Hans-Ludwig Hausen

GMD German National Research Center for Information Technology
EMail: hausen@gmd.de
Mail: GMD; Schloss Birlinghoven; D-53754 Sankt Augustin; Germany
Tel: +49-2241-14-2937___<>___home: +49-2251-51-998______
Fax: +49-2241-14-2197___<>___or___ +49-2241-14-2618_____
WWW: http://www.scope.gmd.de <> ftp://ftp.gmd.de/GMD/SW-Quality

__ Software Process Programming and Testing (Y7, V-Model, ProcePT) __
__ Guides to Software Evaluation and Certification (HAWE, SCOPE) __

Quality is to be defined, measured and assessed with respect to
the extent to which stated or implied requirements are met !!!

uct:

oftware comprising at least
quirements, specifications and
rogram(s)

ess:

anned, controlled and reported
ctions to construct, apply or
aintain software product

ect:

aned, controlled and reported
rocess and product

Assessment:

comparing actual measurement results agains
required

Certification:

checking conditions and eventually issuing a
certificate

Measurement:

mapping of an attribute onto real numbers

Validation:

test against implied needs i.e. assumptions

Verification:

test against stated needs i.e. specifications

Introduction

Software Process and Software Product

Process Evaluation and Certification
CMM, ISO9000, TickIT, Trillium, ami, SPI

Product Evaluation and Certification
Evaluators Guide according ISO9126

tion = verification + validation + measurement + assessment

ment of software

ess of comparing the values obtained from the measurements with quality requirements.

ied software

ware which is classified according to product, process and supportive information or other keys.

on or institution (e.g. producer, distributor, buyer, or user) who negotiates the evaluation.

tion module

psulation of the definition of an evaluation (sub-) method applied on product or process information in orde sure software characteristics or subcharacteristics by applying metrics, checking pass-fail criteria, delivering uation report and cost report.

tion level

rade which is defined by a set of evaluation techniques to be applied and the thresholds of quality metrics be ined by those techniques.

entification of (subcharacteristics and) metrics and attachment of metrics to subcharacteristics and definitior ptance criteria by selecting rating levels for each metric and reference to (sub-) evaluation method to be app

ic Terms II

Hans-Ludwig Hausen

on report

ocument of the software evaluation. It is filled up through the whole evaluation process and consists of four ion requirement, evaluation specification, evaluation plan and evaluation result.

on item

being evaluated.

d software

re which is identified by document identifier, title, condition, and of date of arrival as well as handling infor

ment

ation of a metric for product quality or process productivity.

information

s obtained during the software process.

information

s constituting a software product or one or more parts of it.

e evaluation

s which comprises validation and verification, measurement and assessment of software.

ve information

s which are not evaluated but which are necessary for an evaluation.

E ENTIRE POPULATION PROGRAMS

the early days of the telephone, they were employing nany young women to act as telephone operators. Someone calculated that, at then current growth rates, he number of operators required would quickly reach he entire population.

e solution, of course, was to make the entire populatio ecome operators. Every time you dial a telephone, yo re acting as your own operator.

timately, I don't know if we can do this with software, vhich is substantially more complex than the user-nterface for the telephone.

course, we can try to generate application

vastes 38 Million GB£ on Military Satellite Trackin
ı', ***Daily Express Report (21/10/94)***

- 'The specification did not reflect the true scope of
hat was needed'

n Disasters can be Avoided', ***Computer Weekly Report (12/***

)- 'Study showed that 44.1% of all system faults occ
specification stage'

ɔmpanies spend over 1 Billion GBP per year on Sof
'opriate to their Needs', ***Computing Report (16/11/95)***

- 'Study claimed that systems do not perform as int

nderstanding in systems requirements on the part of customers a

ps between estimates of costs and time with actual expenditures.

riations in programmer productivity levels.

y in dividing labor between design and production coding....

y in monitoring progress in a software project, sice program const a simple progression in which each act of assembly represents a c p.

owth in size of software systems.

munication among groups working on the same project, exacerba ordinated or unnecessary information, and a lack of automation to nformation.

pense of developing on-line production control tools

/ of measuring key aspects of prgrammers and system performan

n among software developers of not writing systems for practica
ite new and better systems... makes it difficult to predict and mar

wth in the need for programmers and insufficient numbers of ad
skilled programmers.

/ of achieving sufficient reliability ... in large software systems.

nce of software on hardware, which makes standardization of sof
oss different machines.

nventories of reusable software components to aid in the building

maintenance costs often exceeding the cost of original system
nt.

m List in Software Development (In 1968 NATO /Naur91/ formulated fifteen difficulties in developing lar

nswers the question:***Where are we going?"*** or
What do we want to be when we gr

nswers the question:***What are my guiding principles?"*** o
What will I do or not do to achieve my

Answers the question: ***Why do we do what we do or plan***

Answers the question: ***What do we do to achieve***
the vision in the short and long te

Answers the question:***What are the enabling approache***
to ensure achievement of the missi
in light of our vision, values and pu

Answers the questi

Answers the question:

hat are our overall
visions, values, purposes, objectives, strategies, and tacti

Answers the question:

hat are the artefacts and sub-artefacts to be considered and
what are the relations amongs

Answers the question:

hat are the actions and sub-actions to be considered and
what are the relations amongst them?

Answers the question:

hat are the methods to be used
w.r.t. goals, products and processes?

Answers the question:

hat are the tools to be used w.r.t. to the other problem domains

Answers the question:

hat are the decission relevant charachteristics of products and

Quality is free if build in.

n-Quality will impose lifetime supp

lity costs, Bad-Quality costs even m

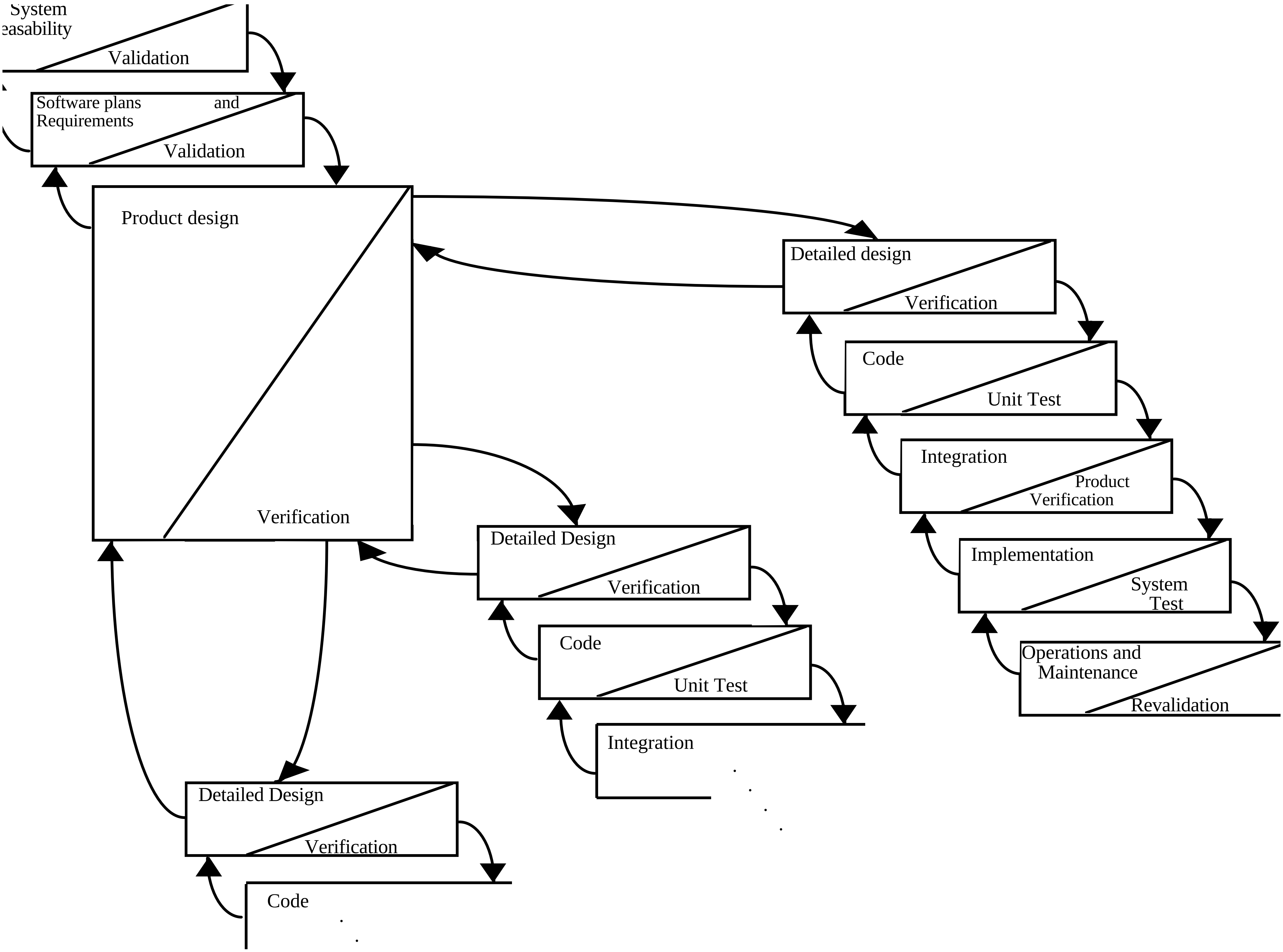
And the trouble is,

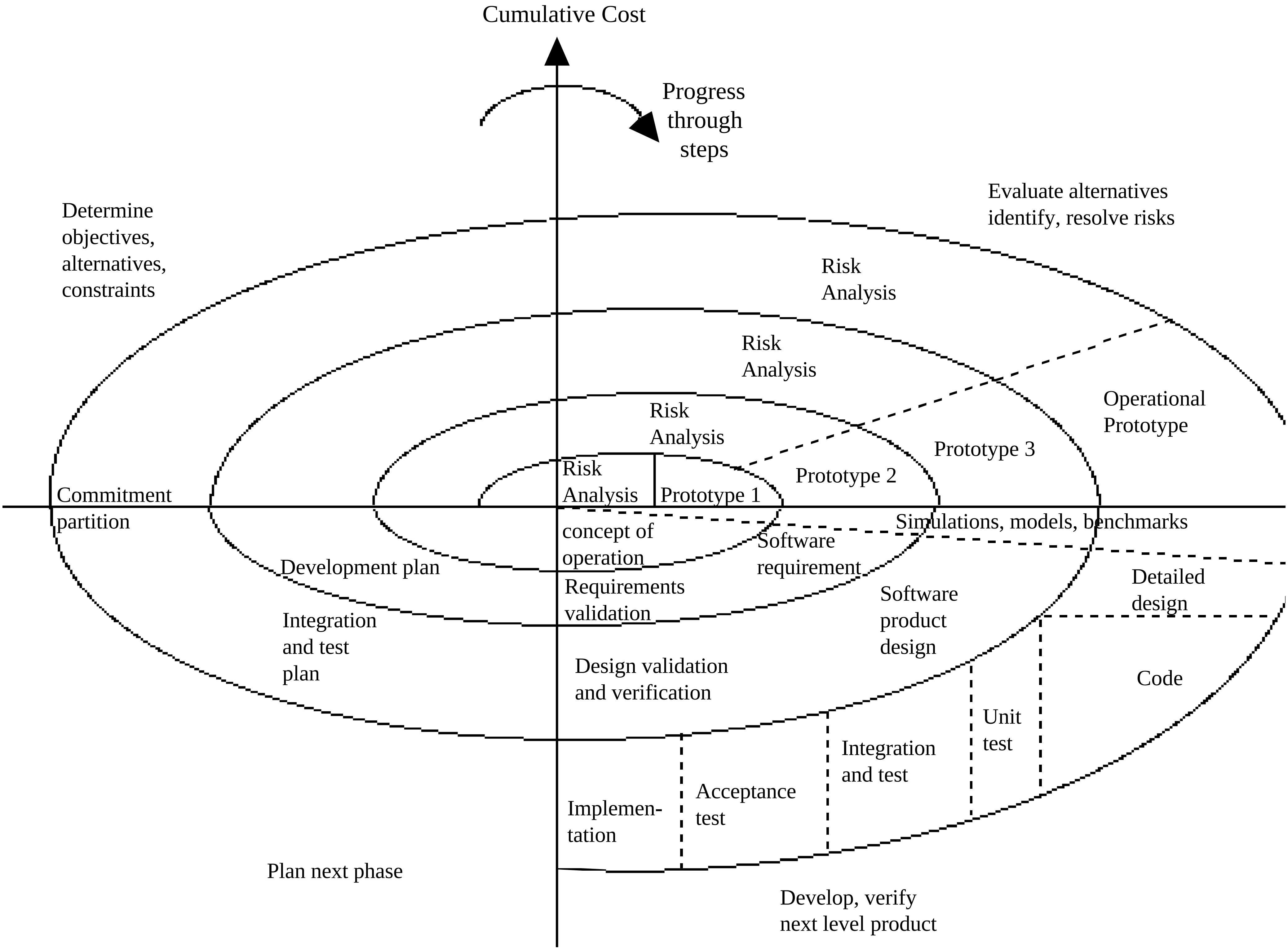
don't risk anything, you risk even

<u>Quality Assurance</u> of system in terms of
validation,
verification,
test,
measurement,
and assessment
of acts
artefacts,
and
states

s

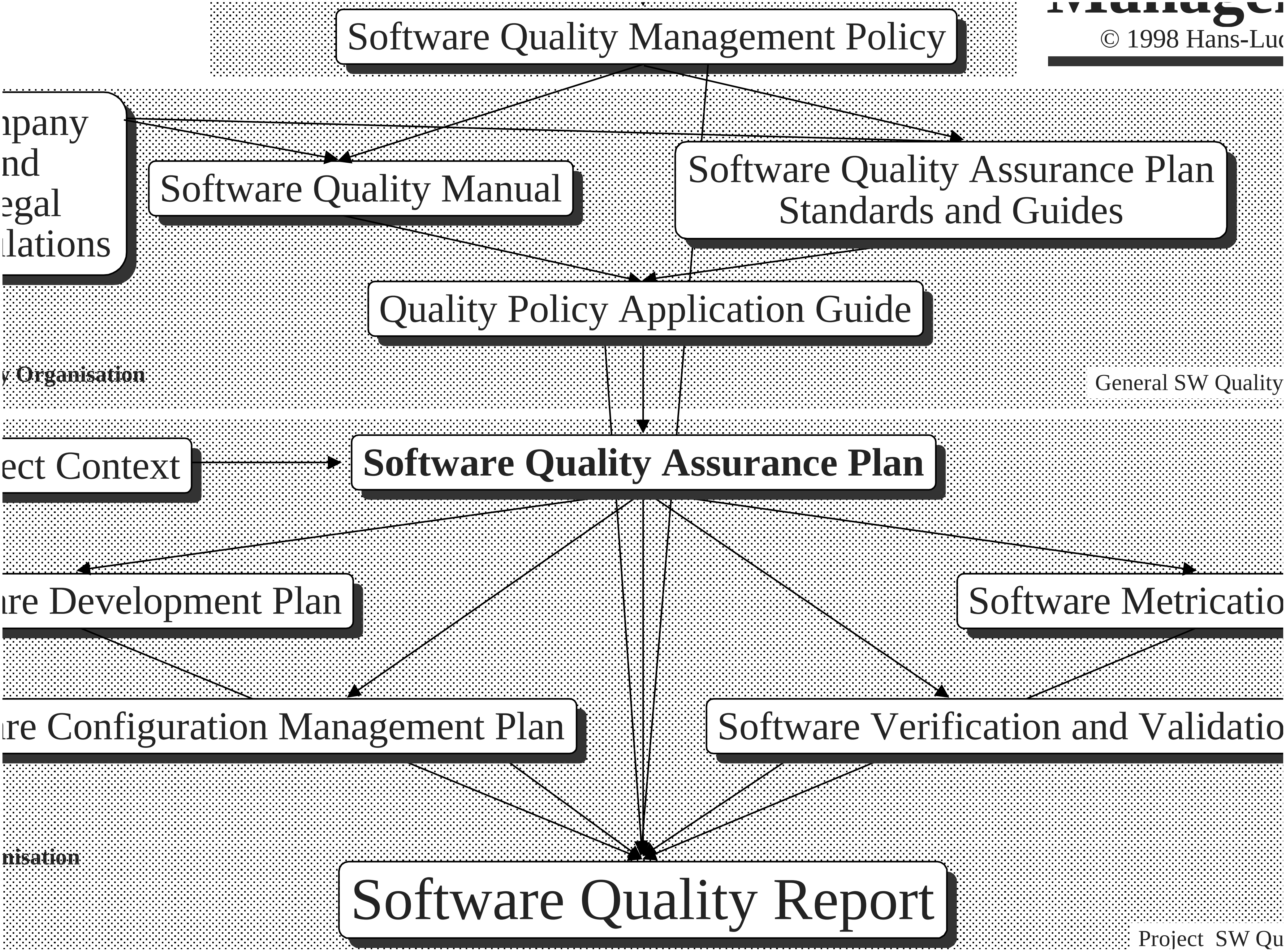
nfiguration of
application system and
information processing system


System
easability
Validation
Software plans and
Requirements
Validation
Product design
Verification
Detailed design
Verification
Code
Unit Test
Integration
Product
Verification
Implementation
System
Test
Operations and
Maintenance
Revalidation
Detailed Design
Verification
Code
Unit Test
Integration
.
.
.
.
Detailed Design
Verification
Code
.
.


Cumulative Cost
Progress through steps
Determine objectives, alternatives, constraints
Evaluate alternatives identify, resolve risks
Risk Analysis
Risk Analysis
Risk Analysis
Risk Analysis
Operational Prototype
Prototype 3
Prototype 2
Prototype 1
Commitment partition
Simulations, models, benchmarks
concept of operation
Software requirement
Development plan
Requirements validation
Detailed design
Software product design
Integration and test plan
Design validation and verification
Code
Unit test
Integration and test
Acceptance test
Implementation
Plan next phase
Develop, verify next level product

Manage

© 1998 Hans-Luc



Software Quality Management Policy

ıpany
nd
egal
ılations

Software Quality Manual

Software Quality Assurance Plan
Standards and Guides

Quality Policy Application Guide

y Organisation

General SW Quality

ect Context

Software Quality Assurance Plan

ıre Development Plan

Software Metricatio

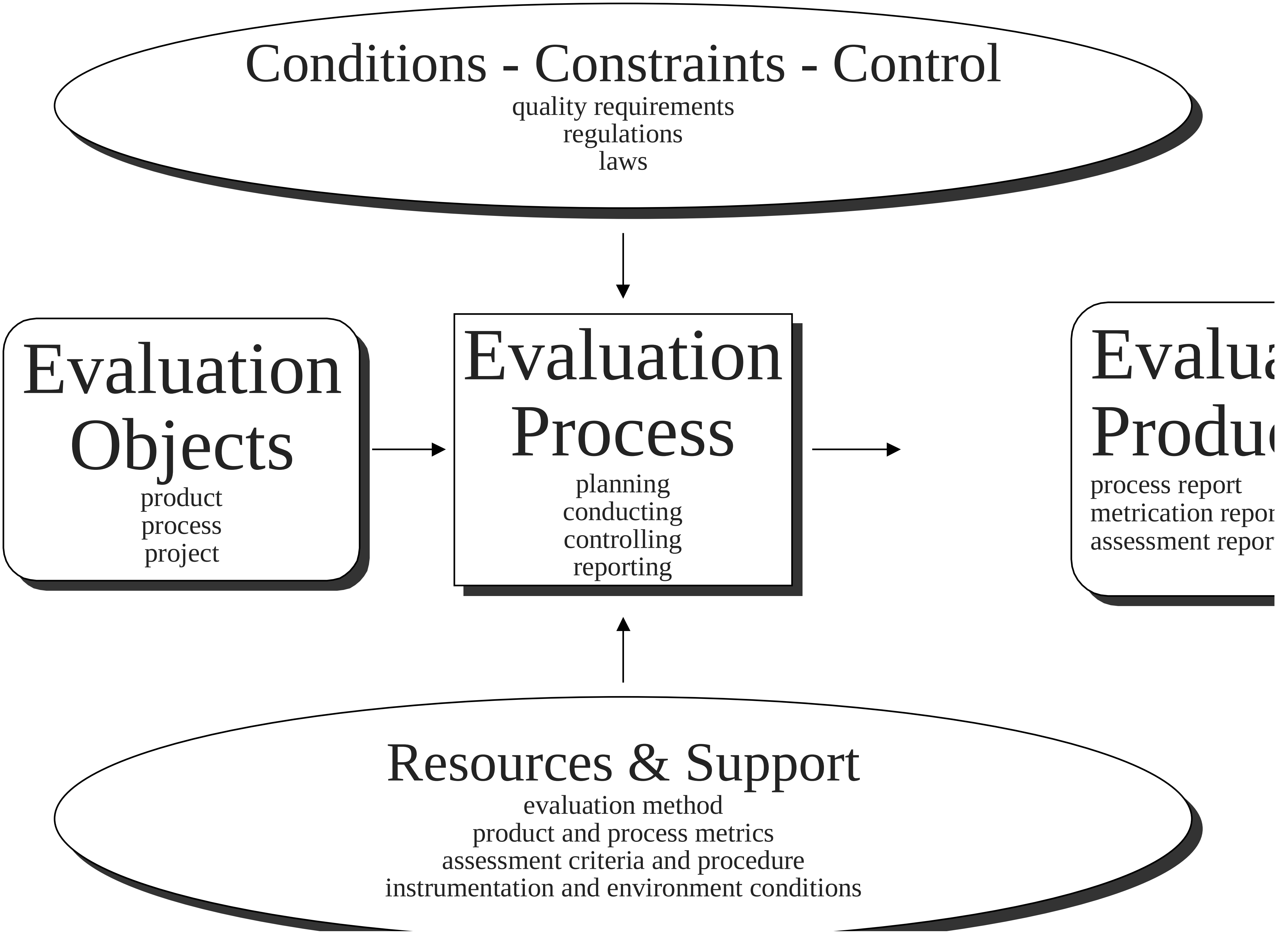
ıre Configuration Management Plan

Software Verification and Validatio

nisation

Software Quality Report

Project SW Qu



Conditions - Constraints - Control

quality requirements
regulations
laws

Evaluation Objects

product
process
project

Evaluation Process

planning
conducting
controlling
reporting

Evalua
Produ

process report
metrication repor
assessment repor

Resources & Support

evaluation method
product and process metrics
assessment criteria and procedure
instrumentation and environment conditions

aterfall Model - Basic requirements, then design, then code, and then tes

ond Model - Code and ideas stagnate and grow other life forms.

