *Draw a Pareto chart to order the causes by frequency.*
- *Take the 80% of frequency.*
- *Map it with 20% of causes.*

Essay 5.

The differences in contracted software developed by an outsourcer include:

- *Quality factors may not be specified.*

There are many factors such as reliability and ease of use which are frequently not included as part of the contractual criteria. Thus when the software is delivered it may not be as easy to use or as reliable as desired by the contractor.

- *Non-testable requirements and criteria.*

If the requirements or contractual criteria in measurable and testable terms then the delivered result may not meet the intent of the contractor.

- *Customer's standards may not be met*

Unless the contract specifies the operational standards and documentation standards the delivered product may be more complex to use than desired by the customer.

- *Missing requirements*

Unless detailed analysis and contractual specifications work is complete the contractor may realize during the development of the software that requirements are missing and thus the cost of the contract could escalate significantly.

- *Overlooked changes in standards in technology*

If changes in standards that the organization must meet, or the introduction of new desirable technology is incorporated into the contract there may be significant cost to modify the software for those new standards in technology.

- *Training and deployment may be difficult*

If software is developed by another organization there may be inadequate knowledge in the contracted organization to provide the appropriate training for staff and to ensure that deployment is effective and efficient.

Part 4 – Quality Assurance Practice Essay Questions

Answer these five essay questions following the guidelines in Guidelines to Answer Questions on page 414. Note that on the actual examination, each page has just one essay question to give you plenty of space to write your response.

Skill Category 6 – Define, Build, Implement, and Improve Work Processes

1. Unusually high failure rates are being experienced on a variety of production systems that have undergone, what were reported to be minor changes or enhancements. What should be done to diagnose and correct their situation?

Skill Category 8 – Metrics and Measurement

2. The help desk in your computer operations department receives numerous calls from IT customers about the status of their work and some questions about their work. You have asked the help desk operators to write down the topics the customers addressed in their calls to the help desk. You collected these and are now ready to do an analysis. You have been asked to do the following:

Create a Pareto chart using the process recommended for developing a Pareto chart. Do an analysis of the completed Pareto chart and describe your analysis of the information contained in that Pareto chart.

The following are the notes recorded by the help desk operators regarding discussion topics with customers:

- Terminal printer did not work.
- Do not understand instructions in customer manual.
- Computer message not clear.
- Did not understand how to make a correction.
- Could not close down terminal.
- Entered a code the software did not recognize.
- Could not find needed topic in customer manual.
- Could not activate terminal screen.
- Terminal printer did not work.
- Computer message not clear.
- Could not close down terminal.
- Could not find needed topic in the customer manual index.
- Terminal printer out of ink.
- Did not understand how to change customer account name in software system.
- Lost an hour of input data entered into the terminal.

Skill Category 2 – Quality Leadership

3. Good quality management principles state that any organization committed to quality should define its vision, values, goals and principles. For each of these:
 - a. Define the term
 - b. List who established the item (use the job title)
 - c. Give a brief example of the item

Vision

- a.
- b.
- c.

Value

- a.
- b.
- c.

Goal

- a.
- b.
- c.

Principle

- a.
- b.
- c.

Skill Category 3 – Quality Baselines

4. Once you develop a baseline for IT performance, the next step is to set an improvement goal. One method used to set an improvement goal is to benchmark your organization's baseline performance against another organization's performance. First, list the steps you would follow to benchmark your organization against another organization, and second describe the benchmarking activities in each step.

Skill Category 4 – Quality Assurance

5. One of the more effective tools is the cause-effect diagram. This diagram can be used to identify the causes that lead to a desired result. First describe how to create a cause-effect diagram, and second give an example with at least four causes that would help you get a good performance review.

Part 4 – Quality Assurance Practice Essay Answers

The following responses are examples of responses expected to receive a good grade. Review these examples as responses that adequately answer the essay question, not as the only correct response.

Essay 1.

- *Determine the type of problem.*
- *Determine the causes of the problem.*
- *Conduct a brainstorming session to find out the causes.*
- *Group the causes using the affinity diagram.*
- *Use cause-and-effect analysis to determine the root cause.*
- *Order the causes using the Pareto chart.*
- *Once the causes are determined, find the type: common cause or special cause.*
- *If mean and standard deviation remains constant over the period it is common cause.*
- *Special cause cannot be controlled.*
- *Take all the common and draw control charts.*
- *Once the common causes are reduced, draw control charts to verify.*

Essay 2.

