TITLE:	SPECIFYING SYSTEM SECURITY REQUIREMENTS
INSTRUCTOR:	Paula A. Moore
OTHER SPEAKERS:	None
SUMMARY:	This tutorial presents systematic approaches for developing system security requirements and provides structures and criteria for generating them. It explores: ▪ Focus and styles for requirements based on the type of audience, ▪ Types of requirements and requirement look-alikes, ▪ Characteristics of effective technical and assurance requirements, ▪ Organization of requirements based on specification styles, and ▪ Pitfalls in constructing individual requirements. Examples of good and poor security requirements are used throughout.

PAULA A. MOORE

Paula has been a computer scientist with the FAA for five years, primarily as the Security Lead for a joint FAA/DoD air traffic control system. Her work there has included security risk assessments, security requirements definition and policy development. Previously she was a systems engineer at NOAA performing IV&V and Software Capability Evaluations. Before Government service, Paula spent four years as a senior software engineer at Loral Aerosys responsible for software requirements on the Hubble Telescope Data Archive. She designed and led development for database cataloging and high-speed ingest of telescope images. She also spent five years developing specifications and database applications for transaction processing in the banking industry. She has also taught physics and math in Baltimore County Public Schools.

Paula has a Bachelor's Degree in Physics Education from University of Maryland, a Master's in Engineering Science In Computer Science from Loyola College of Baltimore, and additional post-graduate education.

Specifying System Security Requirements

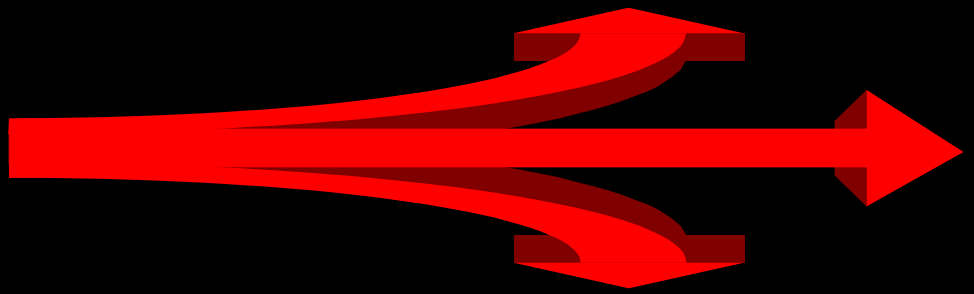
a tutorial

Presented by Paula A. Moore

22nd National Information Systems Security Conference

A Security Requirements Roadmap

- Part I: Who's requirements are they?
- Part II: What could they look like?
- Part III: What should they accomplish?
- Part IV: How can they be organized?
- Part V: What's a good requirement?



Part I

Who's Requirements Are They?