ater Fountain Model - Same as pond model, though looks prettier.

rehose Model - Well focused effort on putting out fires.

oilet Bowl Model - Combination of Spiral and Waterfall models. Usuall
ve problem with things that don't flush.

hunderstorm Model - Loud, noisy and dangerous. Usually results in
ooding with developers moving to higher ground.

ornado Model - Faster implementation of Spiral Model, usually wipes ou
velopment staff.

urricane Model - Close attention paid to tracking its course, though no c
n predict when it will arrive.

uality aspects (of the process and the produ

ed and when are they measured (early on in

ment process or merely in the testing phase

netrics are used to measure quality aspects

f Code, number of Function Points, complex

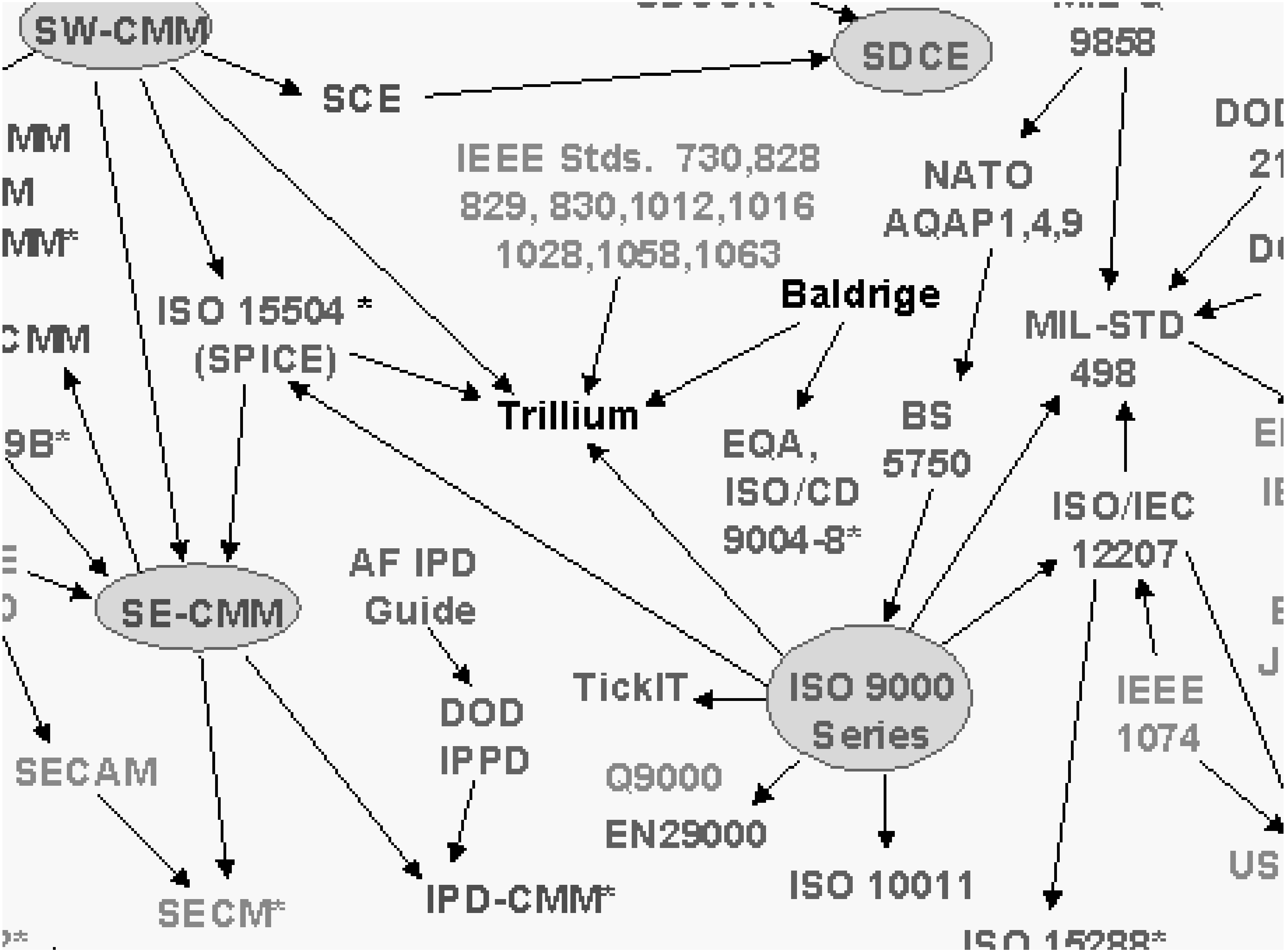
e measurement results used for prediction of

of the final product;

e measurement results used for process

ment;

nethods for process assessment (and


SW-CMM
SCE
SDCE
9858
NATO
AQAP1,4,9
DOD
21
MIL-STD
498
IEEE Stds. 730,828
829, 830,1012,1016
1028,1058,1063
Baldrige
ISO 15504 *
(SPICE)
Trillium
EQA,
ISO/CD
9004-8*
BS
5750
ISO/IEC
12207
MM
M
MM*
CMM
9B*
SE-CMM
AF IPD
Guide
DOD
IPPD
TickIT
ISO 9000
Series
IEEE
1074
Q9000
EN29000
SECAM
SECM*
IPD-CMM*
ISO 10011
ISO 15288*
US

is a set of standards which provides detailed *generic models of quality assuran*
nies can go through a certification process which compares the system again
rd. When challenged by ISO requirements, all employees may be involved to
e and document the processes they use to deliver quality. It gives us a frame
lity management. The series of standards was first published in 1987.

lly, ISO 9000 requires us to document what we do -- and do it.

ationally accepted standard

ompany involvement, especially by management

ation and documentation of common sense

gboard" for managing more effectively

tration technique establishes compliance and involves an assessment by an outside organiz

uous improvement and compliance checked every 6 months. Re-registered every 3 years.

result in smoother development, reduction of time & cost to market, & better communica
projects and departments.

00 and SEI Capability Model are complementary.

ument and Data Control

hasing

rol of Customer Supplied Product

uct Identification and Traceability

ess Control

ection and Testing

coming materials shall be inspected or verified before use.

-process inspection and testing shall be performed.

nal inspection and testing shall be performed prior to release of finished product.

ecords of inspection and test shall be kept.

rol of Inspection, Measuring and Test Equipment

ection and Test Status

rol of Nonconforming Products

ective and Preventative Action (now includes Continuous Improvement)

dling, Storage, Packaging, Preservation and Delivery

rol of Quality Records

nal Quality Audits

ning

icing

3.

is of about 30 internal and external audits of the customer services component of a large
the following breakdown of non-compliances or observations against clauses:

<table>
<tr><th><u>ISO9000</u></th><th><u>frequency</u></th></tr>
<tr><td>Control of Quality Records</td><td>|||||||||||||||||||||||</td></tr>
<tr><td>Corrective and Preventative Action</td><td>|||||||||||||||||||||||</td></tr>
<tr><td>Document and Data Control</td><td>|||||||||||||||||||||||</td></tr>
<tr><td>Handling, Storage, Packaging, Preservation and Delivery</td><td>|||||||||||||||</td></tr>
<tr><td>Control of Inspection, Measuring and Test Equipment</td><td>||||||||||||||</td></tr>
<tr><td>Design Control</td><td>||||||||||||</td></tr>
<tr><td>Process Control</td><td>|||||||||||</td></tr>
<tr><td>Management Responsibility</td><td>|||||||||</td></tr>
<tr><td>Training</td><td>|||||</td></tr>
<tr><td>Control of Nonconforming Products</td><td>||||</td></tr>
<tr><td>Contract Review</td><td>||||</td></tr>
<tr><td>Product Identification and Traceability</td><td>|||</td></tr>
<tr><td>Statistical Techniques</td><td>||</td></tr>
<tr><td>Servicing</td><td>||</td></tr>
<tr><td>Quality System</td><td>||</td></tr>
</table>

r Management Style

A formal organization

Provisions for planning

Procedures for key activities

Quality records

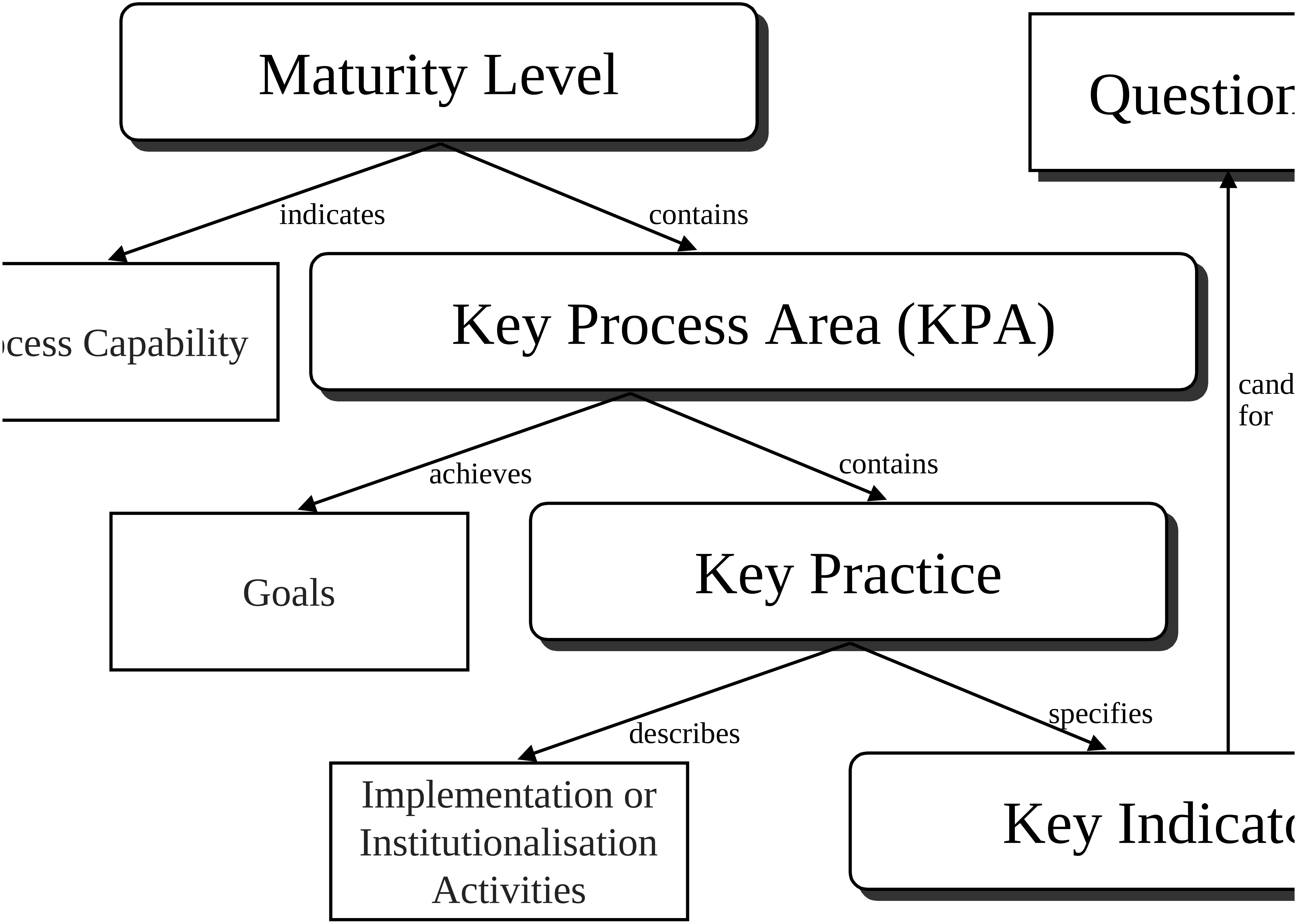
System review and corrections

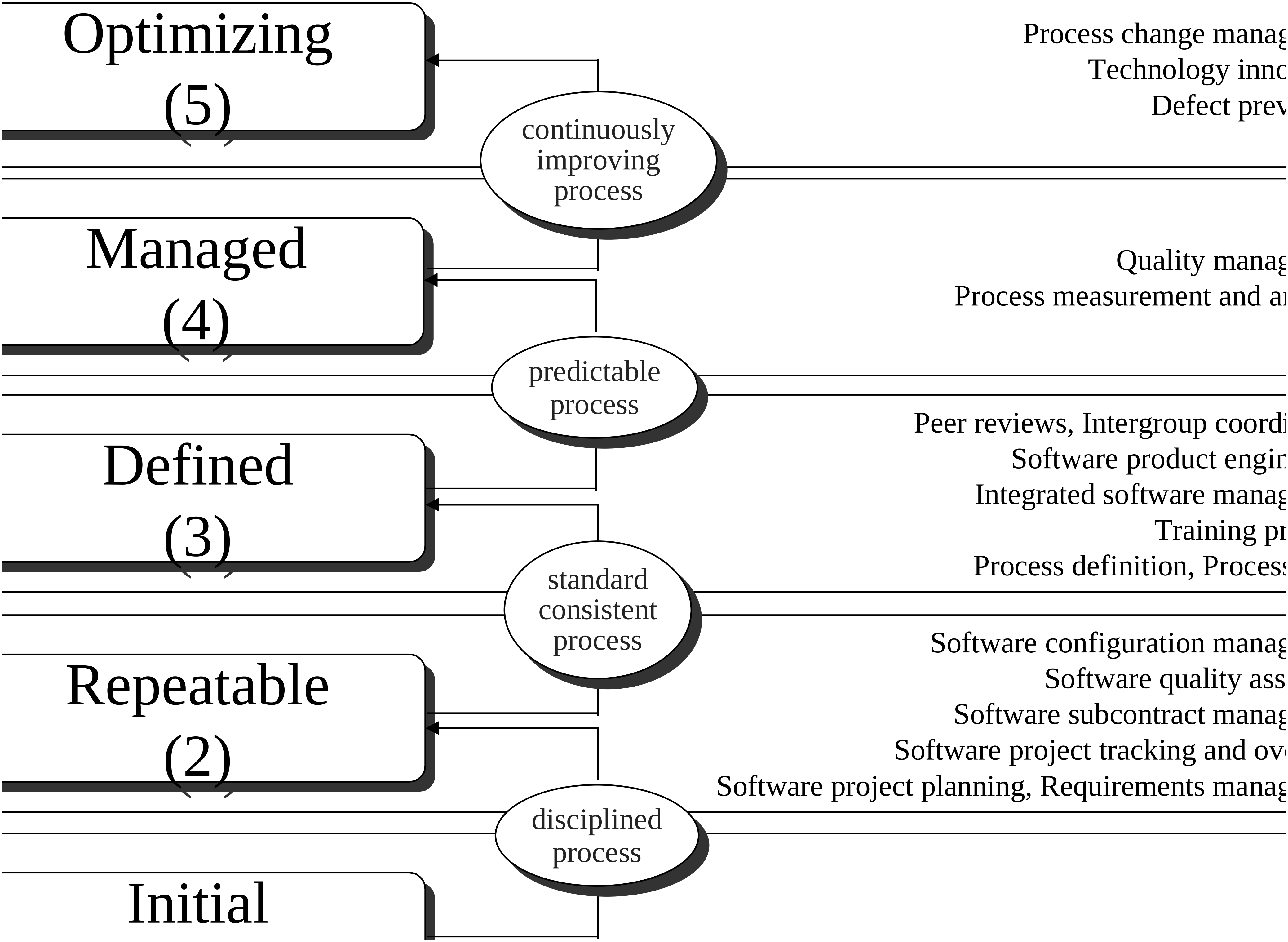
Software Engineering Institute (SEI), Carnegie Mellon University, USA (for US DoD)

November 1986 SEI with assistance of MITRE
began work on a method of assessing and improving
SOFTWARE DEVELOPMENT PROCESSES

September 1987 SEI released:
Process Maturity Framework and Maturity Framework

August 1991 SEI released:
Improved Capability Maturity Model for Software


Maturity Level
Question
indicates
contains
ocess Capability
Key Process Area (KPA)
cand
for
achieves
contains
Goals
Key Practice
describes
specifies
Implementation or Institutionalisation Activities
Key Indicat



Optimizing
(5)

Process change manag
Technology inno
Defect prev

continuously
improving
process

Managed
(4)

Quality manag
Process measurement and an

predictable
process

Peer reviews, Intergroup coordi
Software product engin
Integrated software manag
Training pr
Process definition, Process

Defined
(3)

standard
consistent
process

Software configuration manag
Software quality ass
Software subcontract manag
Software project tracking and ov
Software project planning, Requirements manag

Repeatable
(2)

disciplined
process

Initial

anning

imates are documented for use in planning and tracking the software project.

oject activities and commitments are planned and documented.

ups and individuals agree to their commitments related to the software project.

racking and Oversight

ts and performances are tracked against the software plans.

ctions are taken and managed to closure when actual results and performance deviate signi
ns.

oftware commitments are agreed to by the affected groups and individuals.

ct Management

ontractor selects qualified software subcontractors.

ontractor and the software subcontractor agree to their commitments to each other.

ontractor and the software subcontractor maintain ongoing communications.

ontractor tracks the software subcontractor's actual results and performance against its co

ssurance

ality assurance activities are planned.

f software products and activities to the applicable standards, procedures, and requiremen

ups and individuals are informed of software qualityassurance activities and results.

nce issues that cannot be resolved within the software project are addressed by senior mana ı-level process development and improvement act v t es are planned.

ss Definition

;oftware process for the organization is developed and maintained.

related to the use of the organization's standard software process by the software projects i
ade available.

ivities are planned.

developing the skills and knowledge needed to perform software management and technica

n the software engineering group and software-related groups receive the training necessar
les.

: Management

s defined software process is a tailored version of the organization's standard software proc

is planned and managed according to the project's defined software process.

ngineering

: engineering tasks are defined, integrated, and consistently performed to produce the softw

rk products are kept consistent with each other.