- *Do a brainstorming session to determine the causes.*
- *Rank the causes using nominal group technique.*
- *Determine the frequency of occurrence for each cause.*
- *Draw a chart with frequency versus causes.*
- *Order the chart according to frequency.*
- *Compute the % of occurrence.*
- *Apply 80-20 rule.*
- *80% of frequency will affect 20% of causes.*
- *Find 20% of causes.*
- *Develop solutions to fix the causes.*

Essay 3.**Vision:**

- a. *It states what the organization intends to achieve.*
- b. *Senior Management (CIO)*
- c. *“To become Global Top 10”*

Value:

- a. *It states how to run the business.*
- b. *Senior Management*
- c. *“Will satisfy customer needs”*

Goal:

- a. *It states how the vision will be achieved.*
- b. *Senior Management*
- c. *“Establish a customer vision group”*

Principle:

- a. *It states the quality attribute of the organization.*
- b. *Quality council*
- c. *“Need for a quality function”*

Essay 4.

There are many different steps that organizations follow in benchmarking. However, most baselining processes have these four steps:

1. *Develop a clearly defined baseline in your organization.*

This means that all of the attributes involved in your baseline are defined. In our example of defects per lines of code, clearly defining what is meant by defect and a line of code would meet the objective of this step.

2. *Identify the organization you desire to baseline against.*

Many factors come into this decision, such as do you want to benchmark within your industry, do you want to benchmark what you believe are leading organizations, do you want to benchmark an organization that uses the same tools that are used in your organization, and do you want to benchmark against organizations with a similar culture.

3. *Compare baseline calculations.*

Compare how your baseline is calculated versus the baseline calculation in the company you want to benchmark against. Benchmarking is only effective when you benchmark against an organization who has calculated their baseline using approximately the same approach that your organization used to calculate the baseline.

4. *Identify the cause of baseline variance in the organization you benchmarked against.*

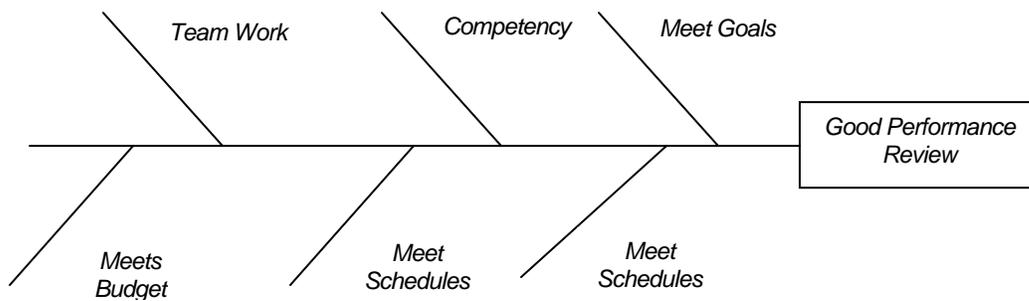
When you find a variance between the baseline calculation in your company and the baseline calculation in the organization you are benchmarking against, you need to identify the cause of variance. For example, if your organization was producing 20 defects per thousand lines of code, and you benchmarked against an organization that only had 10 defects per thousand lines of code you would want to identify the cause of the difference. If you cannot identify the cause of the difference, there is little value in benchmarking. Let us assume that the company you benchmarked against had a different process for requirement definition than your organization. For example, assume they use JAD (joint application development) and you did not. Learning this, you may choose to adopt JAD in your organization as a means for reducing your developmental defects rate.

Essay 5.