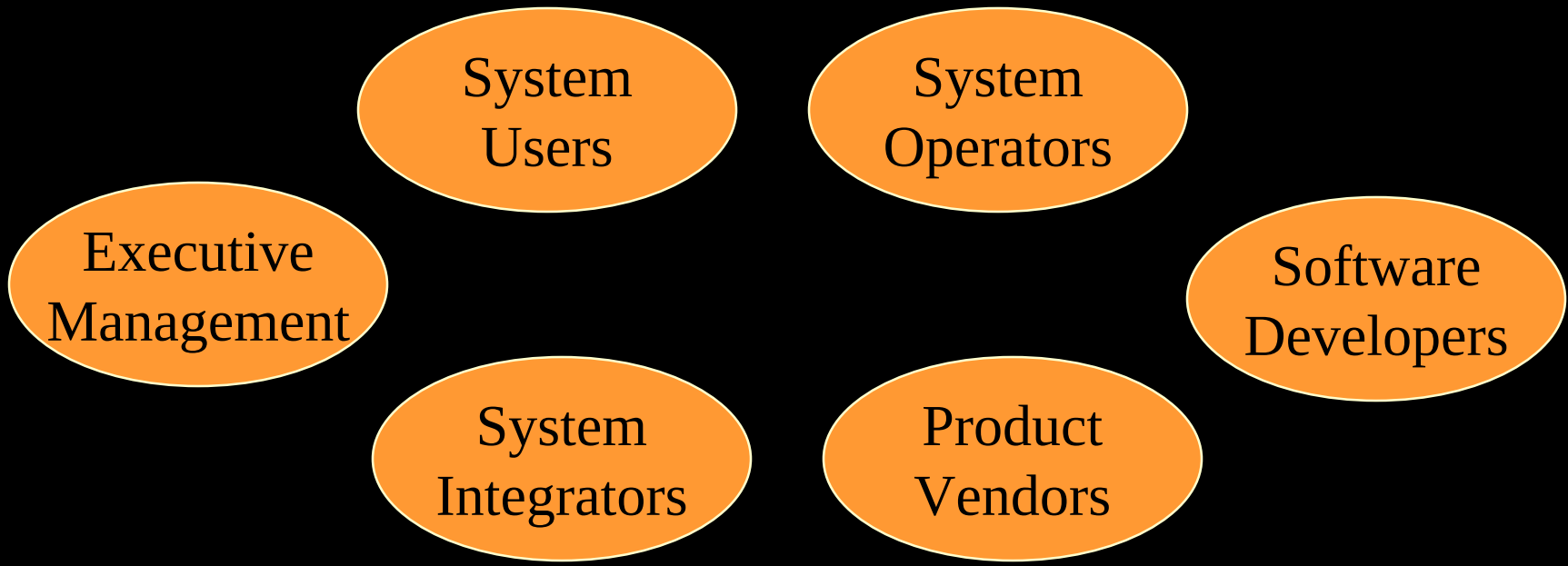
What Ownership Means

The players involved with requirements development drive:

- How much and what kind of security
- The trade-offs among technical, physical, personnel and procedural security
- The cost of the system
- Satisfaction with the system at delivery and beyond

Who's Requirements Are They?

This is the most critical question to answer before writing even the first requirement!



Audiences and Their Interests

System Users
- Meeting the Mission
- Time on Task
- Training Impact

Software Developers
- Schedule
- Complexity
- Allocation
- Test Effort

System Integrators
- Product Applicability
- Allocation
- Test Effort
- Customer Satisfaction

System Operators
- Effectiveness
- Time on Task
- Training Impact
- Reliability

Product Vendors
- Product Applicability
- Support Impacts

Executive Mgmt
- Cost
- Schedule
- Effectiveness
- Strategy

Consequences

- If your audience doesn't understand your requirements, you won't get the system you expect.

  You may never get the system!

- The more audiences for your requirements, the greater the chance that the system will meet no one's expectations.

- If the requirements are not understood and endorsed by your audience(s), operational security will suffer.

Whose Requirement?

A. Unauthorized activity on operational systems shall be continuously monitored, recorded and reported. — Executive Management

B. The unauthorized activity monitor shall be physically protected from unauthorized access and eavesdropping. — System Integrators

C. Recording of unauthorized activity shall be initiated and terminated by an authorized security administrator through both command line and graphical interfaces. — System Operators

D. Recording of unauthorized activity shall be initiated within 50 ms of completion operator input request. — Vendors & Developers

E. The recording mechanism for unauthorized activity shall provide an upgrade path for alternate types of media. — Vendors & Developers

Rules for Requirements Mechanics

Get written agreements on organizational and individual responsibility for:

- ⇨ Creating and maintaining requirements statements
- ⇨ Confirming the legitimacy of requirements
- ⇨ Interpreting requirements
- ⇨ Determining whether tests are adequate
- ⇨ Deciding whether a function as tested meets requirements

Part II

What Could They Look Like?

One Person's Requirement is Another Person's ...