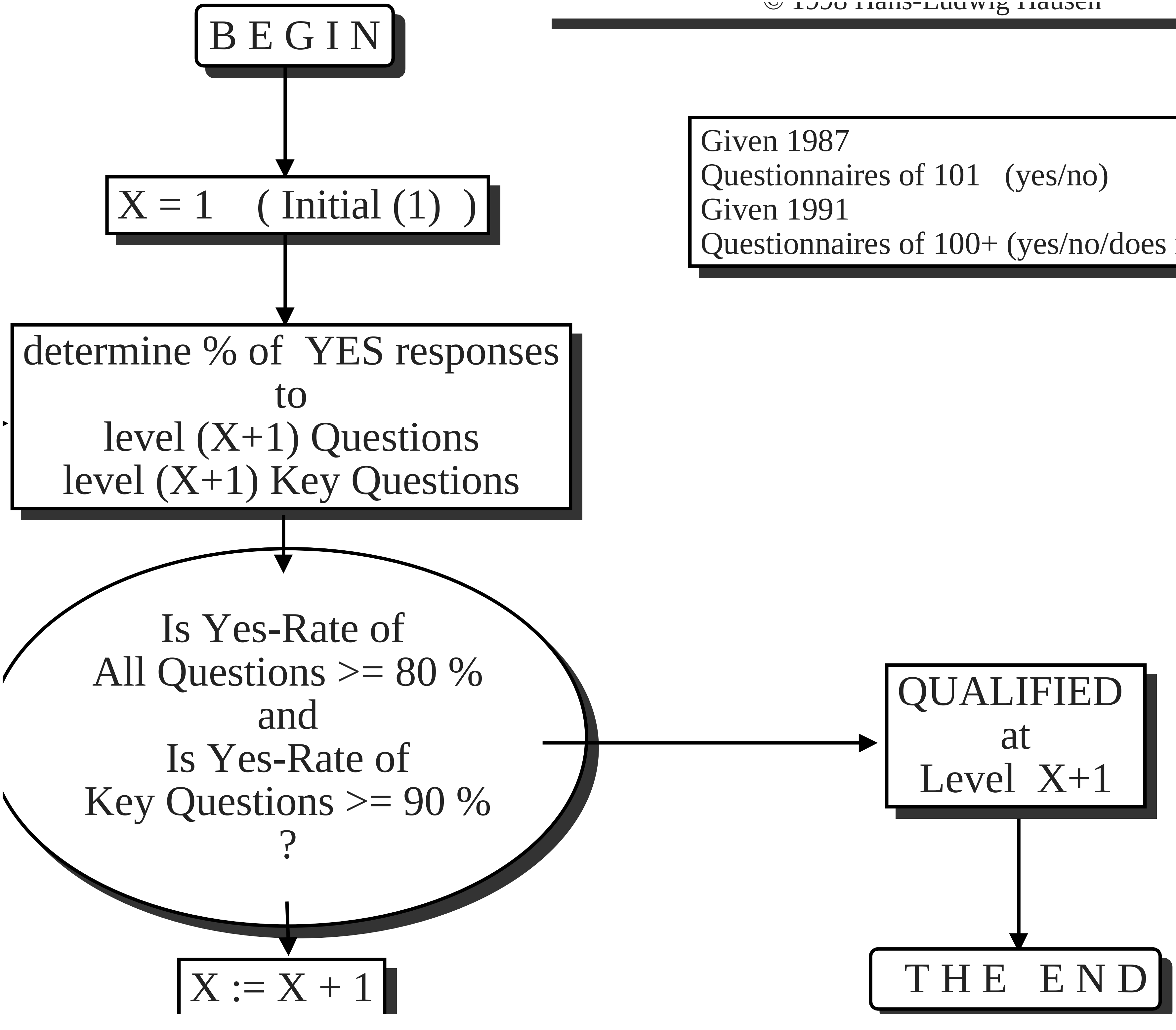
ıation

:r's requirements are agreed to by all affected groups.

ments between the engineering groups are agreed to by the affected groups.

ring groups identify, track, and resolve intergroup issues.

Given 1987
Questionnaires of 101 (yes/no)
Given 1991
Questionnaires of 100+ (yes/no/does not apply/not ap



BEGIN

X = 1 (Initial (1))

determine % of YES responses
to
level (X+1) Questions
level (X+1) Key Questions

Is Yes-Rate of
All Questions >= 80 %
and
Is Yes-Rate of
Key Questions >= 90 %
?

QUALIFIED
at
Level X+1

X := X + 1

THE END

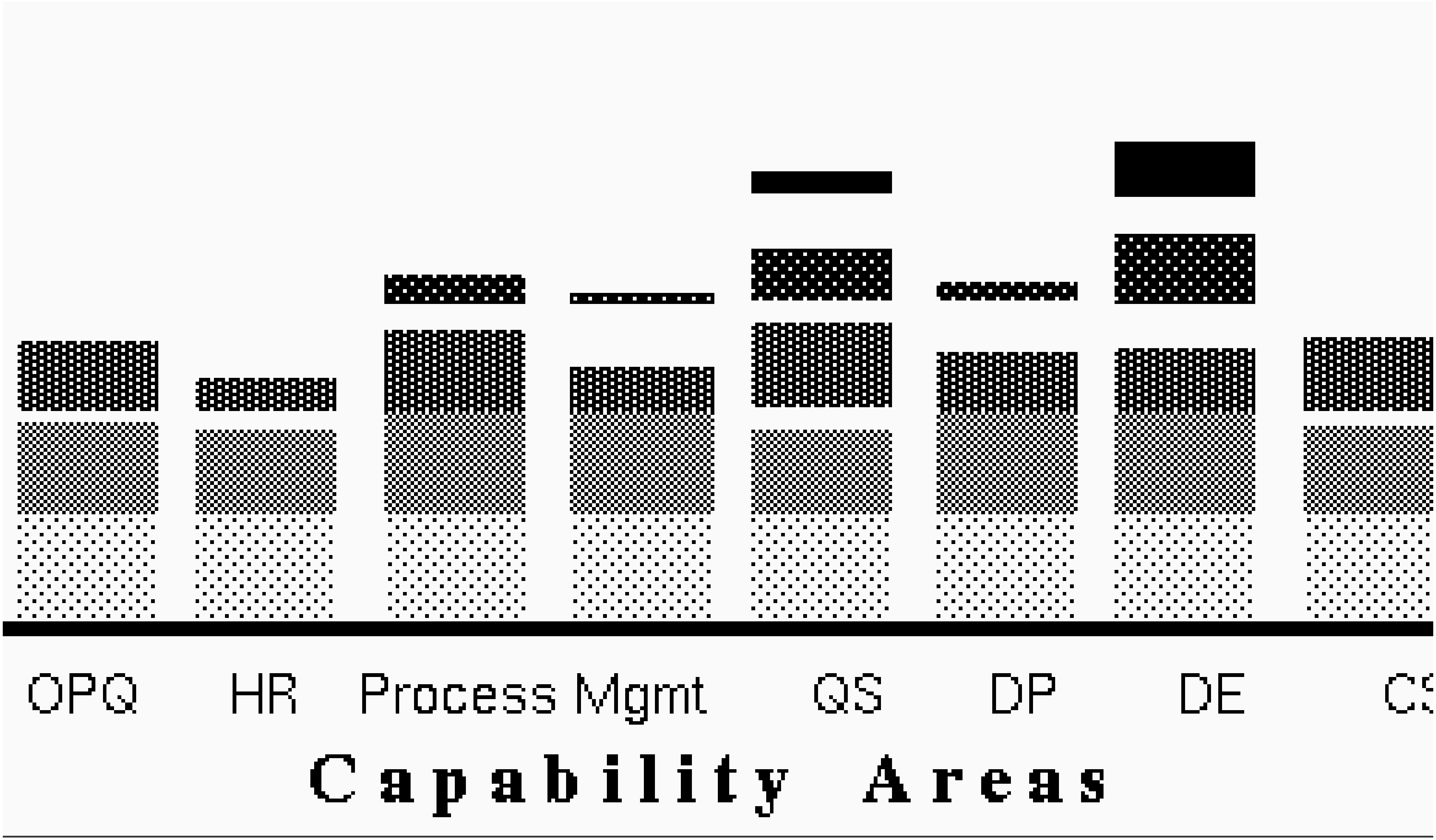
red: The development process is adhoc. Projects frequently cannot meet quality c
ss, while possible, is based on individuals rather than on organizational infrastruc

e and Project Oriented: Individual project success is achieved through strong pr
planning and control, with emphasis on requirements management, estimation tec
management. (Risk - Medium)

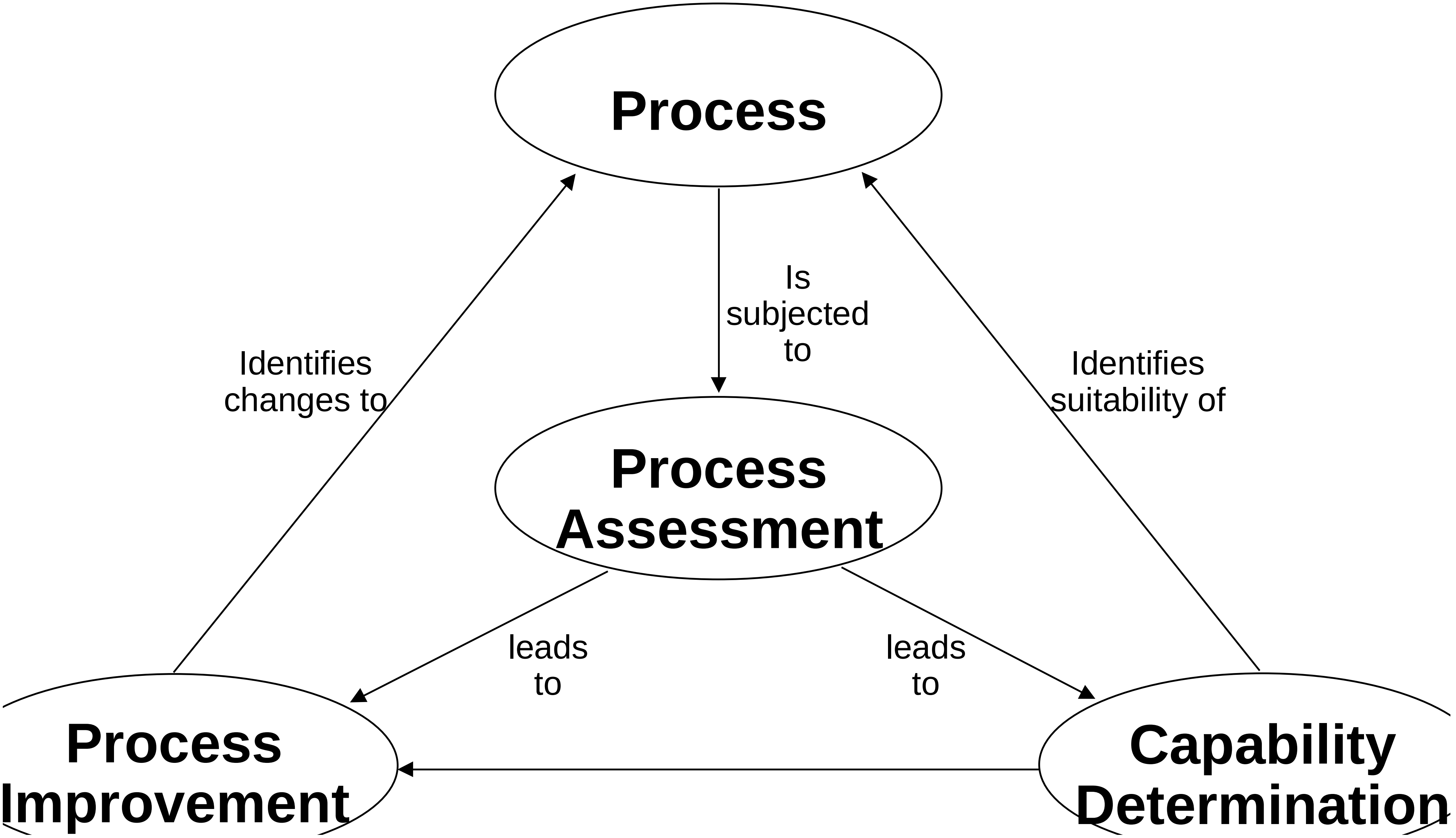
d Process Oriented: Processes are defined and utilized at the organizational leve
nization is still permitted. Processes are controlled and improved. ISO 9001 requi
d internal process auditing are incorporated. (Risk - Low)

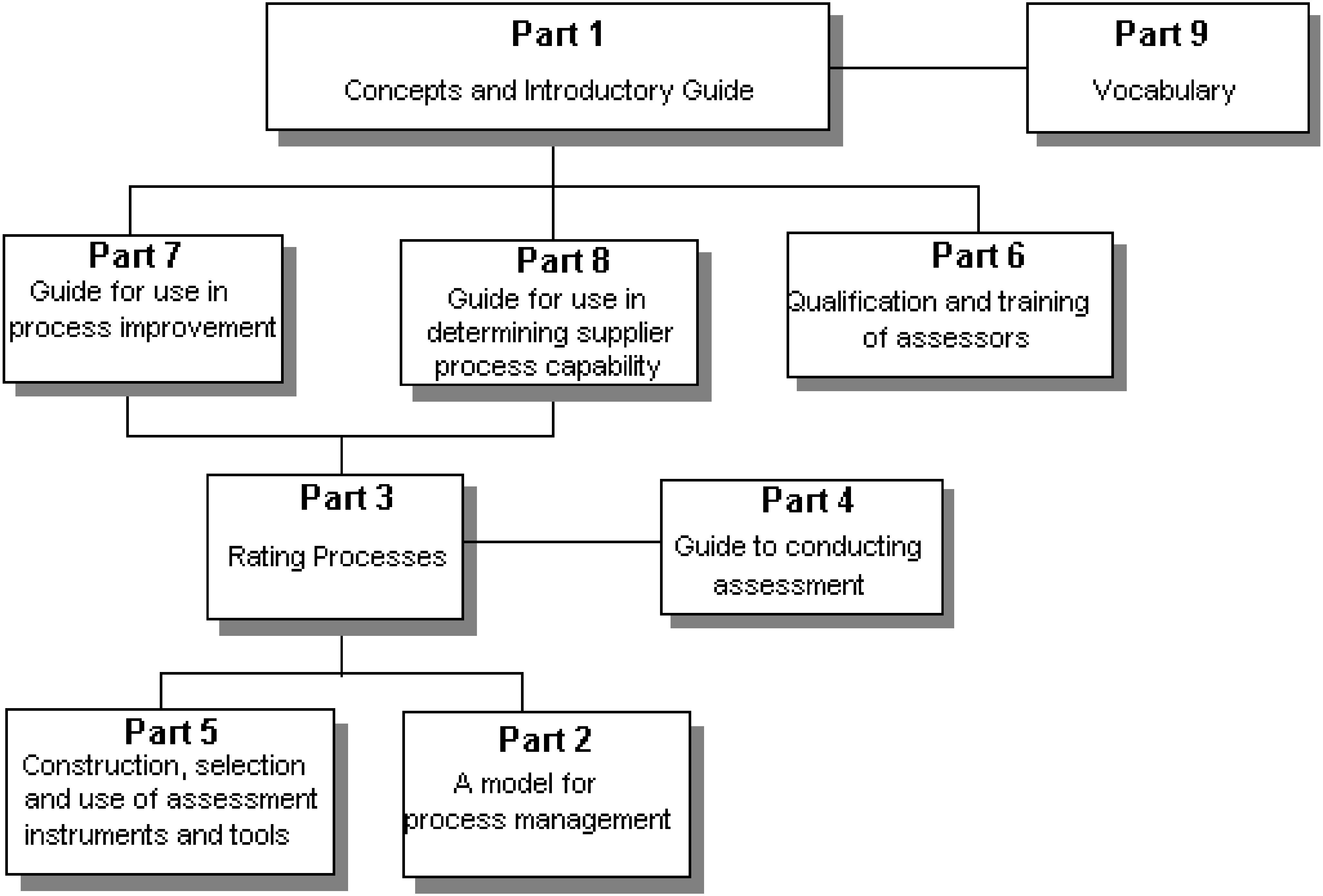
and Integrated: Process instrumentation and analysis is used as a key mechanism
Process change management and defect prevention programs are integrated into
re integrated into processes. (Risk - Lower)

grated: Formal methodologies are extensively used. Organizational repositories f
history and process are utilized and effective. (Risk - Lowest)



ganizational Process Quality, Human Resource Development and Management, Process, Management, Qua
evelopment Practices, Development Environment Customer Support


Process
Is subjected to
Identifies changes to
Identifies suitability of
Process Assessment
leads to
leads to
Process Improvement
Capability Determination


Part 1
Concepts and Introductory Guide
Part 9
Vocabulary
Part 7
Guide for use in process improvement
Part 8
Guide for use in determining supplier process capability
Part 6
Qualification and training of assessors
Part 3
Rating Processes
Part 4
Guide to conducting assessment
Part 5
Construction, selection and use of assessment instruments and tools
Part 2
A model for process management

an entry point into this International Standard. It describes f the suite fit together, and provides guidance for their selec explains the requirements contained within the Standard ar bility to the conduct of an assessment, to the construction a on of supporting tools, and to the construction of extended p ed processes are processes which include base practices a lefined in the part 2 of the Standard, or which are entirely n ses, for example to meet industry specific requirements.

this International Standard defines, at a high level, the nental activities that are essential to software engineering, s ing to increasing levels of process capability. These baselir e extended, through the generation of application or sector s e guides, to take account of specific industry, sector or othe ments.

this International Standard defines a framework for conduc
ment, and sets out the basis for rating, scoring and profiling
lities.

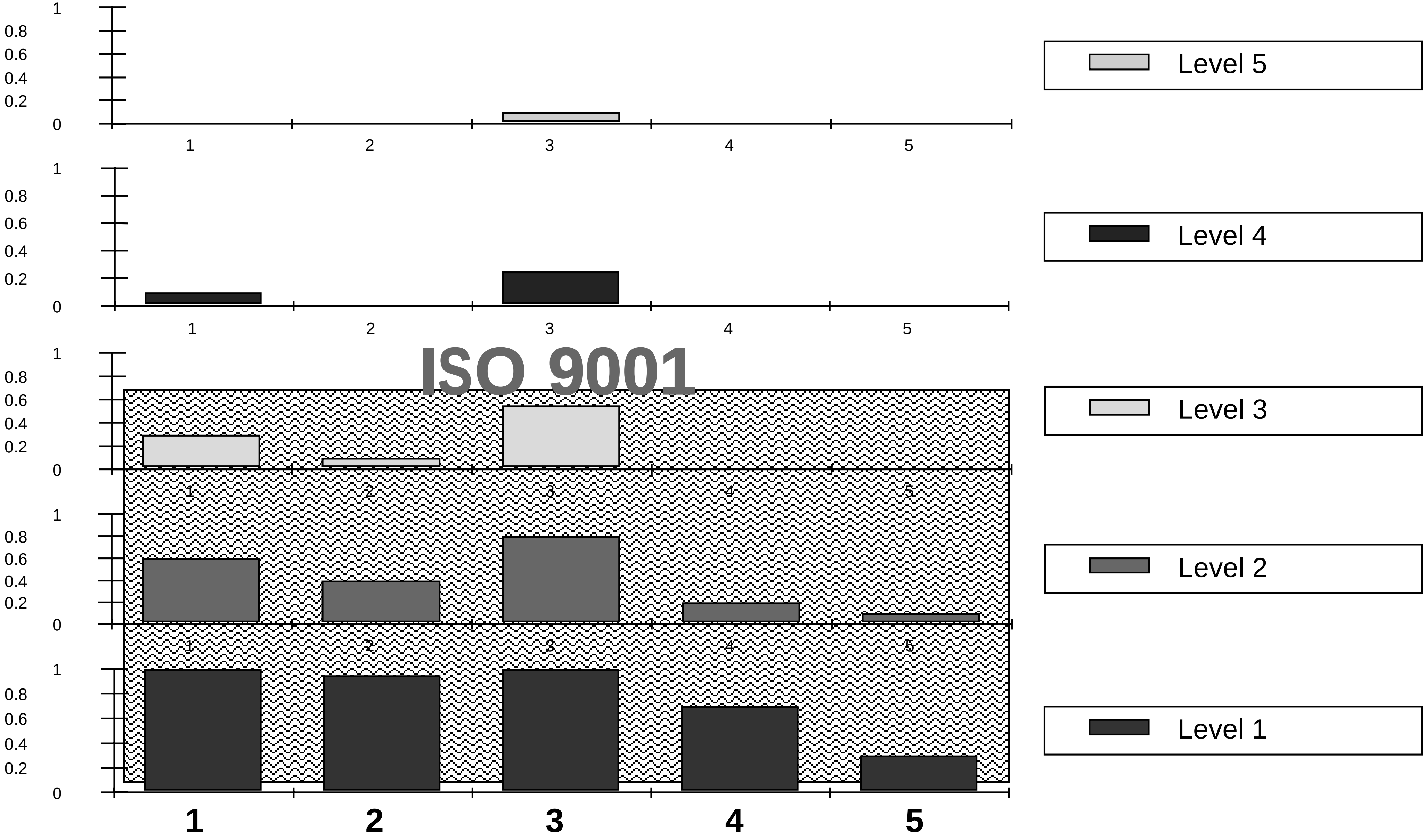
this International Standard provides guidance on the condu
re process assessments. This guidance is generic enough t
ble across all organizations, and also for performing asses
a variety of different methods and techniques, and supporte
range of tools.