Developing a cause-and-effect diagram requires this series of steps:

1. *Identify a problem (effect) with a list of potential causes. This may result from a brainstorming session.*
2. *Write the effect at the right side of the paper.*
3. *Identify major causes of the problem, which become “big branches”. Six categories of causes are often used: Measurement, Methods, Materials, Machines, People, and Environment, but the categories vary with the effect selected.*
4. *Fill in the “small branches” with sub-causes of each major cause until the lowest-level sub-cause is identified.*
5. *Review the completed cause-and-effect diagram with the work process to verify that these causes (factors) do affect the problem being resolved.*
6. *Work on the most important causes first. Teams may opt to use the nominal group technique or Pareto analysis to reach consensus.*
7. *Verify the root causes by collecting appropriate data (sampling) to validate a relationship to the problem.*
8. *Continue this process to identify all validated causes, and, ultimately the root cause.*

Example of a cause-effect diagram:



Vocabulary

Acceptable Quality Level (AQL)

ANSI/ASQC Z1.4-1981 defines AQL as “the maximum percent nonconforming (or the maximum number of nonconformities per hundred units) that, for purposes of sampling inspection, can be considered satisfactory as a process average.”

Acceptance Testing

Testing to ensure that the system meets the needs of the organization and the end user or customer (i.e., validates that the right system was built).

Access Modeling

Used to verify that data requirements (represented in the form of an entity-relationship diagram) support the data demands of process requirements (represented in data flow diagrams and process specifications.)

Activity

An identifiable work task that needs to be controlled.

Affinity Diagram

A group process that takes large amounts of language data, such as a list developed by brainstorming, and divides it into categories.

ANSI

The American National Standards Institute that is the organization that helps set standards and also represents the United States in international standards bodies.

Audience Evaluation

The ability to evaluate audience needs and develop appropriate presentation materials.

Audit

An independent inspection or assessment activity that verifies compliance with plans, policies, and procedures, and ensures that resources are conserved. Audit is a staff function; it serves as the "eyes and ears" of management.

Backlog

Work waiting to be done; for IT this includes new systems to be developed and enhancements to be made to existing systems. To be included in the development backlog, the work must have been cost-justified and approved for development.

Baseline

A quantitative measure of the current level of performance.

Benchmark

An industry or best-of-class norm.

Benchmarking

Searches for the best practices or competitive practices that will help define superior performance of a product, service, or support process.

Black-Box Testing

Data driven testing that focuses on evaluating the function of a program against its specifications.

Brainstorming

A group process for generating creative and diverse ideas.

Cause-and-Effect (Fishbone) Diagram

A tool used to identify possible causes of a problem by representing the relationship between some effect and its possible causes.

Celebrating

A group activity in which a team's success is made known publicly and praised. This may include tangible rewards such as refreshments and award certificates.

Charter

A document defining the formal organization of a corporate body: a constitution. Authorization from a central or parent organization to establish a new branch, chapter, etc.

Check Sheet

A form used to record data as it is gathered.

Client

The customer that pays for the product received, and receives the benefit from the use of the product.

Coaching

Providing advice and encouragement to an individual or individuals to promote a desired behavior.

Computer Society

A constituent society (or technical component) of the IEEE.

Conflict Resolution

The process of bringing a situation into focus and satisfactorily reducing or eliminating a disagreement or difference between parties.

Contributor Measure

A unit of measure by which a result measure is controlled. Contributor measures do not impact the customer directly, but contribute to the success of the result.

Control Chart

A statistical method for distinguishing between common and special cause variation exhibited by processes.

Control Unit

A critical success factor that must be managed to achieve the success of a goal, policy, or strategy.

Cost of Quality (COQ)

Money spent above and beyond expected production costs (labor, materials, or equipment) to ensure that the product the customer receives is defect-free. This includes the cost of repairing a defective product that was shipped to a customer and the associated damage costs.

Customer

The individual or organization, internal or external to the producing organization, that procures the product.

Customer-Recorded Impacts

The positive and negative effects upon the customer resulting from IT actions.

Dashboard

An aggregation of measures, together with their standards, that provides a quantitative analysis of critical components of a process.

Defect

From the producer's viewpoint, a defect is a product requirement that has not been met, or a product attribute possessed by a product or a function performed by a product that is not in the statement of requirements that define the product. From the customer's viewpoint, a defect is anything that causes customer dissatisfaction, whether in the statement of requirements or not.

Defect Rate

Any relationship between identified measures that indicates, to the metric users, a level of quality.

Documentation Structuring

Designing documents that are clearly, logically, meaningfully, and comprehensively laid out.