A. Unauthorized activity on operational systems shall be continuously monitored, recorded and reported.

B. The unauthorized activity monitor shall be physically protected from unauthorized access and eavesdropping.

C. Recording of unauthorized activity shall be initiated and terminated by an authorized security administrator through both command line and graphical interfaces.

D. Recording of unauthorized activity shall be initiated within 50 ms of completion operator input request.

E. The recording mechanism for unauthorized activity shall provide an upgrade path for alternate types of media.

Security Requirements Relationships

What's Wrong With This Picture?

- Policy -- "an informal, generally natural language description of desired system behavior."
- Requirement -- "A statement of the system behavior needed to enforce a given policy. Requirements are used to derive the technical specification of a system."
- Specification -- "A technical description of the desired behavior of a system, as derived from its requirements. A specification is used to develop and test the implementation of a system."

Source: Longley, Shain & Caelli, Information Security. Stockton Press, New York, 1992.

Security Policy

Criteria *

- Identifies what is valuable (assets)
- Identifies the steps to safeguard assets
- Assigns responsibility for protections
- Assigns responsibility for policy changes
- Defines the structure for applying policy

Qualities *

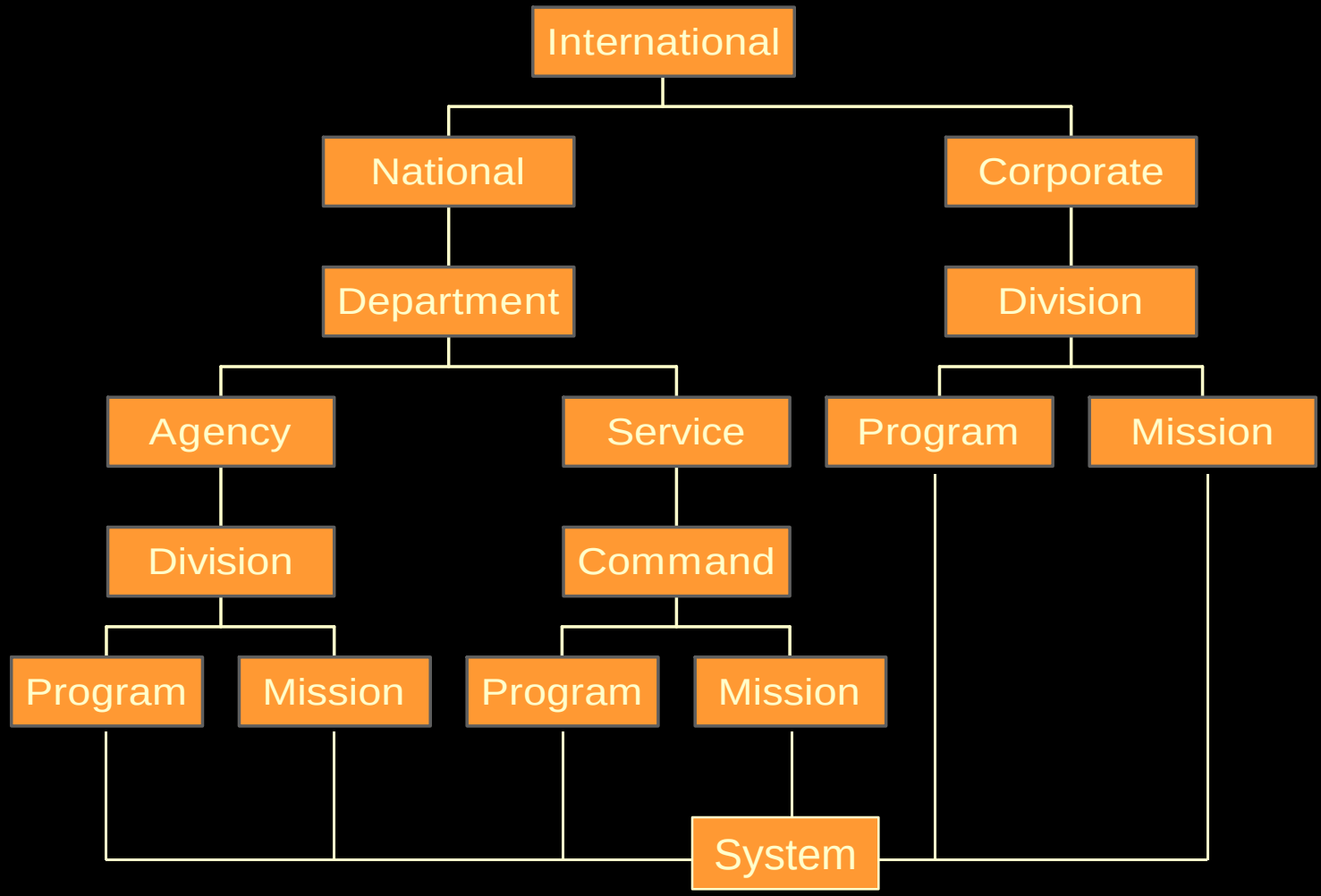
- Brief
- Understandable
- Durable

Example

Each employee is responsible for protecting from unauthorized disclosure or use the information and materials that are required to access protected company assets.

* *Source: Garfinkel and Spafford, Practical UNIX and Internet Security. O'Reilly & Associates, Inc., Sebastopol, CA, 1996.*

The Many Layers of Policy



Technical Requirements

Criteria *

- Observable behaviors of system operation
- Sequencing and timing of system behaviors
- Control and interaction of system behaviors
- Standards and guidance for system construction
- Allocation of behaviors to system components

Qualities *

- Correct
- Unambiguous
- Complete
- Verifiable
- Consistent
- Traceable
- Maintainable