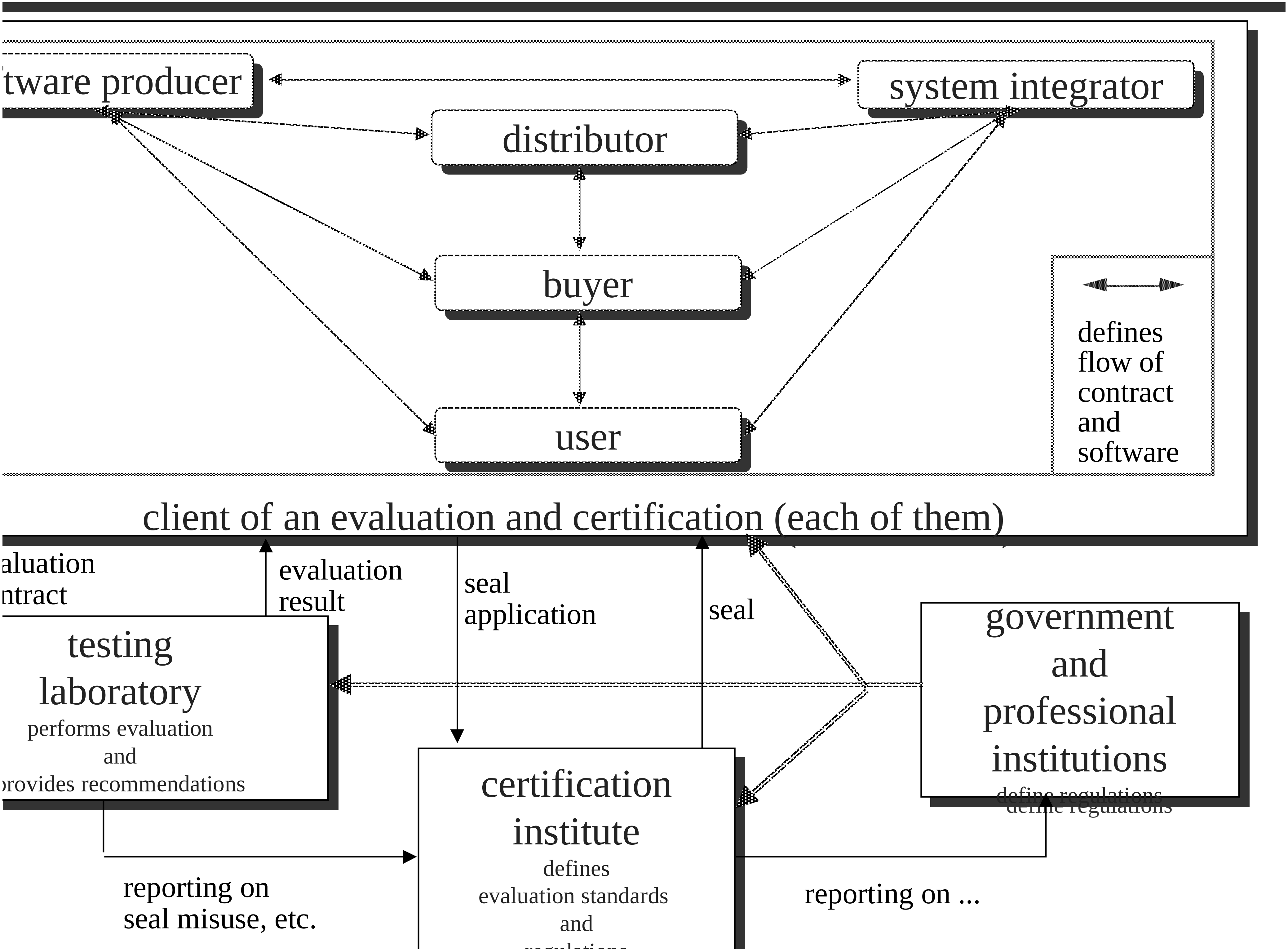
this International Standard defines the framework elements
d to construct an instrument to assist an assessor in the pe
ssessment. In addition, it provides guidance to acquirers or
selection and usability aspects of various types of assessm
nents.

g and experience of assessors that are relevant to conductin
ments. It describes mechanisms that may be used to demo
tence and to validate education, training and experience.

this International Standard describes how to define the inp
e the results of an assessment for the purposes of process
ement. The guide includes examples of the application of p
ement in a variety of situations.

this International Standard describes how to define the inp
e the results of an assessment for the purpose of process c
ination. It addresses process capability determination in bo
tforward situations and in more complex situations involving
ucted or future capability. The guidance on conducting proc
lity determination is applicable either for use within an orga
rmine its own capability, or by a acquirer to determine the c
otential) supplier.
a consolidated vocabulary of all terms specifically defined f


ISO 9001
1
0.8
0.6
0.4
0.2
0
1
2
3
4
5
Level 5
Level 4
Level 3
Level 2
Level 1



tware producer

system integrator

distributor

buyer

user

defines flow of contract and software

client of an evaluation and certification (each of them)

aluation
ntract

evaluation
result

seal
application

seal

testing
laboratory
performs evaluation
and
rovides recommendations

government
and
professional
institutions
define regulations

define regulations

certification
institute
defines
evaluation standards
and
regulations

reporting on
seal misuse, etc.

reporting on ...

gu de This document is a guide to the ISO/IEC 1xxxx series of documents which product quality. It can be used by purchasers, users, producers and independent rs who wish to evaluate the quality of software products.

's guide Planning for software measurement is applicable to all audiences. When uations are to be done, planning is important. This part gives guidance on how to vare measurement and provides an example of a plan.

r's guide The Developer's guide is intended mainly for use during software devel ance. It focuses on the use of those indicators that can predict end product quality liate products developed during the life-cycle.

guide The Buyer's guide focuses on the evaluation of comparable software produc ers who need to select one for specific use. The buyer's guide introduces a metho f quality characteristics defined by ISO/IEC 9126-1.

r's guide The Evaluator's guide is intended for those who perform independent ev onally. Often they work for third party organisations. The Evaluator's guide desc ing the set of quality characteristics defined by ISO/IEC 9126-1. It also describes ual issues relating to third party evaluation.

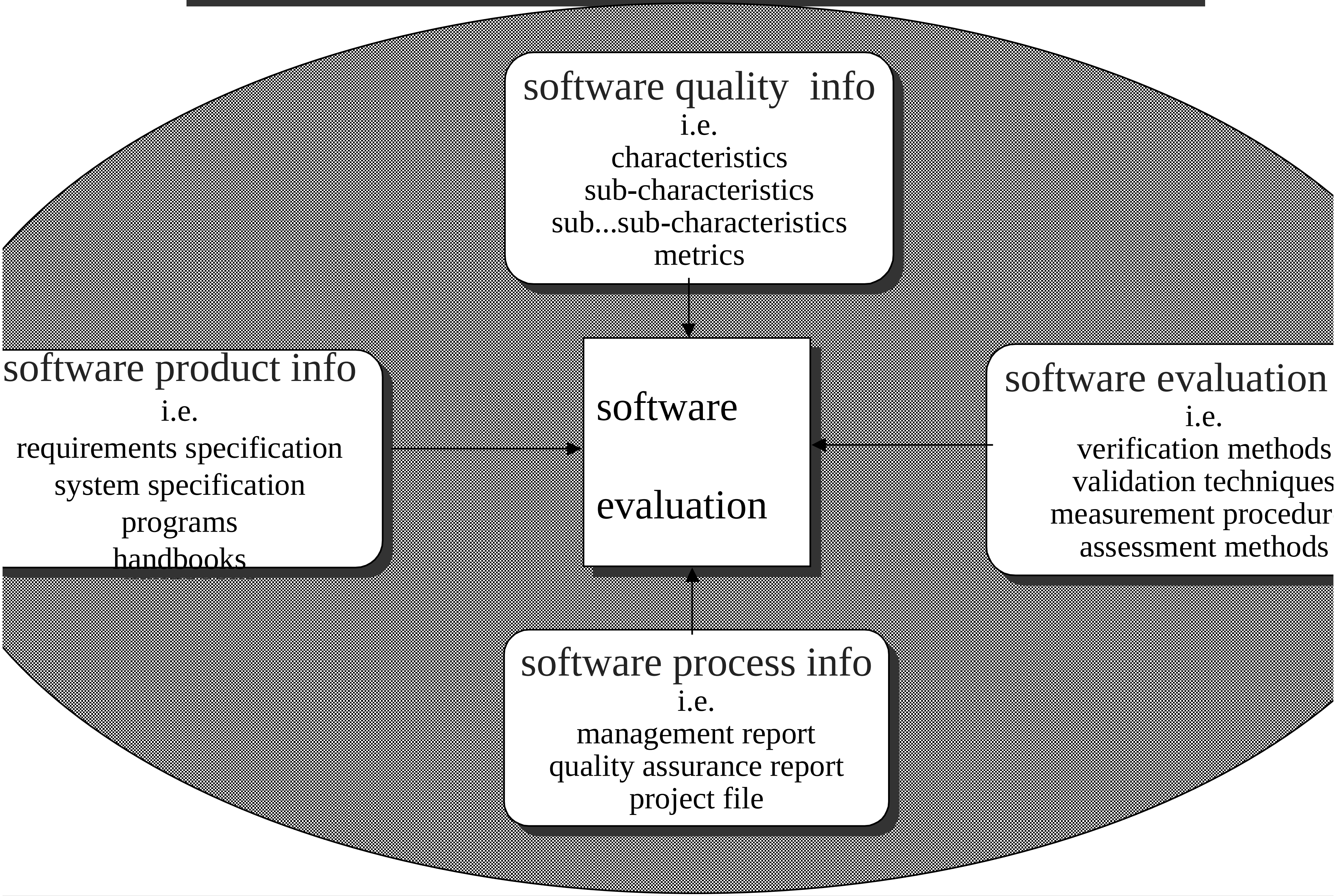
on module guide This part provides guidance for developing, documenting and v on modules. An evaluation module collects together quality characteristics, metri ent techniques.

peatability: Repeated evaluation of the same roduct to the same evaluation specification by he same testing laboratory gives the same result.

producibility: Repeated evaluation of the same roduct to the same evaluation specification by ifferent testing laboratories gives the same result.

partiality: Evaluation is free from unfair bias owards achieving any particular result.

jectivity: The evaluation result is obtained with he minimum of subjective judgement.



software quality info
i.e.
characteristics
sub-characteristics
sub...sub-characteristics
metrics

software product info
i.e.
requirements specification
system specification
programs
handbooks

software evaluation

software evaluation
i.e.
verification methods
validation techniques
measurement procedur
assessment methods

software process info
i.e.
management report
quality assurance report
project file

evaluation = verification + validation + measurement + assessment

Which Evaluation Level ?

© 1998 Hans-Ludwig Hausen

thoroughness
of
evaluation

UATION LEVEL D

ety risks	no impact
onomic risks	small loss
plication domain	small office automaton, entertainment, household
chniques	inspection of important features, some program metrics

UATION LEVEL C

ety risks	few people disabled
onomic risks	company affected by loss
plication domain	fire alarm, process control, financial systems
chniques	inspection, black box testing, selected program and specification metrics

UATION LEVEL B

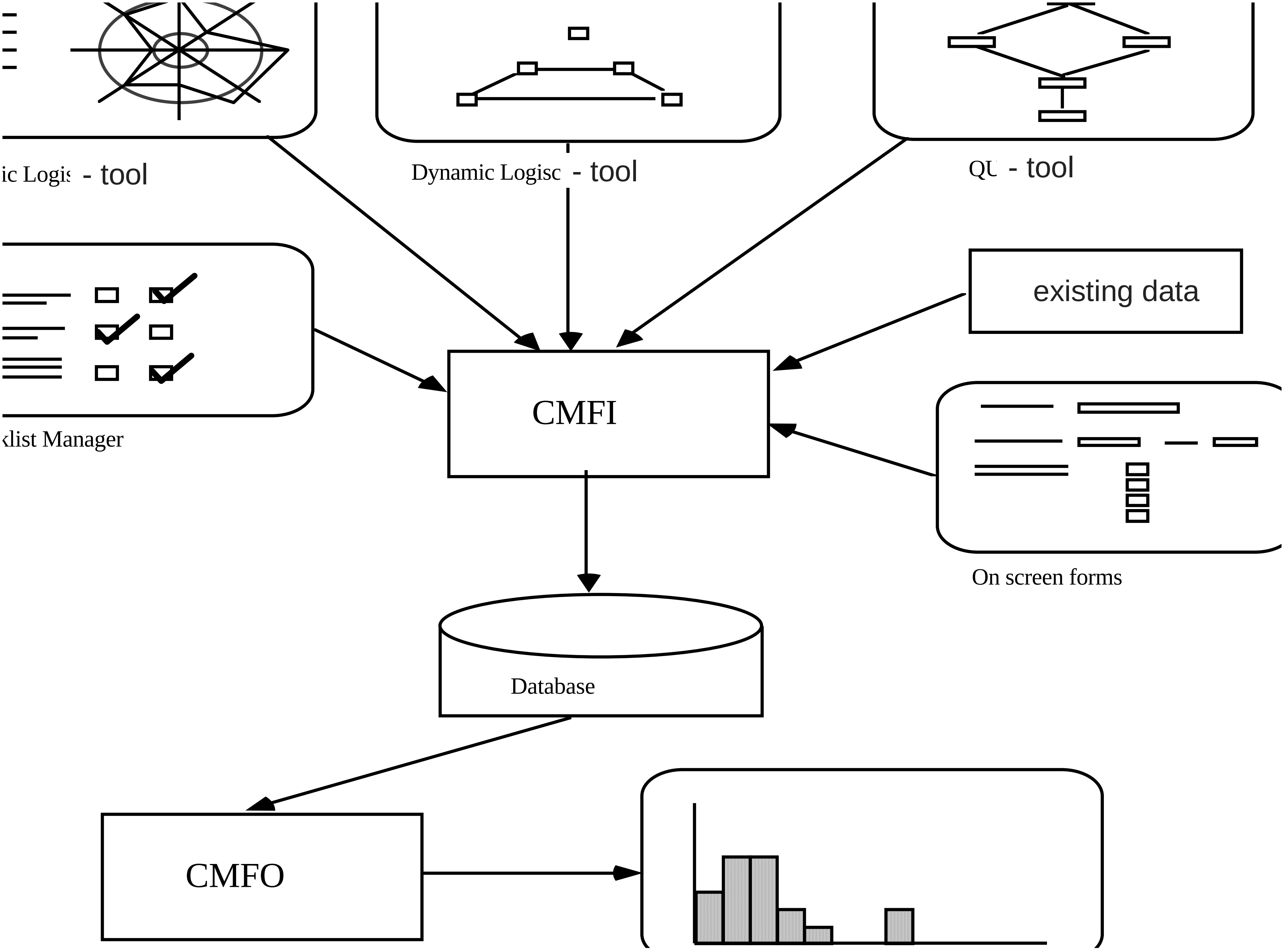
ety risks	some people killed
onomic risks	company endangered by loss
plication domain	fire alarm, process control, financial systems
chniques	inspection, black box testing, glass box testing, program and specification metrics

UATION LEVEL A

ety risks	many people killed

vare Qual ty Evaluat on Module encapsulates:

- the definition of one or more atomic evaluation procedures applied on product or process information in order to measure software characteristics or sub-characteristics,
- the attachment of metrics and evaluation levels to those characteristics,
- the assessment procedure to be applied at the particular evaluation level,
- the format for reporting results and costs.


ic Logis - tool
Dynamic Logisc - tool
QU - tool
existing data
klist Manager
CMFI
On screen forms
Database
CMFO

ntainability

to be collected

ollected are the following:

inability check-lists:

TEM or SUB-SYSTEM Description
gh-Level Specification Description,
gh-Level Design Description, Part C -
scription.

DULE Description Part A - Low-
fication Description, Part B - Low-
gn Description, Part C - Low-Level
escription, Part D - Low-Level
scription, Part E - General Description.

DULE Implementation Multiplicity,

MPOUND MODULE Cohesiveness,

TABASE FILE Description

e following metrics are needed:

Prime, Nesting, Product VINAP

following metrics are needed:

Terminology used: (just one example) **Module:** A M
single logical item which is used with other logical item
software subsystem. The definition of a module is deper
language. Here are some examples of modules for diffe
languages: FORTRAN - function, subroutine, procedur
program, Pascal - function, main program and pro
BASIC - subroutine and main program, C
dbase - procedure, functions CORAL66 - pro
main part of segment, PROLOG - procedure, COBO
procedures, programs

The product level decision for determining the maintainab
whole product is dependent on the number of source c
which pass the pass/fail criteria set out above. The rec
product level assessment for maintainability are:

Assessment Level	Required Pass Percentage
A	90%
B	70%
C	50%

m possible score" and round up the answer to 2
ces. Actual Score: Maximum Possible Score: 8
Score:

available then the pass/fail criteria are:

= 60 .and.

<= 5 .and.

PRIME <= 5 .and.

NABILITY CHECK-LIST <= 40%   .or.

= 69 .and.

<= 5 .and.

PRIME >= 5 .and.

NABILITY CHECK-LIST <= 40% .and.

pection of module source code reveals a CASE
uct.

available then the pass/fail criteria are:

NABILITY CHECK-LIST <= 40% .and.
NTS <= 46 .and.

GE LEVEL <= 6 .and.

TION CONTENT <= 83 .and.

If Q-tool and L-tool are available then the pass/fail crite

PASS

If LENGTH <= 60 .and.

NESTING <= 5 .and.

BIGGEST PRIME <= 5 .and.

MAINTAINABILITY CHECK-LIST <= 40% .and

STATEMENTS <= 46 .and.

LANGUAGE LEVEL <= 6 .and.

INFORMATION CONTENT <= 83 .and.

PENDING NODES <= 2

.or.

If PVINAP <= 69 .and.

NESTING <= 5 .and.

BIGGEST PRIME >= 5 .and.

MAINTAINABILITY CHECK-LIST <= 40% .and

manual inspection of module source code reveals a
type construct. .and.

STATEMENTS <= 46 .and.

LANGUAGE LEVEL <= 6 .and.