Dynamic Testing

Testing which involves executing the system's code.

Effective Listening

Actively listening to what is said by asking for clarification when needed, and providing feedback statements on what was said to reinforce understanding and acknowledge that listening is occurring.

Effective Presentation

Providing or teaching information in a manner that transfers understanding and is appropriate to the audience. The proper use and value of videos, slides, overheads, flipcharts, handouts, brochures, and PC projections are examples of common tools and should be understood.

EIA

The Electronics Industry Association, which is an ANSI-accredited standards developer. The EIA is the national trade organization that represents United States electronics manufacturers.

Empowerment

Giving people the knowledge, skills, and authority to act within their area of expertise to do the work and also improve the process.

Exception Reporting

The process of reporting only significant variances from what was expected.

Facilitation

The process of helping the progress of some event or activity. An understanding of formal facilitation includes well-defined roles, objectivity of the facilitator, a structured meeting, decision-making by consensus, and defined goals to be achieved.

First Party Audit

An internal audit conducted by auditors who are employed by the organization being audited, but who have no vested interest in the audit results.

Flowchart

A diagram that shows the sequential steps of a process or of a workflow around a product or service.

Force Field Analysis

A group technique used to identify both driving and restraining forces that influence a current situation.

Functional Tests

Tests of business requirements that address the overall behavior of the system.

Gainsharing

Sharing in the savings from quality improvement efforts.

Histogram

A graphical description of individual measured values in a data set that is organized according to the frequency or relative frequency of occurrence. A histogram illustrates the shape of the distribution of individual values in the data set along with information regarding the average and variation.

IEC

The International Electrotechnical Commission.

IEEE

The Institute of Electrical and Electronics Engineers, which is the world's largest technical professional society with members in almost 150 countries.

Influencing Skills

Capabilities and techniques developed to cause one person to have a certain planned effect on another.

Information Systems (IS)

This is a general term for the computer systems in an organization that provides information about its business operations. It's also used to refer to the people who manage these systems. Typically, in a large corporation, "IS" or the "IS department" refers to a central or centrally-coordinated system of computer expertise and management, often including the organization's entire network of computer resources.

Information Technology (IT)

Any activity (not limited to systems development) that uses information to fulfill its mission. Also called information services, management information services, and information systems.

Inputs

Products, services, or information needed from suppliers to make a process work.

Integration Testing

Testing performed on groups of modules to ensure that data and control are passed properly between modules.

International Organization for Standardization (ISO)

ISO is a network of the national standards institutes of 156 countries, on the basis of one member per country, with a Central Secretariat in Geneva, Switzerland, that coordinates the system. ISO is a non-governmental organization: its members are not, as is the case in the United Nations system, delegations of national governments.

Interpersonal Effectiveness

The ability to work and negotiate effectively with personnel of different professions, skill levels, organizational levels and varying experience.

Interviewing

Developing and asking questions for the purpose of collecting oral data to be used in an analysis or evaluation.

Joint Application Development (JAD) Session

A meeting where the producer and customer come together to negotiate and agree upon requirements.

Just-in-Time (JIT)

The system known as "the Toyota production system" that has set the standard for world-class manufacturing. The ultimate goal of JIT production is to supply each

process with exactly the required items, in exactly the required quantity, at exactly the required time.

Key Result Areas

Broad-based areas of performance that, when measured, give a unit an evaluation of its critical customer-driven processes.

Leadership

The ability to guide or influence a group to move in some direction, including inspiring others in a shared vision of what can be, taking risks, serving as a role model, reinforcing and rewarding the accomplishments of others, and helping others to act.

Level of Service

The average performance received by the customer per criterion over a given period of time, which is normally 30 days.

Likert Scale

A way to collect measures, typically on a survey. A Likert Scale contains categories such as Very Satisfied, Slightly Satisfied, Satisfied, Slightly Dissatisfied, and Very Dissatisfied.

Maintenance

The process of modifying errors found in a released software product.

Management

A team or individual that manages resources at any level of the organization. Management obtains results through the efforts of other people. It is everyone in the organization except those in positions specifically designated as non-management.