** Source: Davis, Software Requirements: Analysis and Specification. Prentice-Hall, Englewood Cliffs NJ, 1990.*

Technical Requirements (continued)

Example: Architecture Standard

Standard ports shall be used for all TCP and UDP services as specified in RFC 1700.

Example: Functional Requirement

The TSF shall detect when an operator-adaptable number of unsuccessful authentication attempts occur related to: logon to any system account or initiation of a remote connection. *[FIA_AFL.1.1]*

Example: Performance Requirement

The TSF shall enforce a maximum quota for the total hard disk space simultaneously utilized by security audit logs on any individual disk partition. *[FRU_RSA.2.1]*

Example: Quality of Supportability

Security functions shall be portable among UNIX environments.

Assurance Requirements

Criteria *

- Demonstration of functional and performance requirements
- Sampling and inspection of qualities and standards
- Analysis and verification of logic and algorithms
- Traceability of requirements through design, implementation and verification
- Management of components and activities
- Security flaw identification and evaluation

Qualities *

- Scope
- Depth
- Rigor

* *Source: ISO/IEC, Common Criteria for Information Technology Security Evaluation, Version 2.0, Part 3: Security Assurance Requirements, 1998.*

Assurance Requirements (continued)

Example: Analysis and Test

The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure obvious vulnerabilities have been addressed. *[AVA_VLA.1.2E]*

Example: Documentation

The documentation of the development tools shall unambiguously define the meaning of all implementation-dependent options. *[ALC_TAT.1.3C]*

Example: Assurance Planning

The Assurance Maintenance Plan shall describe the assurance maintenance cycle, stating and justifying the planned schedule of assurance maintenance audits and the target date of the next re-evaluation of the Target of Evaluation. *[AMA_AMP.1.6C]*

Part IV

How Can They Be Organized?

Collections of Requirements

Specification:

Documentation containing a precise, detailed, verifiable description of particulars with respect to … characteristics of a system or system component.

* *Source: Evans & Marciniak, Software Quality Assurance and Management.. John Wiley & Sons, Inc., New York NY, 1987.*

Some Spec Fundamentals . . .