INFORMATION CONTENT <= 83 .and.

ng Procedures of Testing Laboratory

Guide 25 a number for requirements for working procedures of testing
tories)

ality System of the Testing Laboratory should include:

eral quality procedures

ity assurance procedures specific for each evaluation

back and corrective actions whenever evaluation discrepancies are detected

edures for dealing with complaints

ıdling of Test Items must include rules for:

fidentiality and Security

A. Specification of the Evaluation

1. Identification of the Parties
2. Identification of the Product
3. Purpose of the Agreement
4. Identification of Evaluation Procedure

B. Conduct of the Evaluation

1. The client's Obligations

   Provisions regarding delivery of software and associated information

2. The testing laboratory's Obligations

   a. Duration of Evaluation

   b. Qualifications of Evaluation Staff

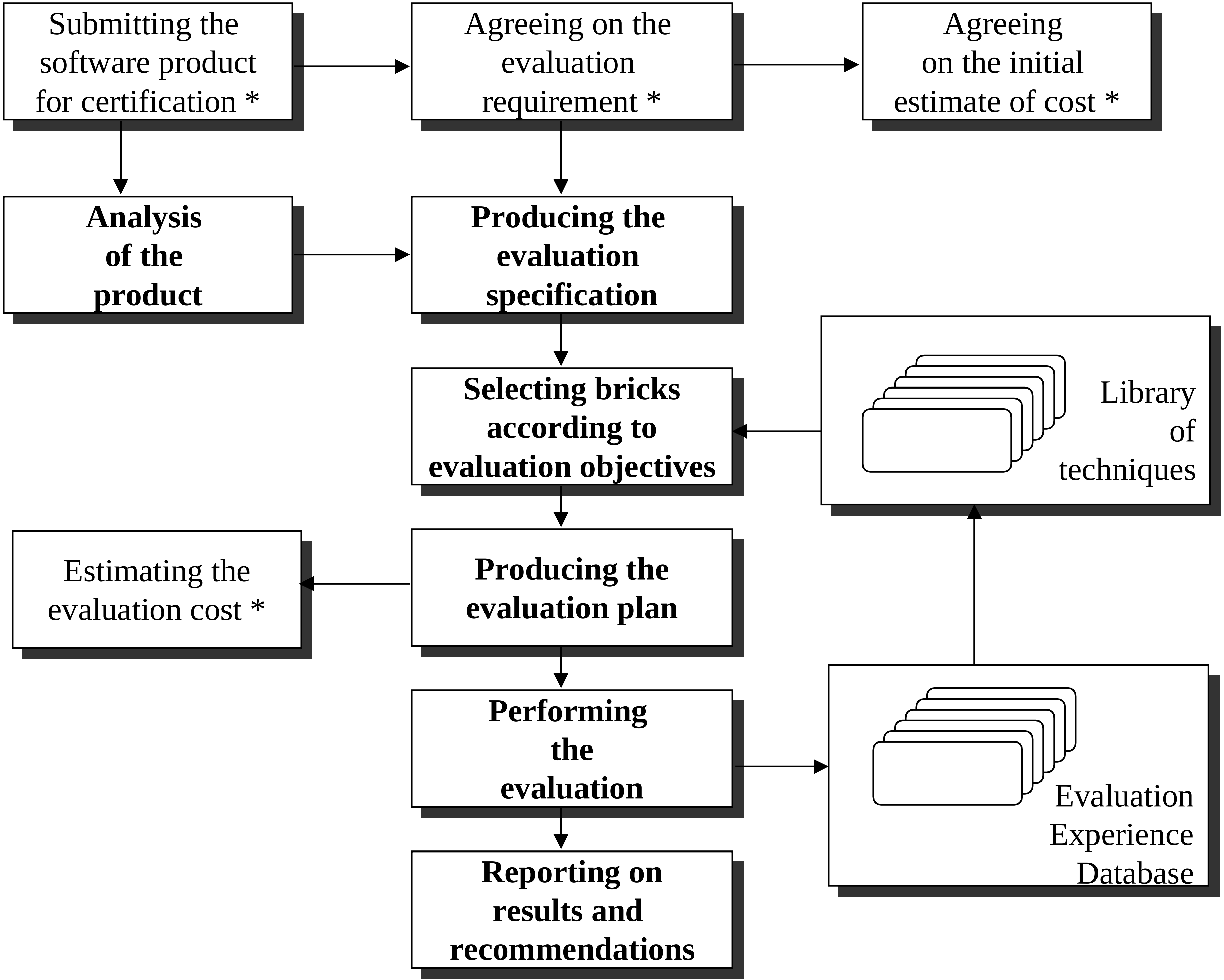
   c. Conduct of the Evaluation

C. Evaluation Report

1. Presentation of the Results/Format of the Evaluation Report
2. Dispute Resolution Procedures
3. Use to Which the Report May be Put
4. Resubmitting of products/Testing of New Versions

D. General Legal Terms and Conditions

1. Confidentiality
2. Intellectual Property Issues
3. Exclusion/Limitation Clauses



Submitting the software product for certification *

Agreeing on the evaluation requirement *

Agreeing on the initial estimate of cost *

Analysis of the product

Producing the evaluation specification

Selecting bricks according to evaluation objectives

Library of techniques

Estimating the evaluation cost *

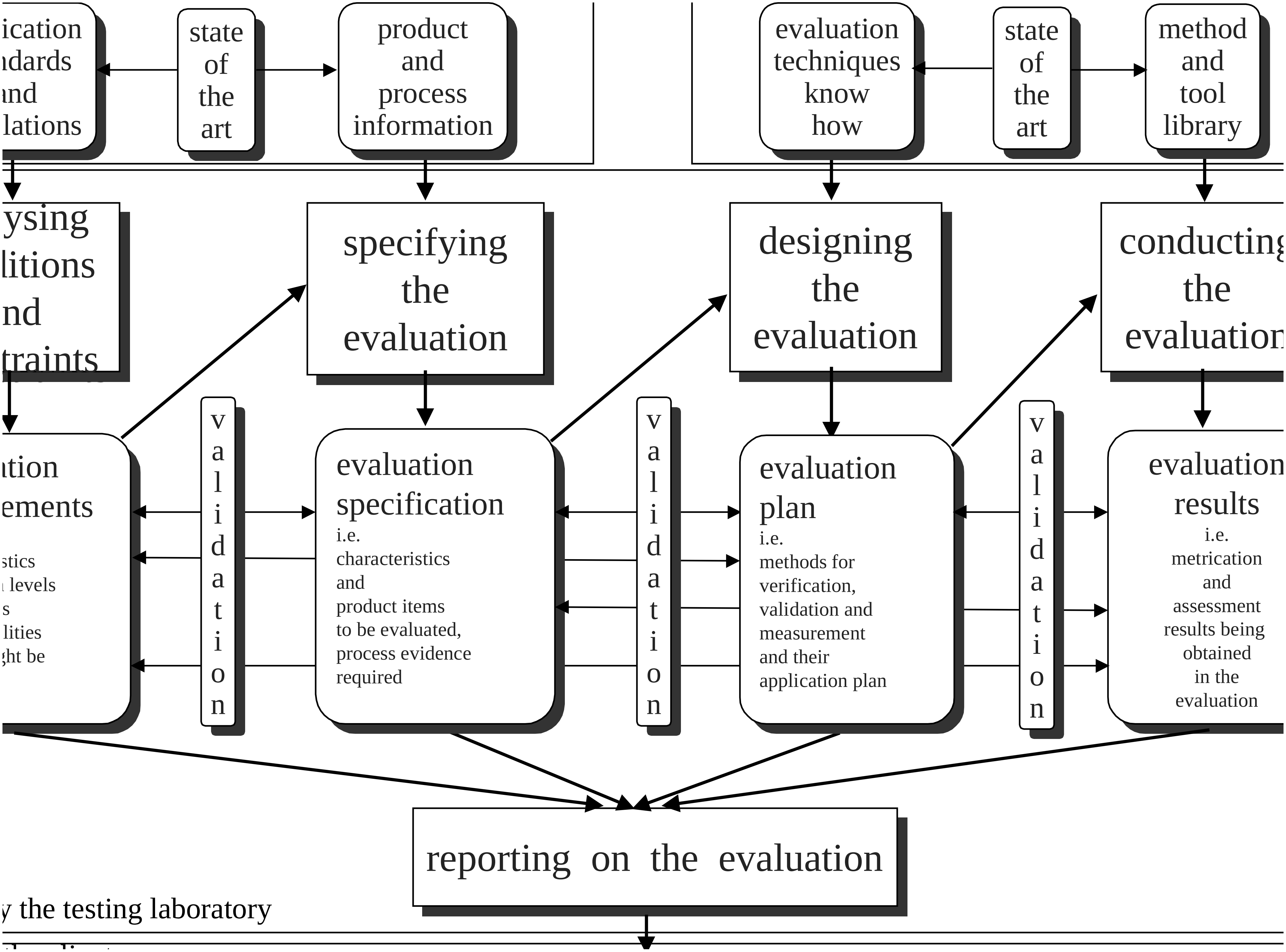
Producing the evaluation plan

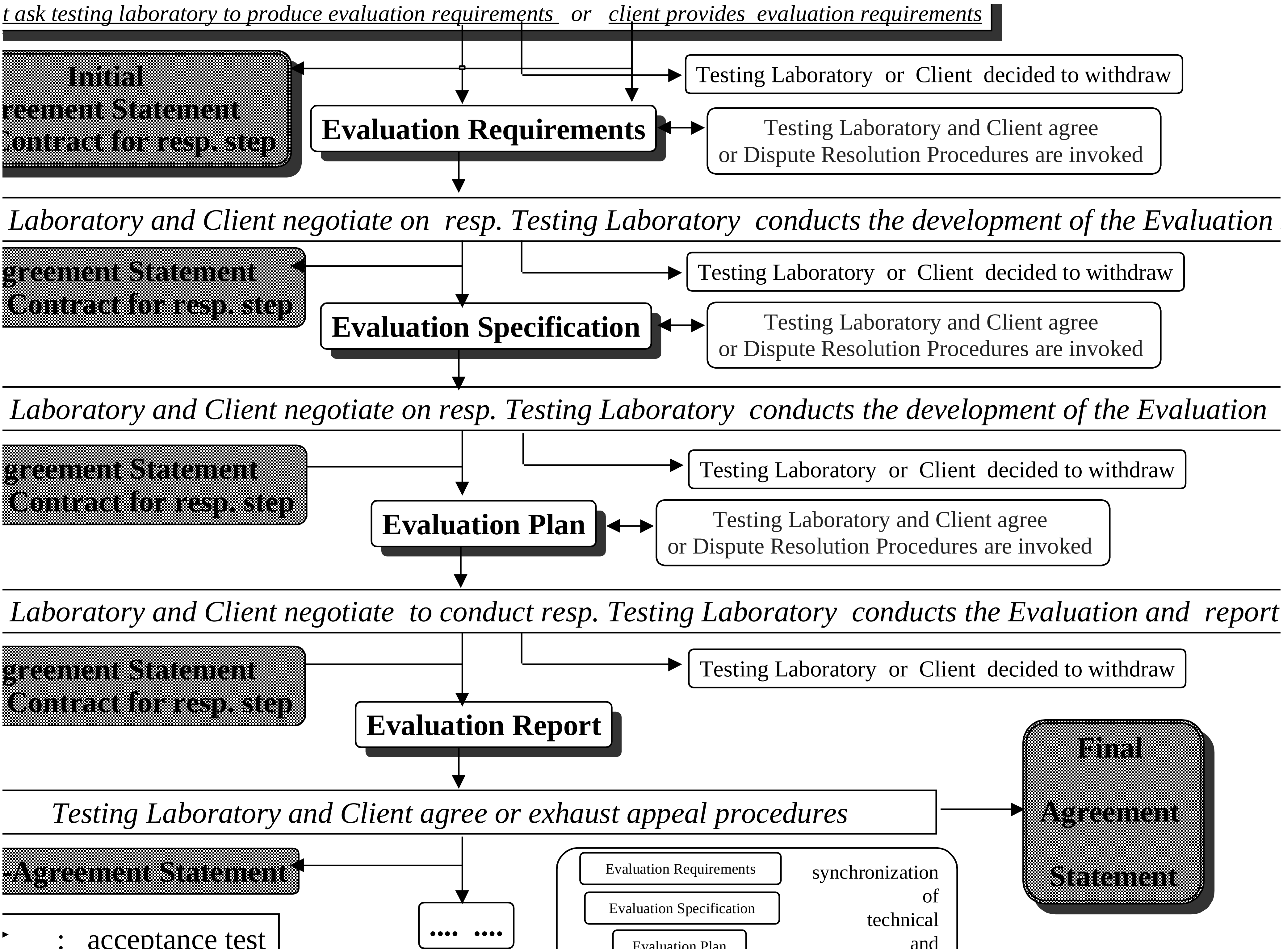
Performing the evaluation

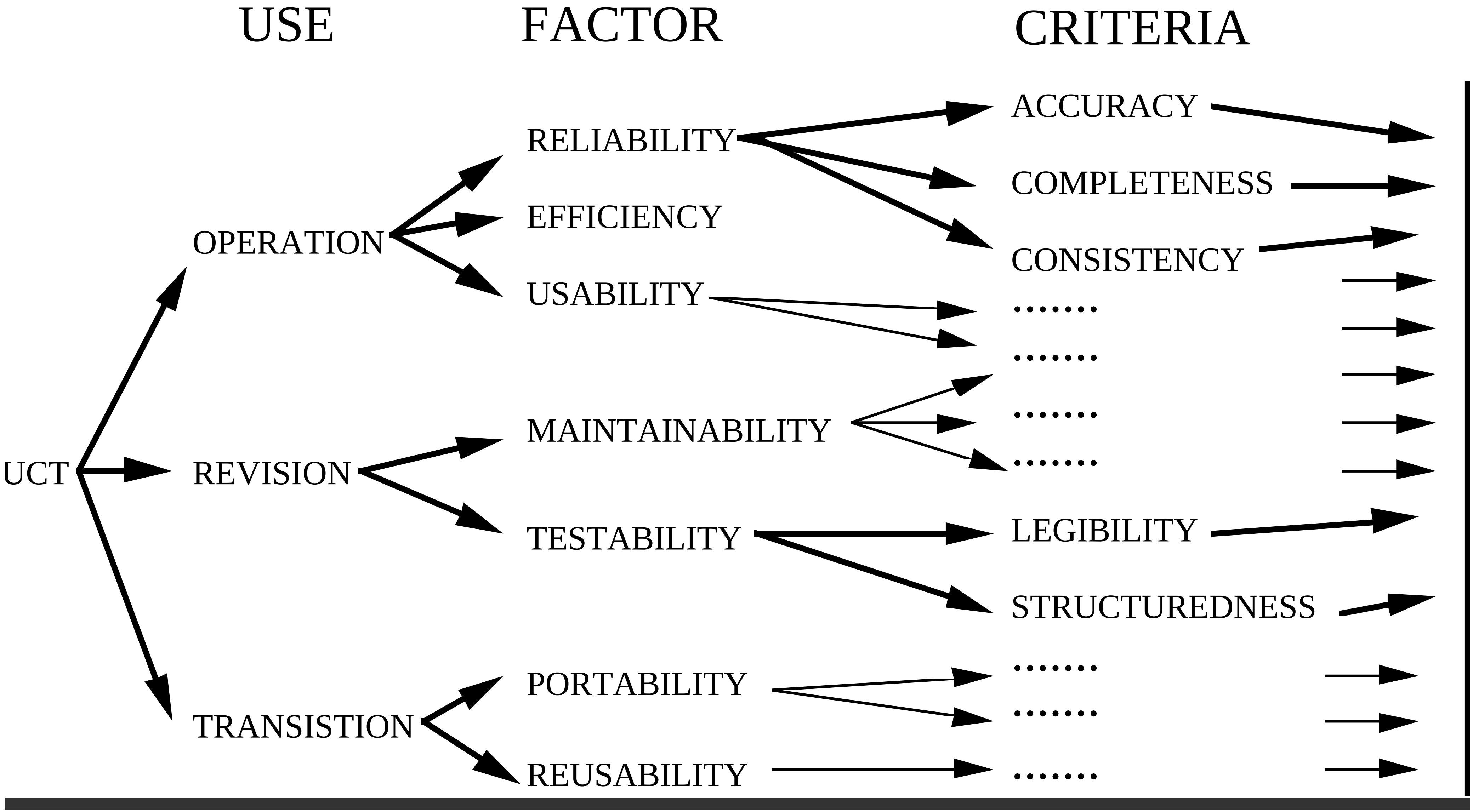
Evaluation Experience Database

Reporting on results and recommendations

* these actions are optional


ication
ndards
and
lations
state of the art
product and process information
evaluation techniques know how
state of the art
method and tool library
ysing
litions
nd
traints
specifying the evaluation
designing the evaluation
conducting the evaluation
ation
ements
stics
levels
s
lities
ght be
validation
evaluation specification
i.e.
characteristics
and
product items
to be evaluated,
process evidence
required
validation
evaluation plan
i.e.
methods for
verification,
validation and
measurement
and their
application plan
validation
evaluation results
i.e.
metrication
and
assessment
results being
obtained
in the
evaluation
reporting on the evaluation
y the testing laboratory


t ask testing laboratory to produce evaluation requirements or client provides evaluation requirements
Initial
reement Statement
Contract for resp. step
Testing Laboratory or Client decided to withdraw
Evaluation Requirements
Testing Laboratory and Client agree
or Dispute Resolution Procedures are invoked
Laboratory and Client negotiate on resp. Testing Laboratory conducts the development of the Evaluation
greement Statement
Contract for resp. step
Testing Laboratory or Client decided to withdraw
Evaluation Specification
Testing Laboratory and Client agree
or Dispute Resolution Procedures are invoked
Laboratory and Client negotiate on resp. Testing Laboratory conducts the development of the Evaluation
greement Statement
Contract for resp. step
Testing Laboratory or Client decided to withdraw
Evaluation Plan
Testing Laboratory and Client agree
or Dispute Resolution Procedures are invoked
Laboratory and Client negotiate to conduct resp. Testing Laboratory conducts the Evaluation and report
greement Statement
Contract for resp. step
Testing Laboratory or Client decided to withdraw
Evaluation Report
Final
Agreement
Statement
Testing Laboratory and Client agree or exhaust appeal procedures
-Agreement Statement
....
Evaluation Requirements
Evaluation Specification
Evaluation Plan
synchronization
of
technical
and
: acceptance test



USE FACTOR CRITERIA

UCT

OPERATION

RELIABILITY

ACCURACY

COMPLETENESS

CONSISTENCY

EFFICIENCY

USABILITY

....

....

REVISION

MAINTAINABILITY

....

....

TESTABILITY

LEGIBILITY

STRUCTUREDNESS

TRANSISTION

PORTABILITY

....

....

REUSABILITY

....

A set of attributes that bear on the ability of software to be transferred from one environment to

istics: **adaptability, conformance, installability, replaceability = acir**

A set of attributes that bear on the relationship between the level of performance of the software
sources used

istics: **resource behaviour, time behaviour = rt**

A set of attributes that bear on the capability of software to maintain its level of performance und
r a stated period of time

istics: **fault tolerance, maturity, recoverability = fm**

lity A set of attributes that bear on the existence of a set of functions and their specified properti
those that satisfy stated or implied needs

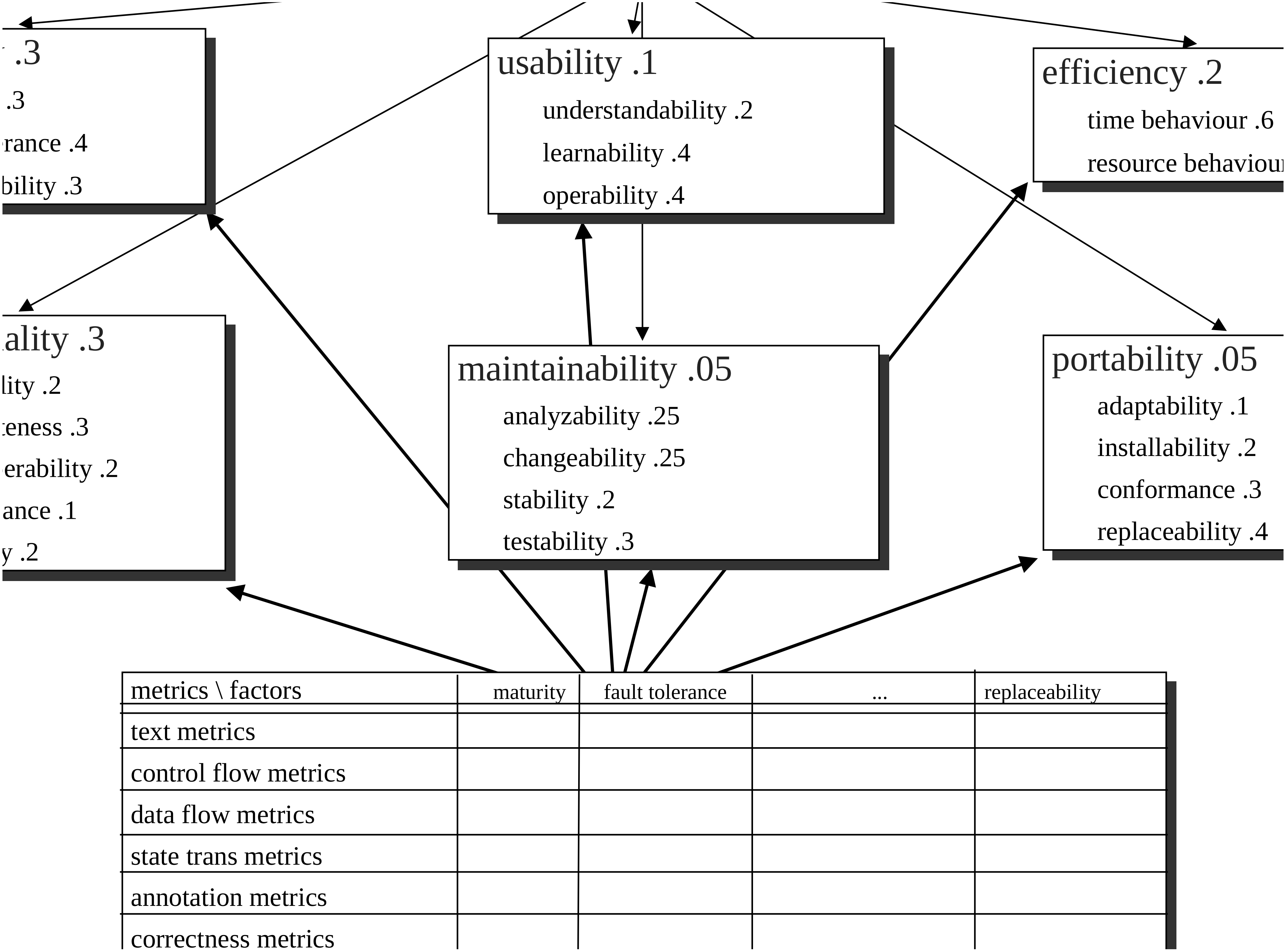
istics: **accurateness, compliance, interoperability, security, suitability = aciss**