Management by Fact

The process of using qualitative and quantitative data produced from and about work processes to make informed decisions regarding the operation of those work processes. The two components of this process are meeting desired results, and managing the processes to drive the results.

Management by Process

The use of processes to achieve management's desired results.

Mean

A number that represents a set of numbers in any of several ways determined by a rule involving all members of the set: AVERAGE.

Measure

A single attribute of an entity; a standard unit.

Meeting Management

The process of organizing and conducting meetings to provide maximum productivity over the shortest time period.

Metric

Two or more measures combined in a relationship to each other to produce information about an entity.

Metric Name

A short name or expression that conveys the intent of the quality characteristic.

Mission

A customer-oriented statement of purpose for a unit or a team.

Negotiation

The process of working together with one or more parties to establish goals to be reached, create options that will satisfy all parties, and select an option that is best for all parties. May utilize skills such as compromising and consensus building.

Objective Performance Measurement

A quantifiable means of measuring the level of service received, such as the number of times reports were late during the month.

Outputs

Products, services, or information that results from a process.

Perfective Maintenance

The act of improving the product at the same time a fix is made. This is risky and not recommended.

Performance Criteria

Major performance related categories, such as accuracy, quality, or timeliness.

Performance Standards

Defined, measurable levels of IT service applicable to an individual customer organization or a group of customer organizations.

Perspective

The point of view from which assessments or customer satisfaction and quality can be made. Examples include the customer's view and the provider's view of quality.

Policy

Managerial desires and intents concerning either processes (intended objectives) or products (desired attributes).

Problem

Any deviation from defined standards. Same as a defect.

Problem Reporting/Tracking

The process of reporting outstanding problems; having them assigned for resolution, and closing them out when the customer has been notified that the problems have been solved.

Procedure

The step-by-step method followed to ensure that standards are met.

Process

(1) The work effort that produces a product. This includes efforts of people and equipment guided by policies, standards, and procedures. (2) A statement of purpose and an essential set of practices (activities) that address that purpose. A process or set of processes is used by an organization or project to plan, manage, execute, monitor, control, and improve its software related activities.

Process Definition

A description of the organization's current best practice approaches so that the process is understood, consistently performed, and ready for improvement or redesign.

Process Improvement

The progression of steps taken to change a process to make it produce a given product faster, more economically, or of higher quality. Such changes may require the product to be changed. It involves improving process capability and reducing variation by analyzing the process and product results, identifying root-cause problems, and changing the process to eliminate root causes. The defect rate must be maintained or reduced.

Process Re-engineering

The fundamental rethinking and radical redesign of business processes.

Product

The output of a process: the work product. There are three useful classes of products: Manufactured Products (standard and custom), Administrative or Information Products (invoices, letters, etc.), and Service Products (physical, intellectual, physiological, and psychological). Products are defined by a statement of requirements.

Product or Service

Something produced or provided to meet the customer's requirement.

Product Improvement

Changing the statement of requirements that defines a product to make the product more satisfying and attractive to the customer (more competitive). Such changes may add to, or delete from, the list of attributes or the list of functions defining a product. Such changes frequently require the process to be changed. This process could result in a totally new product.

Production Costs

The cost of producing a product. Production costs, as currently reported, consist of (at least) two parts; actual production or right-the-first time costs (RFT) plus the Cost of Quality. RFT costs include labor, materials, and equipment needed to provide the product RFT.

Productivity

The ratio of the output of a process to the input, usually measured in the same units. It is frequently useful to compare the value added to a product by a process, to the value of the input resources required (using fair market values for both input and output).

Productivity Metric

The ratio of work product (i.e., function points) divided by work effort (i.e., staff days).

Quality

Operationally, the word quality refers to products. A product is a quality product if it is defect-free. To the producer, a product is a quality product if it meets or conforms to the statement of requirements that defines the product. This statement is usually shortened to: quality means meets requirements. To the customer, a product is a quality product if it meets the customer's needs, regardless of whether the requirements were met. This is referred to as fit for use.

Quality – Producer View

The producer's view of quality has these four characteristics: Doing the right thing, Doing it the right way, Doing it right the first time, and Doing it on time without exceeding cost.

Quality – Customer View

Meeting requirements is a producer's view of quality. This is the view of the organization responsible for the project and processes, and the products and services acquired, developed, and maintained by those processes.