For each requirement statement, provide a:

- √ Unique identifier
- √ Title
- √ Level indicator
- √ Security tag
- √ Trace to policy

14.6.2 *Mail Guarantee.* The *Security Functions shall enforce the generation of evidence of receipt for electronic mail. [NR-3.6]*

Should Security Stand Out?

Separation of security from other requirements supports:

- Visibility to audience
- Improved verification
- Internal consistency
- Ease of requirements maintenance and upgrade
- COTS product identification and selection

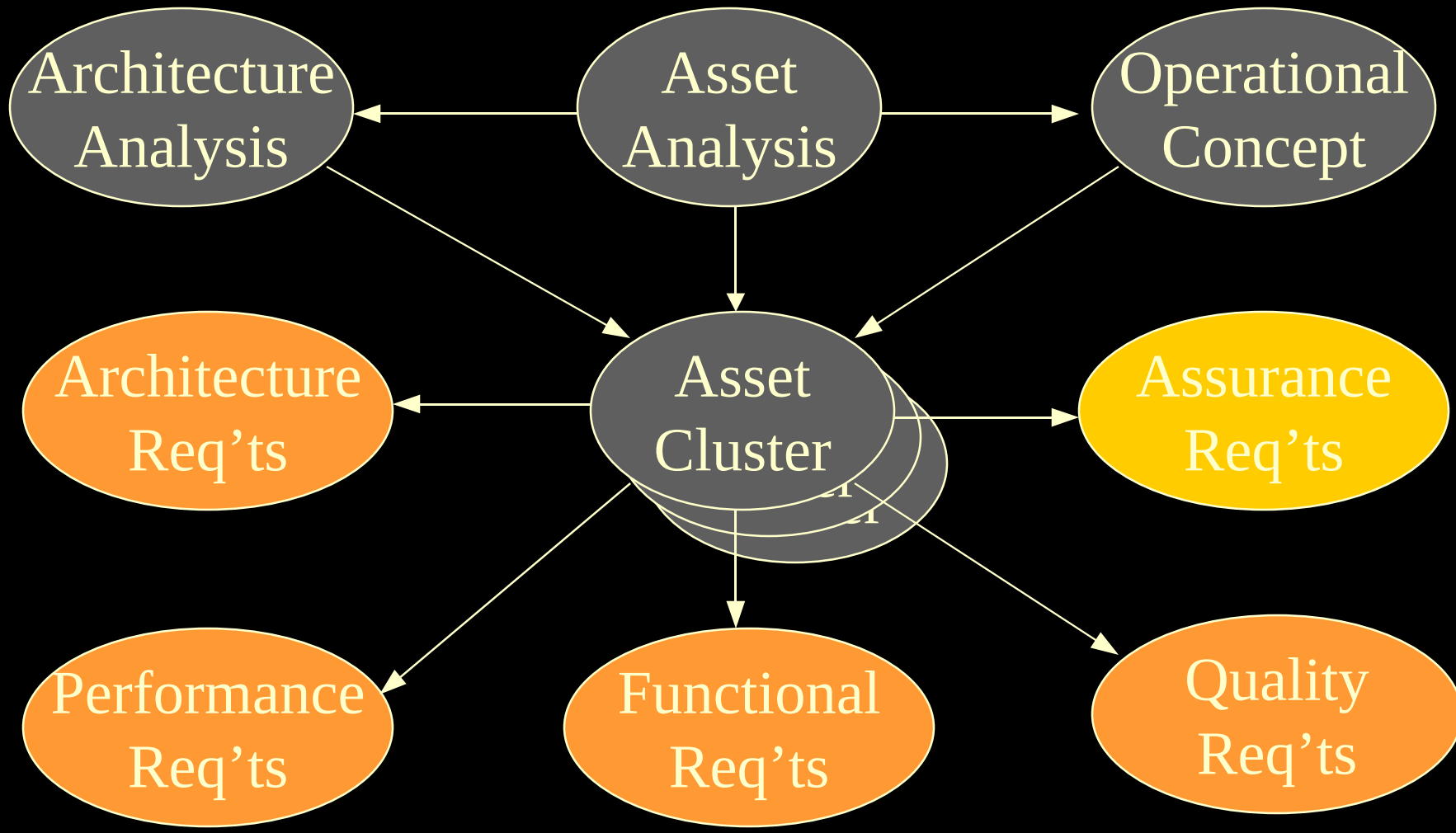
Integration of security with other requirements supports:

- Clearer understanding of security operations
- Saleability of security as a system function
- Consistency with other system functions
- Design, implementation and integration processes

Some Specification Dimensions

Segment	System-wide, Subsystem, Subnet, Site-adaptation, …
Service	Access control, Encryption, Non-repudiation, Integrity, …
Type	Architectural, Functional, Performance, Quality
Timing	Operational Threads of Activity: Logon, System Startup, File Transfer
Level of Detail	System, Subsystem, Platform, Application or Utility, …
Risk Impact	Information sensitivity, Implementation priority
Confidence	Technology-limited, COTS-limited, User-essential, User preference, …

Sensitivity-Driven Analysis



Sensitivity-Driven Structure

1.0 Electronic Mail

2.0 Web Services

- 2.S.1 Web Security Architecture
- 2.S.2 Web Security Functions
- 2.S.3 Web Security Performance
- 2.S.4 Web Security Qualities
- 2.S.5 Web Security Assurance

3.0 Personnel Data Management

4.0 Software Development

- 4.S.1 Development Security Architecture
- 4.S.2 Development Security Functions
- 4.S.3 Development Security Performance
- 4.S.4 Development Security Qualities
- 4.S.4 Development Security Assurance

5.0 System Security Management

. . .

X.0 Common Security Assurance

Requirement Subtypes

Architecture	Protocols Software Architecture Software-to-Hardware Allocation
Function	Input Storage Transformation Output Control
Performance	Capacity Throughput Response Time
Quality	Reliability Useability Maintainability Supportability Portability

More Subtypes: Davis, Software Requirements: Analysis and Specification. Prentice-Hall, Inc., Englewood Cliffs NJ, 1990. Section 3.5.