set of attributes that bear on the effort needed for use and on the individual assessment of such u
lied set of users

istics: **learnability, operability, understandability = lou**

bility A set of attributes that bear on the effort needed to make specified modifications

istics: **analyzability, changeability, stability, testability = acst**



. .3

.3

rance .4

bility .3

usability .1

understandability .2

learnability .4

operability .4

efficiency .2

time behaviour .6

resource behaviou

ality .3

lity .2

eness .3

erability .2

ance .1

y .2

maintainability .05

analyzability .25

changeability .25

stability .2

testability .3

portability .05

adaptability .1

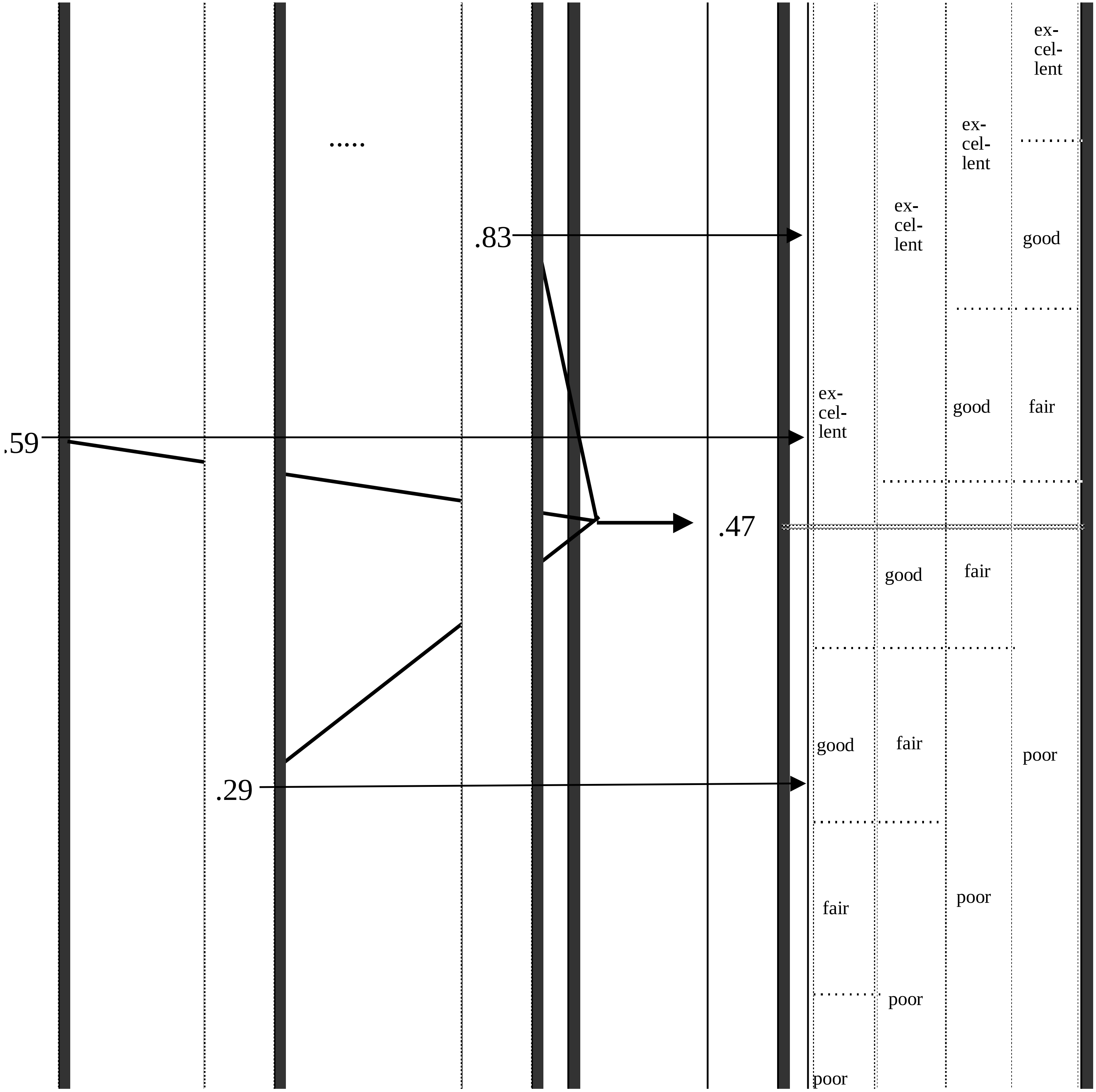
installability .2

conformance .3

replaceability .4

metrics \ factors	maturity	fault tolerance	...	replaceability
text metrics				
control flow metrics				
data flow metrics				
state trans metrics				
annotation metrics				
correctness metrics				

Evalua



i.e.

for each met

we get
three thresh
each level

and

two extreme

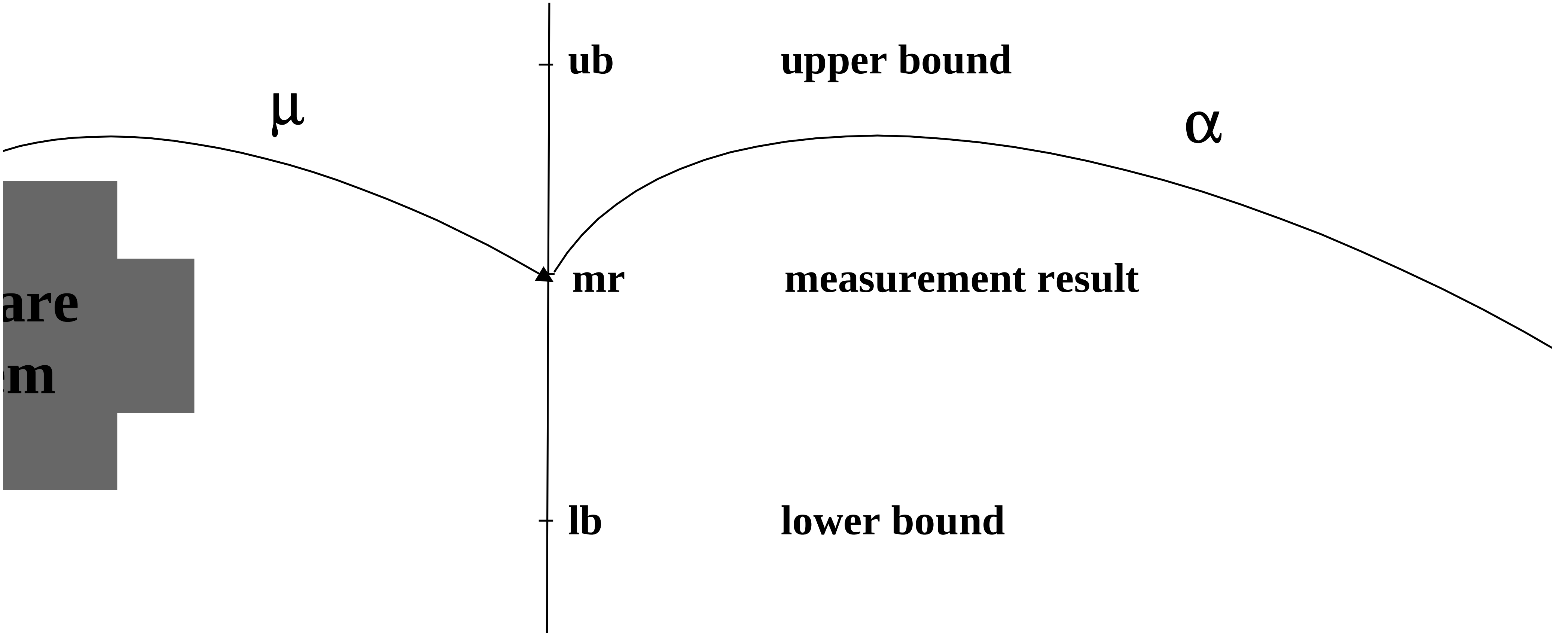
or

we have to h
all metrics v
evaluation n
onto the 16
ranking cate

needs to distinguish:

a) assessment of “goodness” of one measurement re

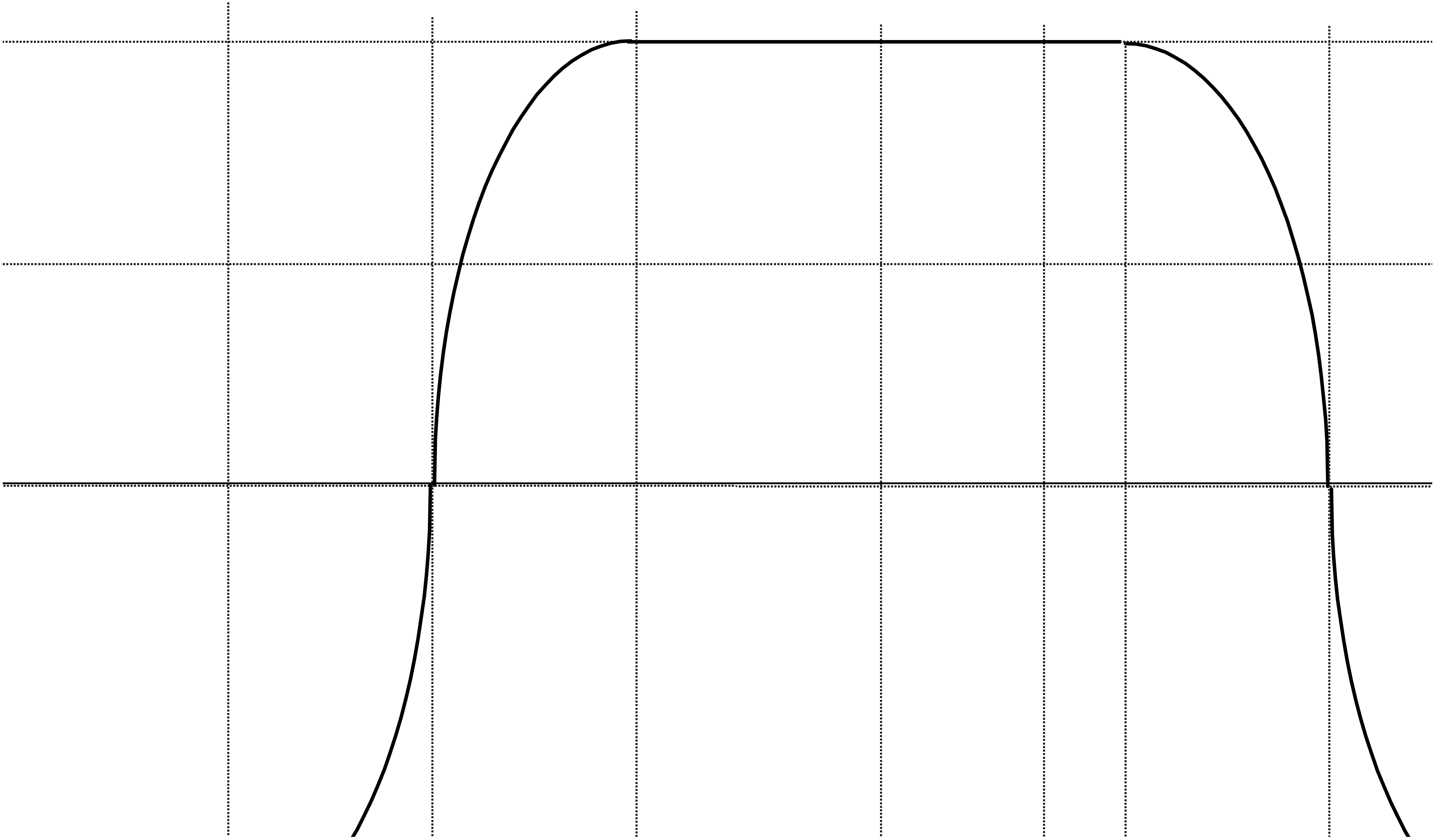
b) assessment of “goodness” of aggregated measure

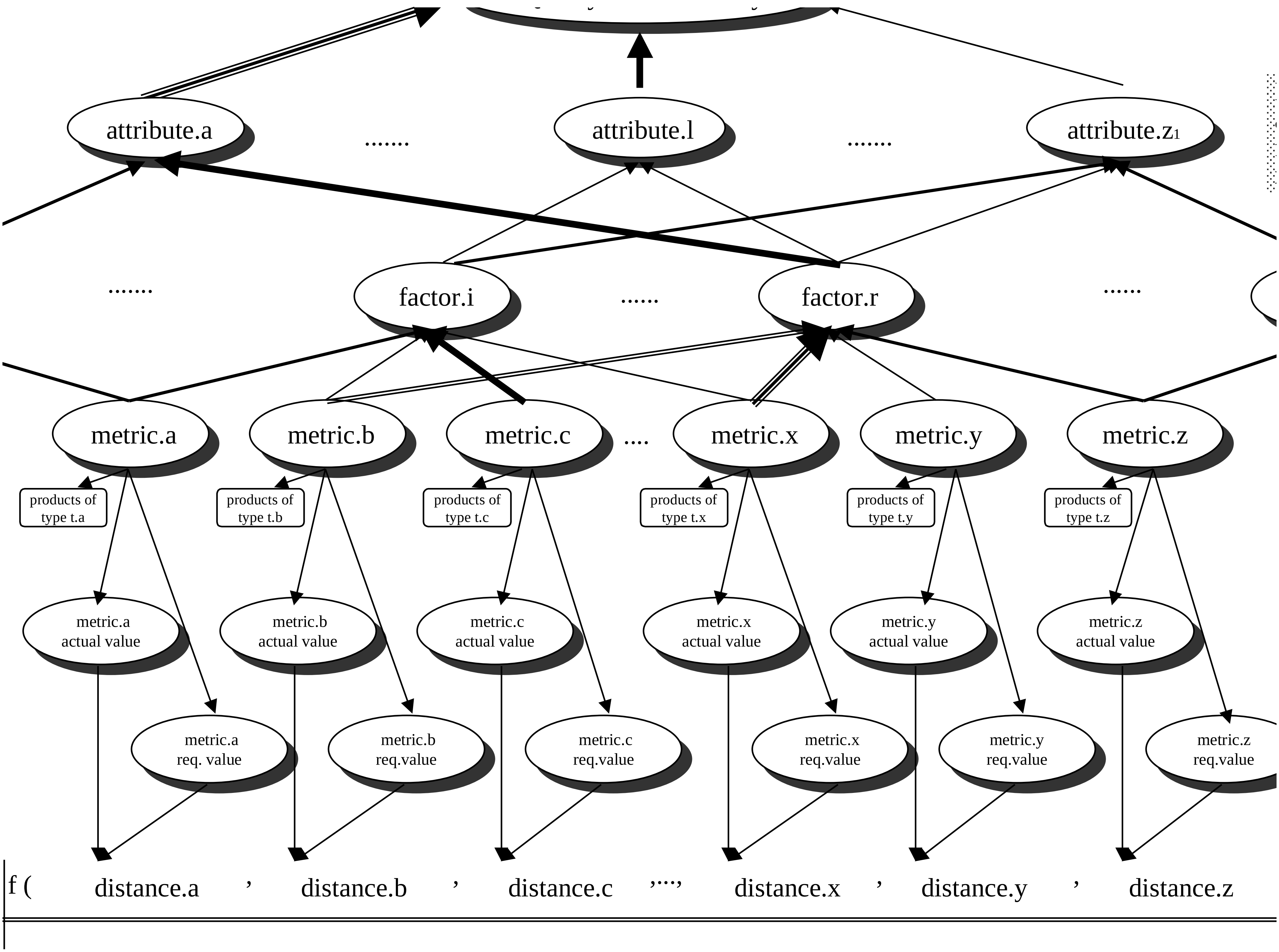


me-fct(software system attribute) α := some-fct(ub,m ulb Š mr Š lub then 1 else

llb Š mr Š lub then 1-(cos(mr-llb,ulb-mr) / ((ulb-llb

llb Š mr Š lub then 1-(sin(mr-lub,uub-mr) / ((uub-lub

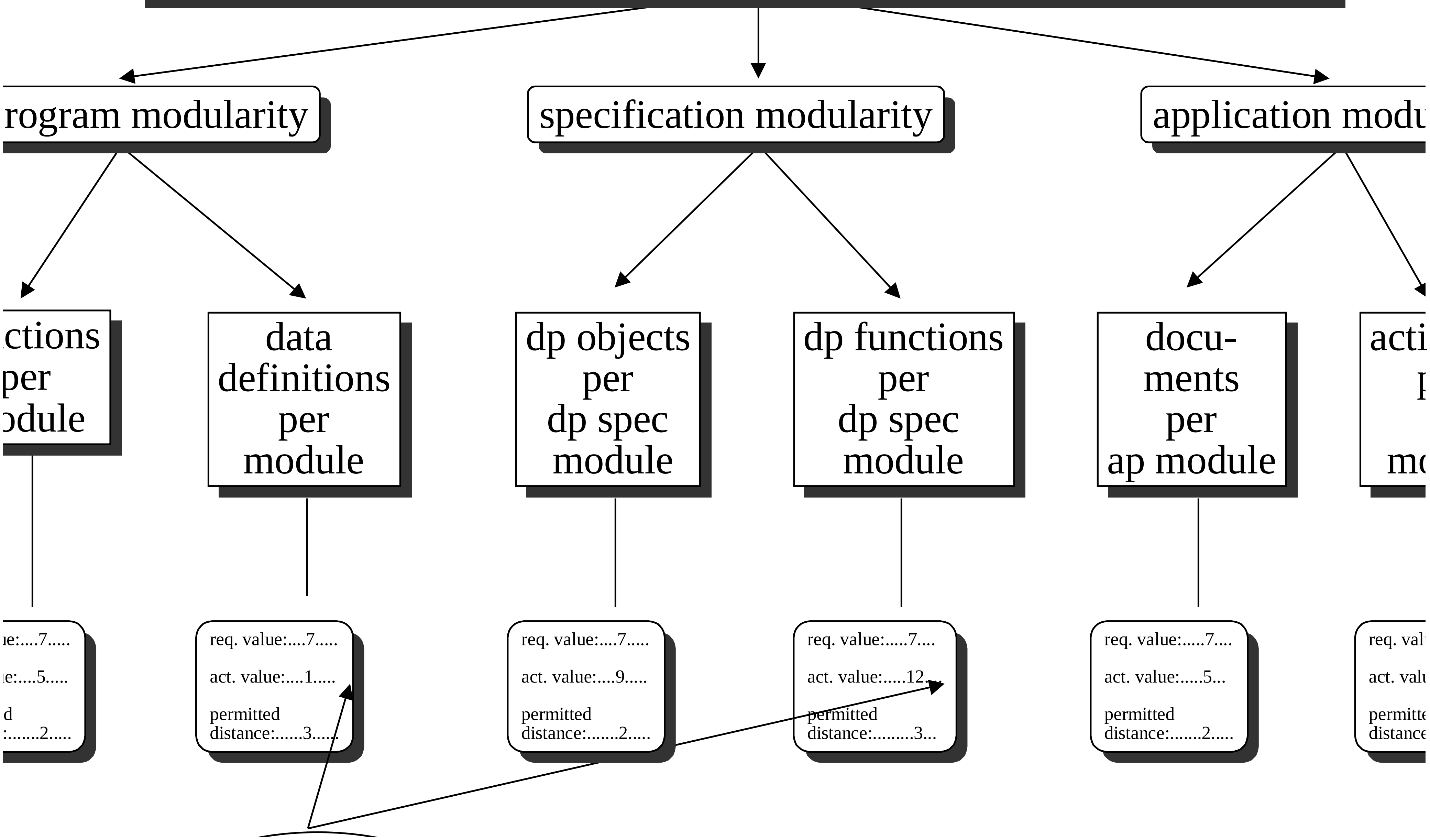



attribute.a
....
attribute.l
....
attribute.z1
....
factor.i
......
factor.r
......
metric.a
metric.b
metric.c
....
metric.x
metric.y
metric.z
products of type t.a
products of type t.b
products of type t.c
products of type t.x
products of type t.y
products of type t.z
metric.a actual value
metric.b actual value
metric.c actual value
metric.x actual value
metric.y actual value
metric.z actual value
metric.a req. value
metric.b req.value
metric.c req.value
metric.x req.value
metric.y req.value
metric.z req.value
f (distance.a , distance.b , distance.c ,..., distance.x , distance.y , distance.z

Modularity

:=

< program modularity, specification modularity, application modularity>



rogram modularity

specification modularity

application modu

ctions per odule

data definitions per module

dp objects per dp spec module

dp functions per dp spec module

docu- ments per ap module

acti p mo

ie:....7.....
ie:....5.....
d
:.......2.....