Quality Assurance (QA)

The set of support activities (including facilitation, training, measurement, and analysis) needed to provide adequate confidence that processes are established and continuously improved to produce products that meet specifications and are fit for use.

Quality Control (QC)

The process by which product quality is compared with applicable standards, and the action taken when a nonconformance is detected. It focuses on defect detection and removal. This is a line function - performance of these tasks is the responsibility of the people working within the process.

Quality Function Deployment

A systematic matrix method used to translate customer wants or needs into product or service characteristics that will have a significant positive impact on meeting customer demands.

Quality Improvement

The change to a production process so that the rate at which defective products (defects) are produced is reduced. Some process changes may require the product to be changed.

Quality Improvement Plans (Action Plans)

Plans developed to correct identified problems.

Quality Management (QM)

(1) A philosophy consisting of continuous process improvement activities involving everyone in the organization in an integrated effort to improve performance at every level. It requires the commitment of executive management, and an empowerment of employees at all levels that enables them to participate in the improvement of the processes that create products and services. (2) Quality management integrates fundamental management techniques, existing improvement efforts, teamwork, and technical tools in a disciplined approach

focused on continuous process improvement. It is also called total customer focus, total quality control, quality assurance, and a leadership program.

Quality Measure

A quantitative assessment of the extent that a product or service demonstrates successful performance, conforms to its requirements, or possesses a given attribute.

Quality Metric

Any relationship between identified measures that indicates, to the metric users, a level of quality.

Quality Professional

The individual or group who assists IT management in improving quality, productivity, and customer satisfaction. Other names used for this, depending on specific assignments, include quality function, quality management coordinator, quality assurance, quality consultant, quality control, quality analyst, and QA analyst.

RAD

Rapid Application Development.

Regression Testing

Testing after changes have been made to ensure that no unwanted changes were introduced.

Reliable Measure

A reliable measure is one that: a) if the measure were to be taken again, the result would be the same; and b) if two or more different people developed the same measure, they would produce the same results.

Reporting Frequency

How often a particular report is developed and distributed to its audience.

Requirement

A formal statement of: 1) an attribute to be possessed by the product or a function to be performed by the product; 2) the performance standard for the attribute or function; or 3) the measuring process to be used in verifying that the standard has been met.

Result Measure

A critical success factor or value that must be controlled through measurement. Result measures directly impact the customer.

Run Chart

A graph of data points in chronological order used to illustrate trends or cycles of the characteristic being measured to suggest an assignable cause rather than random variation.

Sampling

Looking at a small number of work products or a section of the work product rather than at each product.

Scatter Plot (Correlation Diagram)

A graph designed to show whether there is a relationship between two changing factors.

Second Party Audit

An external audit performed on a supplier, by a customer or a contracted (consulting) organization on behalf of the customer.

Service-Level Objectives

A published level of service that information technology will provide customers by performance criterion.

Services

See Product.

Stakeholders

Individuals who have a vested interest in the success or failure of a quality initiative.

Standard

A requirement of a product or process. For example: 100 percent of the functionality must be tested.

Standardize

The implementation of procedures to ensure that the output of a process is maintained at a desired level.

Statement of Requirements

The exhaustive list of requirements that define a product. The Statement of Requirements should document requirements proposed and rejected (including the reason for the rejection) during the requirement determination process.

Static Testing

Testing performed without executing the system's code; can be manual (e.g., reviews) or automated (e.g., code or writing analyzers).

Statistical Process Control

The use of statistical techniques and tools to measure an ongoing process for change or stability.

Structural Tests

Tests that require knowledge of the internal logic of a system.

Subjective Performance Measurement

A person's perception of a product or activity, including personal attitudes, feelings, and opinions, such as how user-friendly the application is. Different people may measure different values for the same item because of their subjective judgment.

Supplier

An individual or organization that provides the inputs needed to generate a product, service, or information to a customer.

System Testing

1) A generic term that differentiates various types of higher order testing from unit testing; 2) a predetermined combination of tests that, when executed successfully, satisfy IT management that the system meets requirements.

Taxonomy

Categorization of items for understanding and use.