Sensitivity-Driven Approach

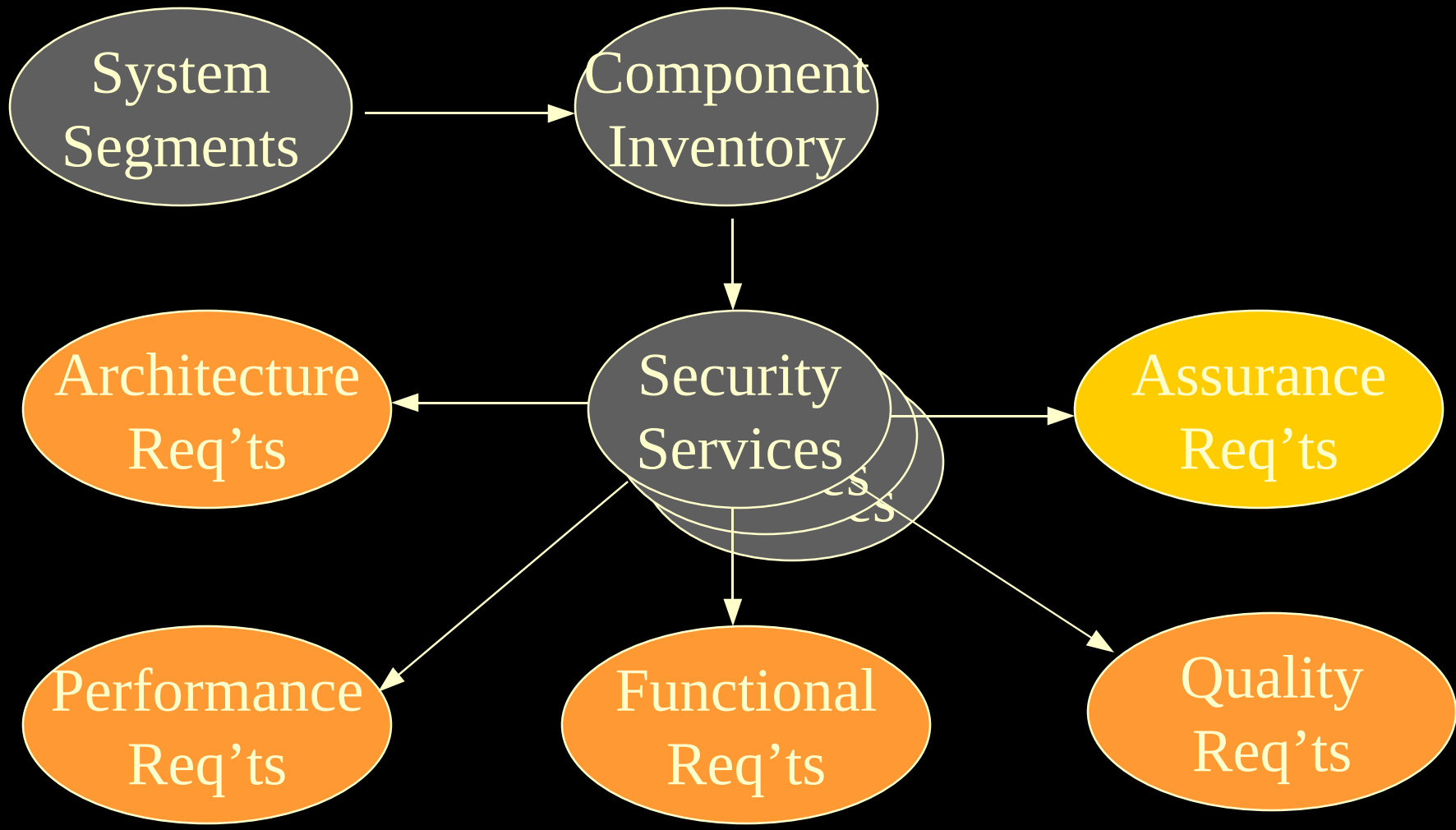
Advantages

- Validates priority and scope of security based on asset evaluation
- Provides comprehensive view of asset management
- Ensures coverage of requirements types for each info-asset