req. value:....7.....
act. value:....1.....
permitted
distance:......3......

req. value:....7.....
act. value:....9.....
permitted
distance:.......2.....

req. value:.....7....
act. value:.....12.....
permitted
distance:.........3...

req. value:.....7....
act. value:.....5...
permitted
distance:.......2.....

req. val
act. valu
permitte
distance

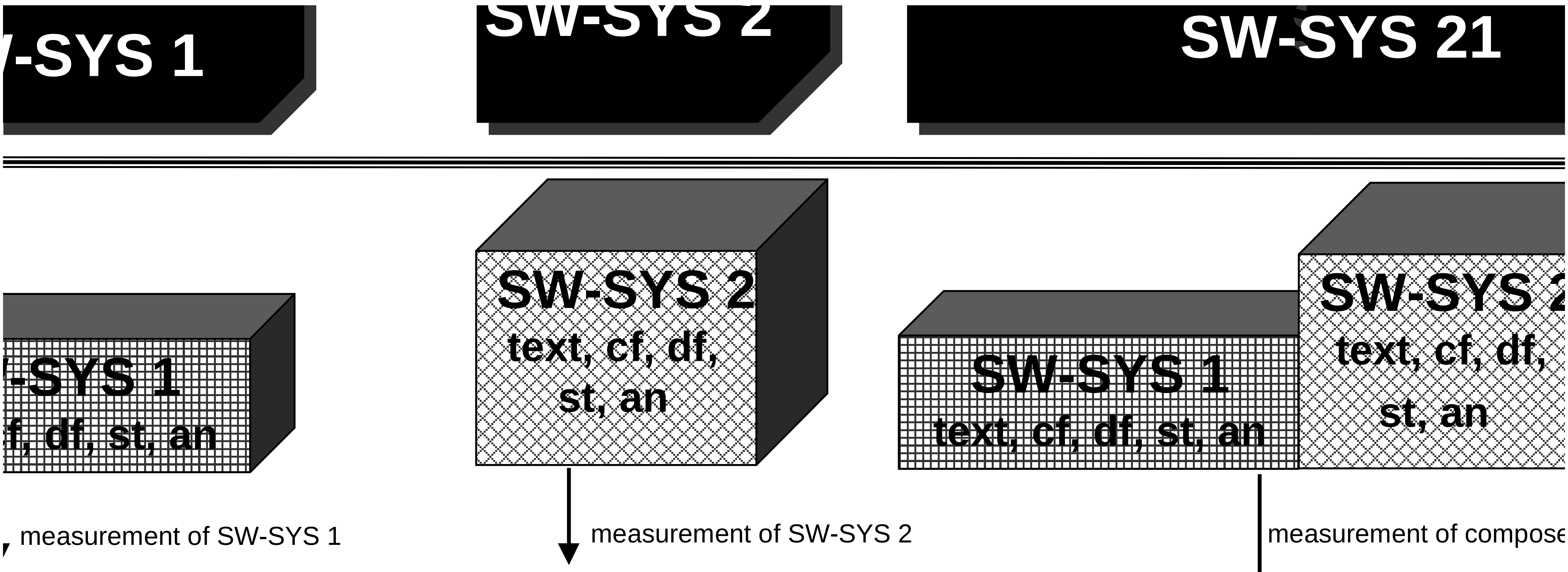
ll-du-Paths	**(ADUP)**
ll-Uses	**(AU)**
ll-p-Uses/Some-c-Uses	**(APU + C)**
ll-c-Uses/Some-p-Uses	**(ACU + P)**
ll-c-Uses	**(ACU)**
ll-Definitions	**(AD)**
ll-p-Uses	**(APU)**
ranch coverage	**(AB)**
tatement coverage	**(AS)**

Example: OO Metrics

© 1998 Hans-

- od Complexity
- nce Variables
- ber of Methods
- od size in source
- od Size in byte s
- s Heirarchy Depth

- Number of Direct Subclasses
- Total Number of Subclasses
- Method Access Permissions
- Method Comments
- Class Comments
- Total Number of Key/Support classe



l1   = 77777 loc
cfc1 =17 path/prg
dfc1 = 7 uses/definition
stc1   = 5 functions/transition
/ anc1 = 3 vocables/sentence
?
Program
?
?
(q1) = 87.654.321,50 DM

text length   l2   = 55555 loc
cf complexity cfc2 =11 path/prg
df complexity dfc2 = 5 uses/definition
st complexity stc2   = 7 functions/transition
an complexity anc2 = 5 vocables/sentence
volume v2 = ???
type   t2= Assembler-Program
illity   i2 = ???
quality q2 = ???
cost   C2 = f2(q2) = 12.345.678 DM

text length   l3   = l1 + l2 ???
cf complexity cfc3 = cfc1 * cfc2 ???
df complexity dfc3 = dfc1 ** dfc2 ???
st complexity stc3   = stc1 + stc2 ???

are metric should be
d with respect to
ematical description

-SYS 1

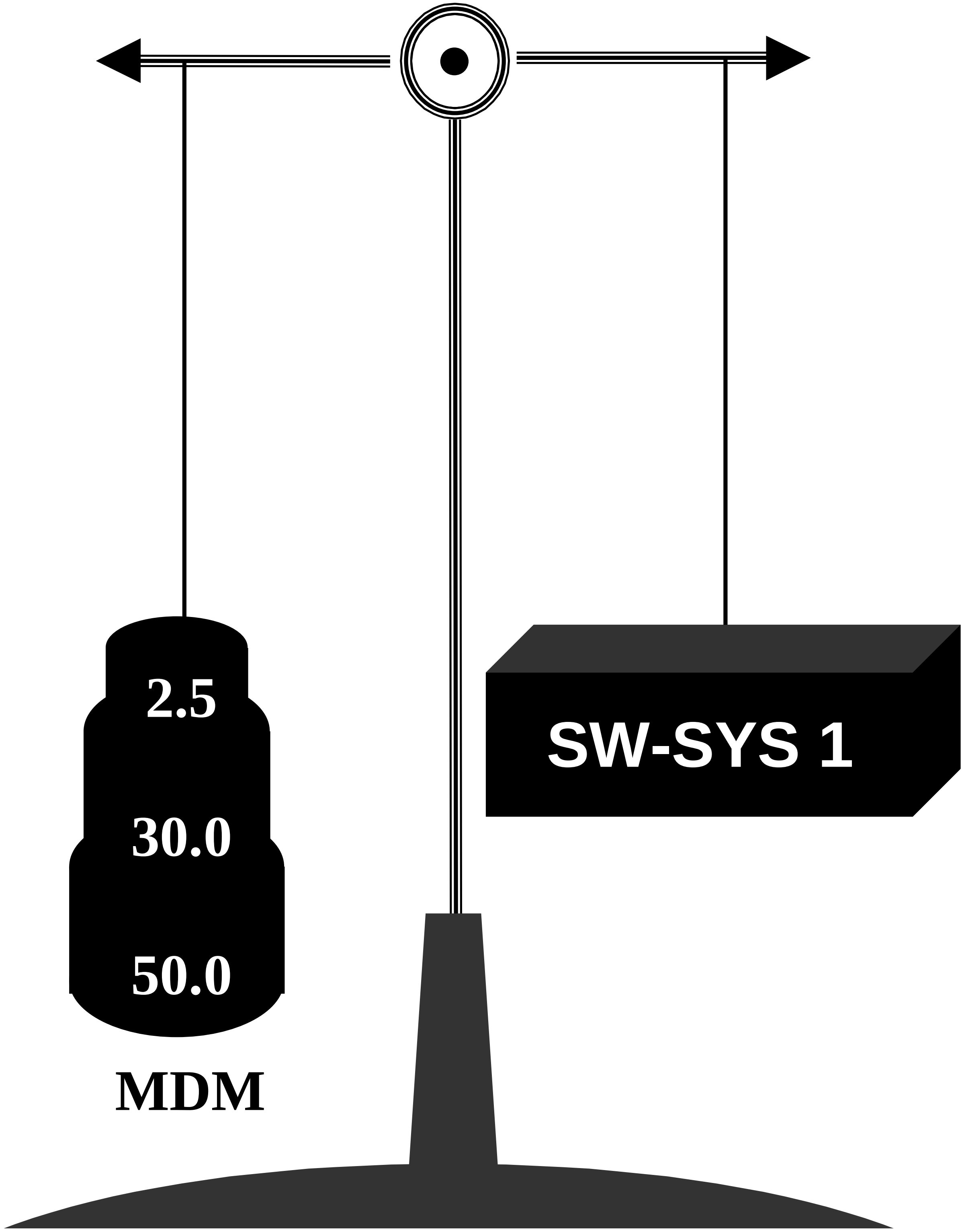
abstraction

-SYS 1
f, df, st, an

, measurement

= 77777 loc
cfc =17 path/prg
dfc = 7 uses/definition
stc = 5 functions/transition
anc = 3 vocables/sentence
?
Program
?
t(cfc,dfc,stc,anc,t,...) = ???
stfct(q) = 82.500.000,00 DM

n



r it, P(A) the set of all subsets of A, and let Z be the se

items, with relations $R_{z.1}$, ... , $R_{z.}$ and operations $o_{z.1}$,...

r it.

$_{a.1}$, ... , $R_{a.r}$, $o_{a.1}$,..., $o_{a.t}\rangle$ is called an emperical relati

$_{z.1}$, ... , $R_{z.r}$, $o_{z.1}$,..., $o_{z.t}\rangle$ is called a numerical relativ

$_{1}$, ... , R_{r}, o_{1},..., $o_{t}\rangle$ is called a rational relative

ng **m: P(**A**)** —> Z **is a r**

on **d:** A × A —> Z **is a n**

norphic mapping **s:** A —> Z **is a sc**

cost and benefit requires the mapping of A onto a rational relativ ation of a scale is a mapping of a scale into itself.

<u>scale</u> is a scale whose permitted transformations are only the **one**
ons.

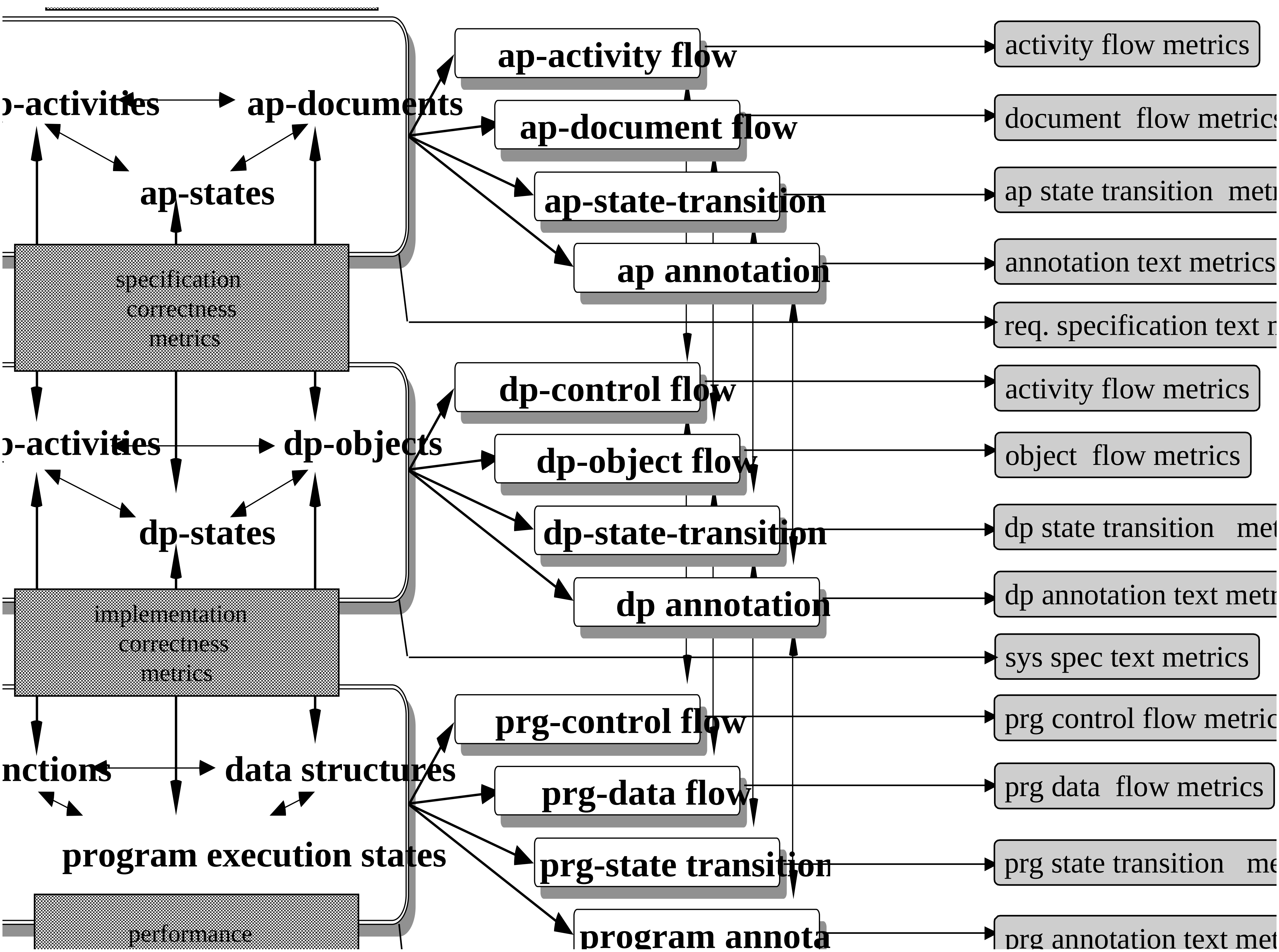
se permitted transformations are ***monotonic increasing*** is an **<u>or</u>**
c increasing transformation of a scale h is of the form **h—>f(h)**,
onic increasing, real-valued function.

se permitted transformations are ***positive linear*** is an **<u>interval sc</u>**
ar transformations of a scale h are of the form **h—> a** · **h + b,** w
a, b ∈ ℜ.

se permitted transformations are only the ***similarity transformat***
<u>o scale</u>. Thereby the similarity transformations of a scale h are o
where **a** > **0** and **a** ∈ ℜ**.**

es, interval scales, log-interval scales and ratio scales are also no

s possess a natural unit in addition to the fixed zero point, they a



p-activities
ap-documents
ap-states

specification
correctness
metrics

p-activities
dp-objects
dp-states

implementation
correctness
metrics

nctions
data structures
program execution states

performance

ap-activity flow
ap-document flow
ap-state-transition
ap annotation

dp-control flow
dp-object flow
dp-state-transition
dp annotation

prg-control flow
prg-data flow
prg-state transition
program annota

activity flow metrics
document flow metrics
ap state transition met
annotation text metrics
req. specification text m
activity flow metrics
object flow metrics
dp state transition met
dp annotation text metr
sys spec text metrics
prg control flow metric
prg data flow metrics
prg state transition me
prg annotation text met

$$= \#e - \#n + 2*(\#p)$$

otes the complexity of a program ('cyclomatic number complexity

e control structure represented by a graph G

ber of edges in the control graph

ber of nodes in the control graph

ber of connected components

tnod) * log$_2$(ndor + ndod)   = Volume

nod /2*ndod   = Density

nod * (tnor + tnod) * log$_2$(ndor + ndod) / 2*ndod

ming effort

er of distinct operators appearing in a program

er of distinct operands appearing in a program

umber of occurences of the operators in a program

umber of occurences of the operands in a program

logarithm

resented by T after the conversion to time units:

ming time of a program in seconds

umber, mean number of elementary mental

ations in the vocabulary, $5 \leq S \leq 20$ per second, usually $S = 18$

ow measure in a module m $\mathbf{INFO_m = (fi * fo)^2}$

ation measure in a module m $\mathbf{INFO\text{-}LOC_m = LOC_m * (fi * fo)^2}$

, fo -fan-out of a module, LOC_m -# lines-of-code of the module m .

ow measure for all modules in a call graph

$$\mathbf{INFO = \bullet_{i=1...n} (fi(i) * fo(i))^2},$$

umber of modules

ation measure for all modules in a call graph

$$\mathbf{INFO\text{-}LOC = \bullet_{i=1...n} LOC_m * (fi(i) * fo(i}$$

umber of modules in a call graph.

specification states for which requirements states fulfilment is verified / specification states
program execution states for which specification states fulfilment is verified / program execution states

programs symbolic executed correctly / programs

y

code predicates / code predicate variables
procedures / code predicates
(code variables - code predicate variables)/ code variables
procedures / variables
functions / data
activities / objects
activities / functions
objects / data
functions / procedures
data / variables
control flow complexity of the programs
data flow complexity of the programs
control flow complexity of the specification
data flow complexity of the specification
control flow complexity of the requirements specification
data flow complexity of the requirements specification
module connection complexity of the programs
module connection complexity of the specification
module connection complexity of the requirements specification
min(data-to-variable-links) / max(data-to-variable-links)
min(function-to-procedure-links) / max(function-to-procedure-links)
min(object-to-data-links) / max(object-to-data-links)
min(activity-to-function-links) / max(activity-to-function-links)
test predicates / test predicate variables
procedures / test predicates
(variables - test predicate variables) / variables
procedures / test predicate variables
(variables - test predicate variables) / test predicate variables

efficiency

transactions / (data processes per transaction -times- transactions)
transactions / (module calls per transaction -times- transactions)

redundancy

R.1 = repeatable modules / modules
R.2 = reproducible data capsules / modules
R.3 = logged transactions / transactions

integrity

I.1 = edited system input data items / system input data items
I.2 = edited system output data items / system output data items

generality

G.1 = application independent modules / modules
G.2 = application independent procedures / procedures
G.3 = application independent variables / variables
G.4 = application independent functions / functions
G.5 = application independent data / data
G.6 = application independent activities / activities
G.7 = application independent objects / objects

portability

P.1 = environment independent modules / modules
P.2 = environment independent procedures / procedures
P.3 = environment independent variables / variables
P.4 = environment independent functions / functions
P.5 = environment independent data / data
P.6 = environment independent activities / activities
P.7 = environment independent objects / objects

test coverage

TC.j.i = programs C.i tested / programs
TC. ... = programs C. ... tested / programs
TC.k.i = modules S.i tested / modules
TC. ... = modules S. ... tested / modules

inspection coverage

IC.1 = programs accepted after inspection / programs

M.1 = modu
M.2 = modu
M.3 = modu
M.4 = modu
M.5 = modu
M.6 = modu
M.7 = modu
M.8 = modu
M.9 = modu

specification states for which requirements states fulfilment is verified / specification states
program execution states for which specification states fulfilment is verified / program execution states

programs symbolic executed correctly / programs

y

code predicates / code predicate variables
procedures / code predicates
(code variables - code predicate variables)/ code variables
procedures / variables
functions / data
activities / objects
activities / functions
objects / data
functions / procedures
data / variables
control flow complexity of the programs
data flow complexity of the programs
control flow complexity of the specification
data flow complexity of the specification
control flow complexity of the requirements specification
data flow complexity of the requirements specification
object connection complexity of the programs
object connection complexity of the specification
object connection complexity of the requirements specification
min(data-to-variable-links) / max(data-to-variable-links)
min(function-to-procedure-links) / max(function-to-procedure-links)
min(object-to-data-links) / max(object-to-data-links)
min(activity-to-function-links) / max(activity-to-function-links)
test predicates / test predicate variables
procedures / test predicates
(variables - test predicate variables) / variables
procedures / test predicate variables
(variables - test predicate variables) / test predicate variables

efficiency

transactions / (data processes per transaction -times- transactions)
transactions / (object calls per transaction -times- transactions)

redundancy

R.1 = repeatable objects / objects
R.2 = reproducible data capsules / objects
R.3 = logged transactions / transactions

integrity

I.1 = edited system input data items / system input data items
I.2 = edited system output data items / system output data items

generality

G.1 = application independent objects / objects
G.2 = application independent procedures / procedures
G.3 = application independent variables / variables
G.4 = application independent functions / functions
G.5 = application independent data / data
G.6 = application independent activities / application activities
G.7 = application independent objects / application objects

portability

P.1 = environment independent objects / objects
P.2 = environment independent procedures / procedures
P.3 = environment independent variables / variables
P.4 = environment independent functions / functions
P.5 = environment independent data / data
P.6 = environment independent activities / activities
P.7 = environment independent objects / objects

test coverage

TC.j.i = programs C.i tested / programs
TC. ... = programs C. ... tested / programs
TC.k.i = objects S.i tested / objects
TC. ... = objects S. ... tested / objects

inspection coverage

IC.1 = programs accepted after inspection / programs

modularity

M.1 = objec
M.2 = objec
M.3 = objec
M.4 = objec
M.5 = objec
M.6 = objec
M.7 = objec
M.8 = objec
M.9 = objec

development effort planning model offers complexity

rs can map their proposed developments to.

$$= \alpha \cdot (\,\text{KDSI}\,)^{\beta}$$

= programmer months effort

= complexity coefficient

= complexity exponent

= estimate of thousands of delivered lines o

ity Level	α	β
on	2.4	1.05
	3.0	1.12
	3.6	1.20

$$= \tau \cdot (\sigma(\mu_1(\text{code}), \mu_2(\text{specs}), \mu_3(\text{reqs}$$

$$, \qquad \sigma\colon \Re \times \Re \times \Re \to \Re$$

$$\text{E} \to \Re, \ \mu_2\colon \text{SPECS} \to \Re, \mu_3\colon \text{REQS} \to \Re,$$

$${}_{\text{e-company}} = 1.75 \cdot (\#\text{locode} + \#\text{lospecs} + \#\text{lore}$$

$$= \alpha \cdot (\#\text{lines-of-code})^{\beta}$$

$$\text{Effort-Estimate}_{\text{funtion-points}} = \gamma \cdot (\#\text{funct}$$

$$\text{where} \quad \#\text{function-points} = f(\#\text{loreq}) =$$

prsdt / nfr = ___ hours/reqspec = 5.5 hours/reqspec ???

hours/reqspec post-requirements-specification time are required to fully implement each uirement spec

dt = post-requirements-specification development time

dt = sum of design, implementation, testing, and documentation hours

= number of functional requirements in the requirements specification

st / prsdt = ____ = 1 / 4 ???

requirements specification time / post-requirements-specification development time

---

ther words:

ox. 7 requirements specs per week can be fully mplemented, tested, documented, etc. (schedule nds upon the lifecycle model being used, ofcourse)

Requirements specs are written according IEEE SRS standard ss = sst / nfr = _____ hours/reqspec = 3 hours/reqspec ???