Team Building

The process of aiding a group to define a common goal and work together towards that goal.

Third Party Audit

An external audit performed on a supplier by an external participant other than the customer.

Unit Testing

Testing performed on a single, stand-alone module or unit of code.

User

The customer that actually uses the product received.

Validation

Any activity that helps assure that the end product (e.g., system) under defined operating conditions meets its currently approved requirements and expectations.

Verification

All QC activities throughout the life cycle that assure that interim product deliverables process their inputs in accordance with specifications and standards.

Vision

A statement that describes the desired future state of a unit.

White-Box Testing

Testing based on knowledge of internal code structure and logic.

References

It is each candidate's responsibility to stay current in the field and to be aware of published works and materials available for professional study and development. Software Certifications recommends that candidates for certification continually research and stay aware of current literature and trends in the field. There are many valuable references that have not been listed here. These references are for informational purposes only.

Clements, Richard Barrett. Quality Manager's Complete Guide to ISO 9000. Prentice Hall, First Edition, 1993.

Daughtrey, Taz. Fundamental Concepts for the Software Quality Engineer. ASQ Quality Press, 2001.

Dobbins, James H. Software Quality Assurance and Evaluation. American Society for Quality, Quality Press, 1990.

Donaldson, Scott E. and Stanley G. Siegel. Cultivating Successful Development: A Practitioner's View. Prentice-Hall, 1997.

Fenton, Norman & Robin Whitty & Yoshinori Lizuka. Software Quality Assurance and Measurement: Worldwide Perspective. International Thomson Computer Press, 1995.

Galín, Daniel. Software Quality Assurance: From Theory to Implementation. Addison Wesley, 2003.

Gao, Jerry Zeyu, et al. Testing and Quality Assurance for Component-Based Software. Artech House, Inc., 2003.

Ginac, Frank P. Customer Oriented Software Quality Assurance. Prentice Hall PTR, First Edition, 1997.

Godbole, Nina S. Software Quality Assurance: Principles and Practice. Alpha Science International, Ltd, 2004.

Goodman, Paul. Practical Implementation of Software Metrics. McGraw-Hill Companies, 1993.

- Holdsworth, Jacqueline. Software Process Design: Out of the Tar Pit. McGraw-Hill Companies, 1994.
- Horch, John W. Practical Guide to Software Quality Management. Artech House, 1996.
- Hutcheson, Marnie L. Software Testing Fundamentals: Methods and Metrics. John Wiley & Sons, Inc., 2003.
- Ince, Darrel. ISO 9001 and Software Quality Assurance. McGraw-Hill Companies, 1994.
- Institution of Electrical Engineers. Software Quality Assurance: Model Procedures (Computing Series). Institution of Electrical Engineers, 1999.
- Jarvis, Alka and Vern Crandall. Inroads to Software Quality: "How To" Guide and Toolkit. Prentice-Hall, 1997.
- Jenner, Michael G. Software Quality Management and ISO 9001: How to Make Them Work for You. Wiley & Sons, Inc., First Edition, 1995.
- Juran, J. M. Juran on Quality by Design: The New Steps for Planning Quality Into Goods and Services. The Free Press, 1992.
- Kan, Stephen H. Metrics and Models in Software Quality Engineering (Second Edition). Addison-Wesley, 2002.
- Kitchenham, B. A., & B. Littlewood. Measurement for Software Control and Assurance. Elsevier Science Publishers Co., 1990.
- Perry, William E. Effective Methods for Software Testing (Second Edition). John Wiley & Sons, Inc., 2000.
- Perry, William E. Quality Assurance for Information Systems: Methods, Tools, and Techniques. John Wiley & Sons, 1991.
- Sanders, Joc and Eugene Curran. Software Quality: A Framework for Success in Software Development and Support. Addison-Wesley, 1994.
- Schulmeyer, G. Gordon & James I. McManus. The Handbook of Software Quality Assurance (Third Edition). Prentice Hall PTR, 1999.
- Tian, Jeff. Software Quality Engineering: Testing, Quality Assurance, and Quantitative Improvement. Wiley-Interscience, 2005.
- Wieggers, Karl E. Software Requirements. Microsoft Press, Second edition, 2003.