= software specification time

= time for writing, testing, documenting software specification

= number of functional requirements in the requirements specification

ss = rst / sst = 1 / ___ = 1 / 2 ???

= requirements specification time / software specification ti

ec = ect / nfs = _____ hours/swspec = 3 hours/swspec ???

= executable code time

= time for writing, testing, documenting executable code

= number of functional specifications in the software specification

ec = sst / ect = 1 / ___ = 1 / 2 ???

requirements specification time / executable code development time t / nfr , prsqat / nfr , prcmdt / nfr , prspmt / nfr,

rsqat / nfr , rcmdt / nfr , rspmt / nfr >

st-requirements-specification development time

st-requirements-specification quality assurance time

ost-requirements-specification configuration management time

ost-requirements-specification project management time

quirements-specification quality assurance time

equirements-specification configuration management time

quirements-specification project management time

umber of functional requirements in the requirements specification

/ prsdt =1/___ t.o.qa = rst / prqat =1/___

/ prscmt =1/___ t.o.pm = rst / prspmt =1/___

f development ratio

f quality assurance ratio

configuration management ratio

f project management ratio

ements specification time

e Representations:

red Feature Points / Staff Month

- 2.0

as - 1.6

- 1.4

< 1.0

red defects / Feature Point

- 0.3

as - 0.7

- 0.8

> 1.0

etrics for
am text
fication text
rements text
ation text
natural language documentation text

Graph metrics for

- control flow graphs
- data flow graphs
- state transition diagrams
- module interconnection graphs

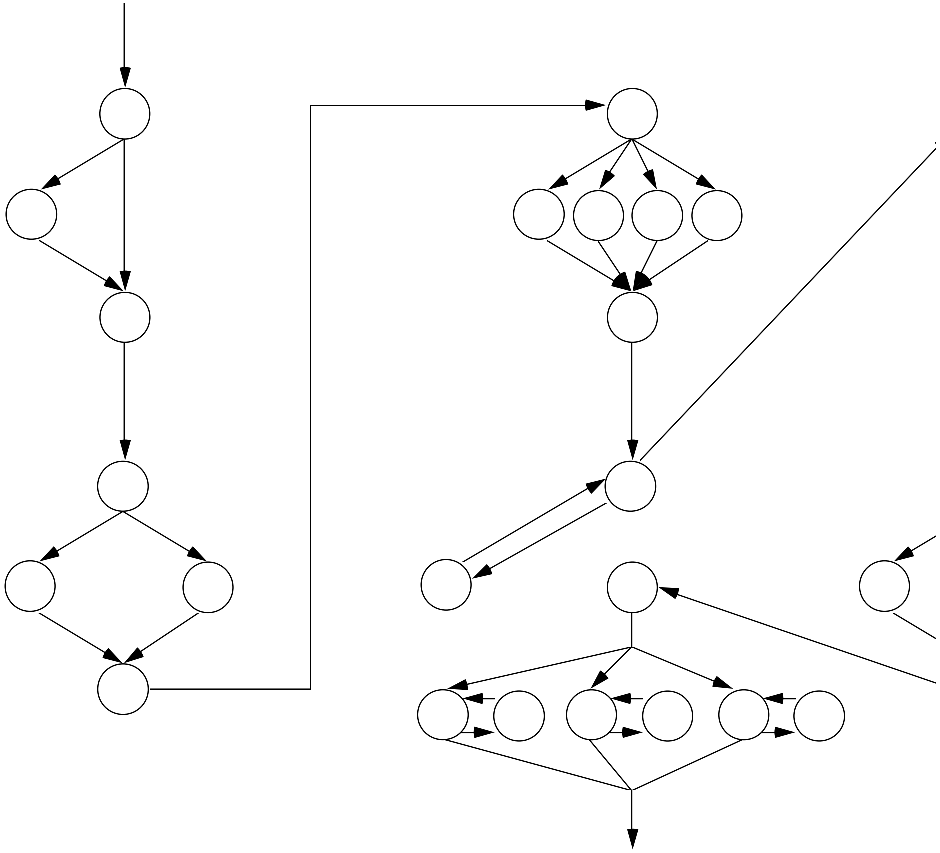
Check-lists for obtaining

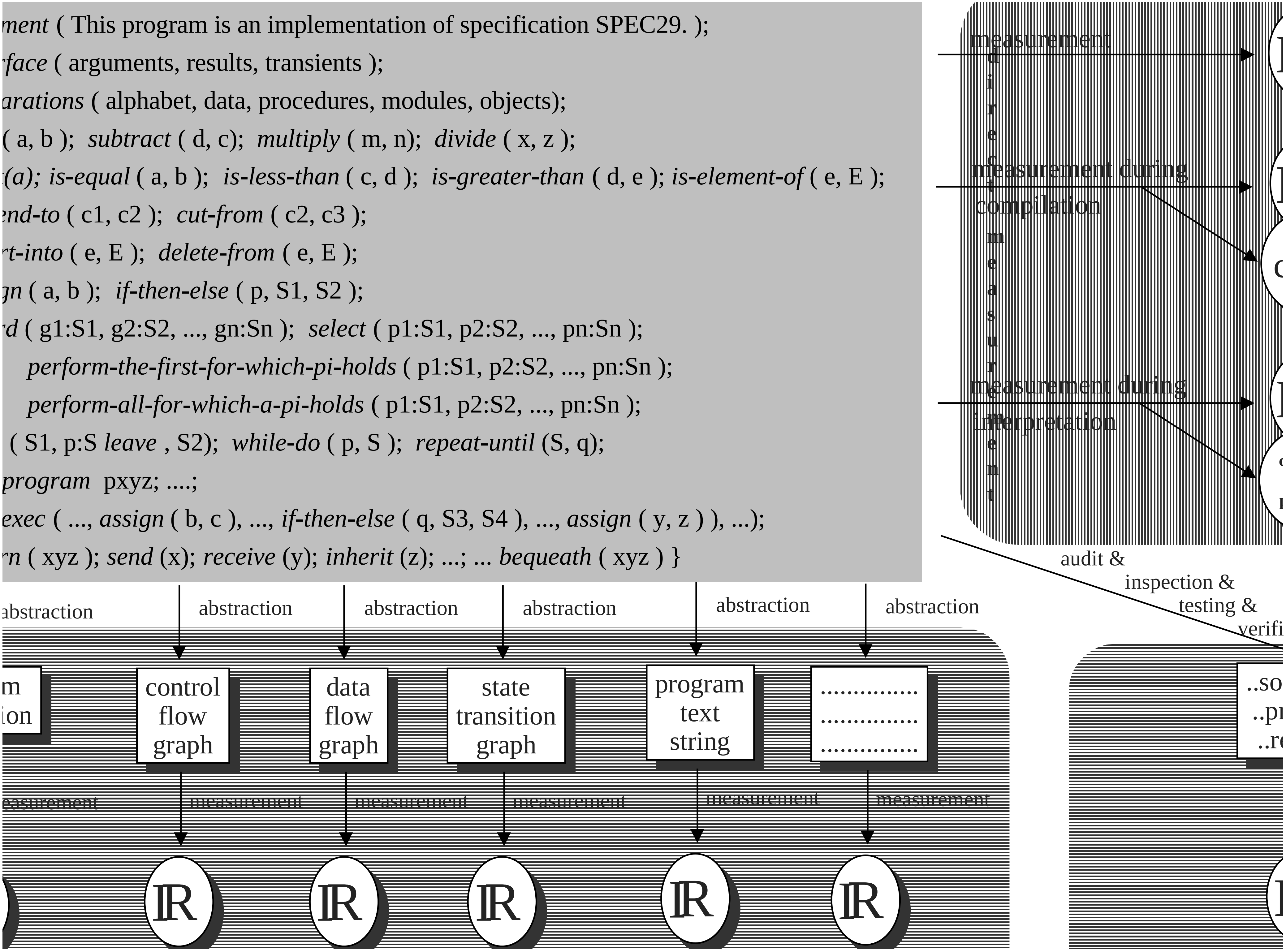
- alternative evaluation answers
- multiple choice results

be measured according to the
the language the text is written in
measured according to the grammar,
rogramming language the

equency might be measured)

to be measured according to the
the basic components the graph is build with
is measured according to the edges, nodes,
nposition rules
directed graph the
nodes,
les and nestings,
requency might be measured)





ment (This program is an implementation of specification SPEC29.);

rface (arguments, results, transients);

arations (alphabet, data, procedures, modules, objects);

(a, b); *subtract* (d, c); *multiply* (m, n); *divide* (x, z);

(a); is-equal (a, b); *is-less-than* (c, d); *is-greater-than* (d, e); *is-element-of* (e, E);

end-to (c1, c2); *cut-from* (c2, c3);

rt-into (e, E); *delete-from* (e, E);

gn (a, b); *if-then-else* (p, S1, S2);

rd (g1:S1, g2:S2, ..., gn:Sn); *select* (p1:S1, p2:S2, ..., pn:Sn);

perform-the-first-for-which-pi-holds (p1:S1, p2:S2, ..., pn:Sn);

perform-all-for-which-a-pi-holds (p1:S1, p2:S2, ..., pn:Sn);

(S1, p:S *leave* , S2); *while-do* (p, S); *repeat-until* (S, q);

program pxyz; ....;

exec (..., *assign* (b, c), ..., *if-then-else* (q, S3, S4), ..., *assign* (y, z)), ...);

rn (xyz); *send* (x); *receive* (y); *inherit* (z); ...; ... *bequeath* (xyz) } ram:= cfc statement

nt := < S1 <; or ||> S2 <; or ||> ... <; or ||> Sn >

nent = f statement (cfc $_{S_1}$, ... , cfc $_{S_n}$)

n := f assign (cfc left hand side, cfc right hand side)

n := (cfc left hand side + cfc right hand side)

hand side := case

rithmetic expression : cfc arithmetic~expression

oolean expression : cfc boolean~expression

en-else :=f if-then-else (cfc cond-part , cfc then-part ,

two ways of
implementin
measuremen

- generate meas
into a compil
compiler-co
(prototype ba
Lex and Yac

- enhance interp
rules of an in

set of programs,

ons *equal*, *less*, *more* and operations *sequencial copos*
mposition, *nested composition*, defined for it

nore-or-equal; *sequence*, *parallel*, *nest*, *a*
oop, *guard*, *fork*, *join*〉

; °, ||, •, :=, ite, lup 〉 is emperical relative fo

• ; +, -, *, /, ** 〉 is correspondig numerical r

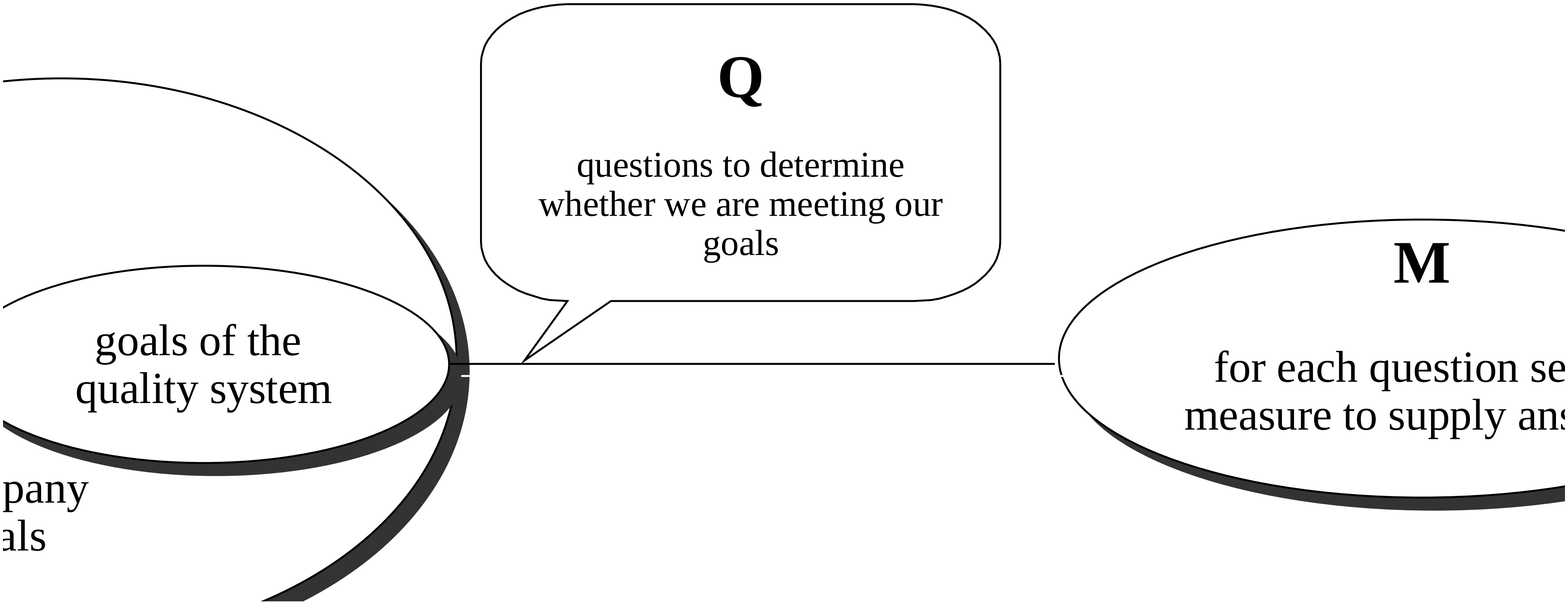
ch emperrical relation a correspondig numerical relation and

ch emperical operation a corresponding numerical operation

the goals of the quality system!
end goal and relation to the overall company business goals?

questions?
ation do we need to know (what questions do we need to ask) to determine if we are me

question, what measure can we take to supply answers to the questions?



Q

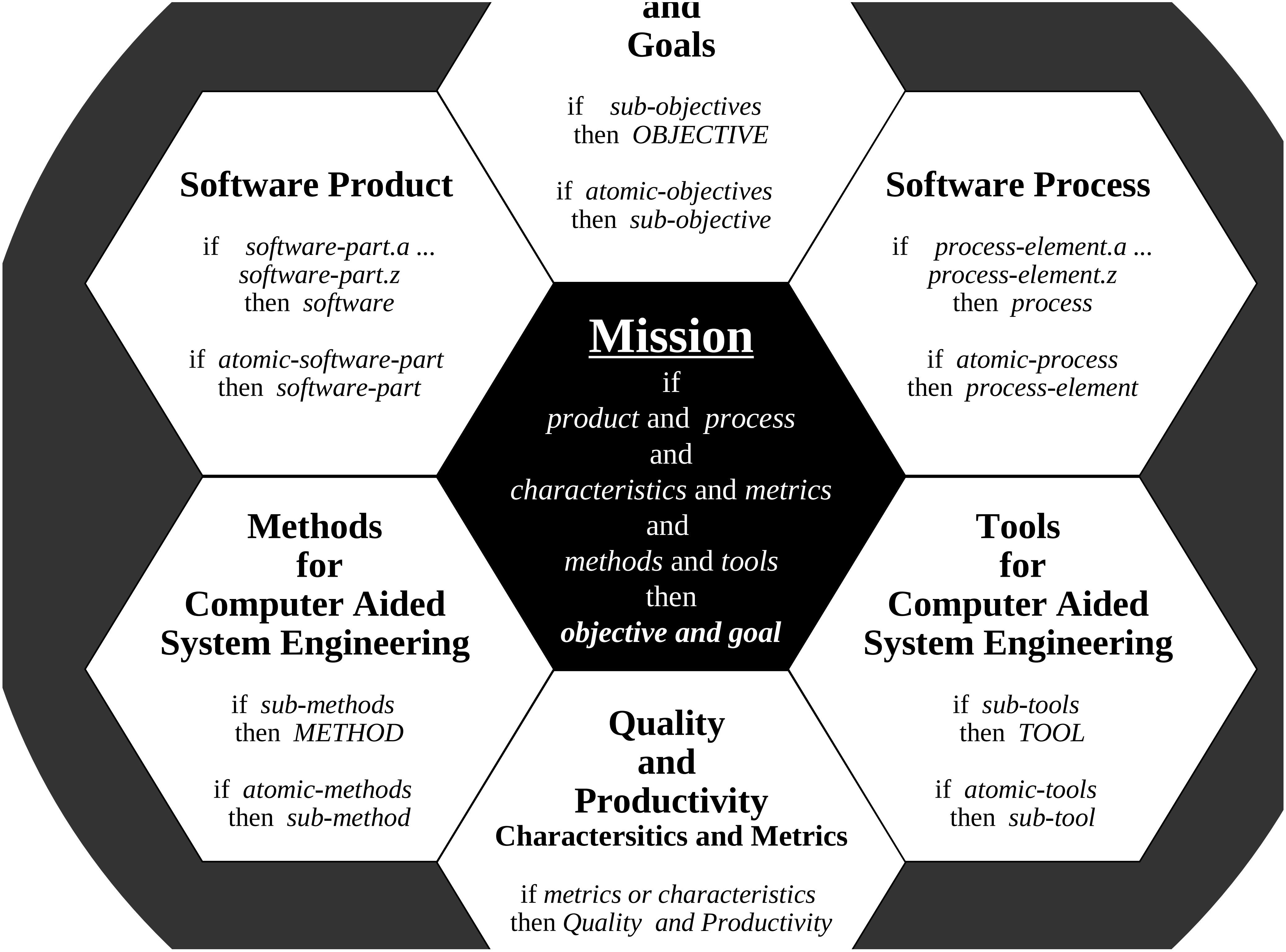
questions to determine whether we are meeting our goals

goals of the quality system

pany
als

M

for each question se
measure to supply ans



and
Goals

if *sub-objectives*
then *OBJECTIVE*

if *atomic-objectives*
then *sub-objective*

Software Product

if *software-part.a ...*
software-part.z
then *software*

if *atomic-software-part*
then *software-part*

Software Process

if *process-element.a ...*
process-element.z
then *process*

if *atomic-process*
then *process-element*

Mission

if
product and *process*
and
characteristics and *metrics*
and
methods and *tools*
then
objective and goal

Methods
for
Computer Aided
System Engineering

if *sub-methods*
then *METHOD*

if *atomic-methods*
then *sub-method*

Tools
for
Computer Aided
System Engineering

if *sub-tools*
then *TOOL*

if *atomic-tools*
then *sub-tool*

Quality
and
Productivity
Charactersitics and Metrics

if *metrics or characteristics*
then *Quality and Productivity*

Divide each procedure in Quality Manual into three se

ory - These practises are adequate to meet XYZ ation requirements, and are meant to be assessed l auditors.

nended - This is the internal target which all ation members are expected to meet.   These wil ed in the Internal Audit and an "Internal non- nance " might be raised.

actises - Employees who implement some of thes es will be rewarded for adhering to these best pra

ractises will gradually become recommended and recomme

onder how to get t all the way full.”