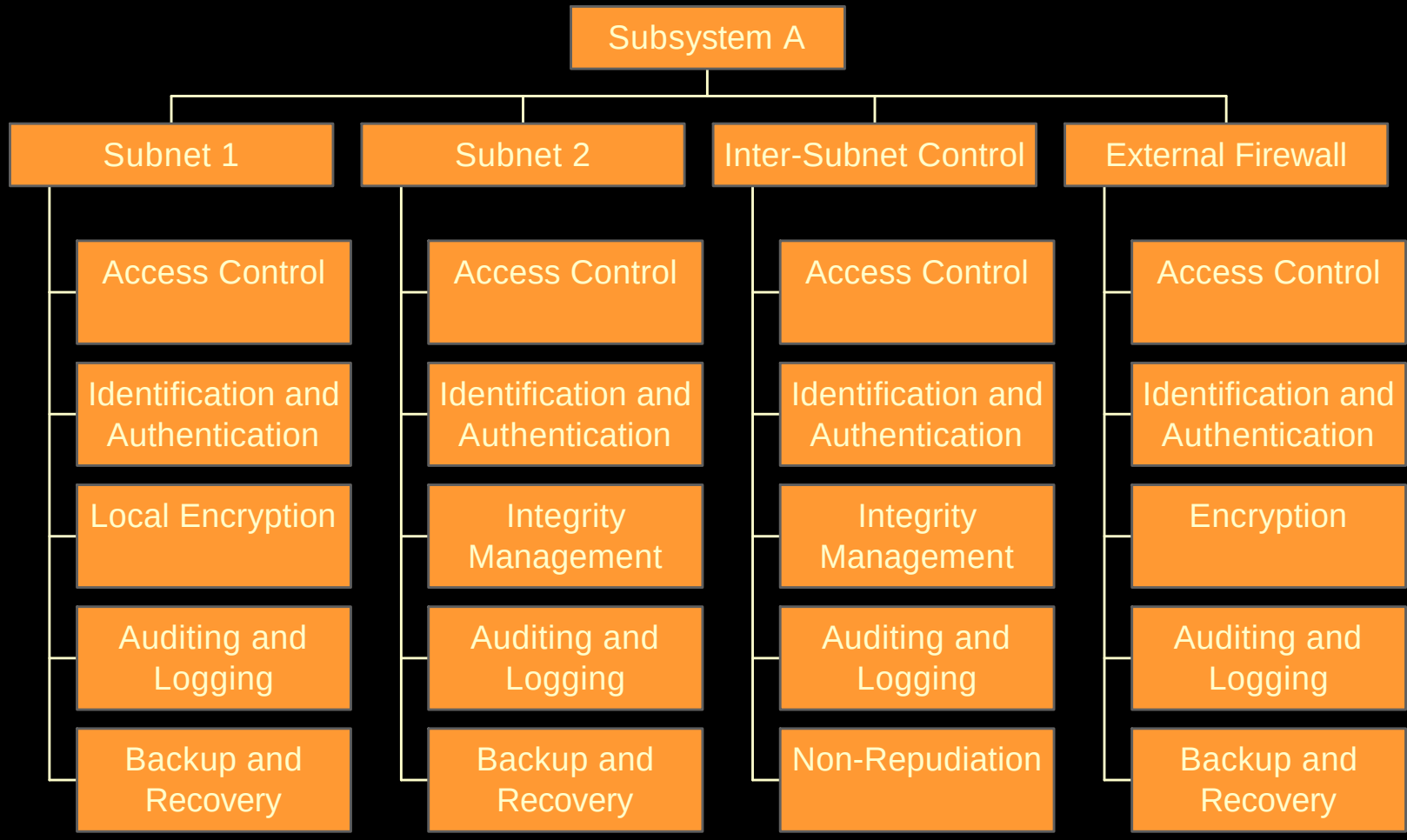
Disadvantages

- Structure can mask efficient application of security services
- Analysis is exhaustive and time-consuming
- Spec may be difficult to use and maintain due to redundancies

Architecture-Driven Analysis



Architecture-Driven Structure



Suggested Architectures: Chapman & Zwicky,Building Internet Firewalls. O'Reilly & Assoc.., Inc., Sebastopol CA, 1995. Chapter 4.

Architecture-Driven Sample

A.0 Subsystem A
- **A.1 Subnet 1**
- **A.2 Subnet 2**
- **A.3 Inter-Subnet Cont**
- **A.4 External Firewall**
- **A.5 Common Security**

B.0 Subsystem B

C.0 Subsy

A.1.1 Architecture Requirements
- A.1.1.1 Port Conventions
- A.1.1.2 . . .

A.1.2 Functional Requirements
- A.1.2.1 Access Controls
- A.1.2.2 Identification and Authentication
- A.1.2.3 Integrity Management
- A.1.2.4 Auditing and Logging
- A.1.2.5 Backup and Recovery

A.1.3 Performance Requirements

A.1.4 Quality Requirements

A.5.1 Architecture Requirements
- A.5.1.1 Network Protocols

A.5.2 Performance Requirements

A.5.3 Quality Requirements
- A.5.3.1 Portability

Architecture-Driven Approach

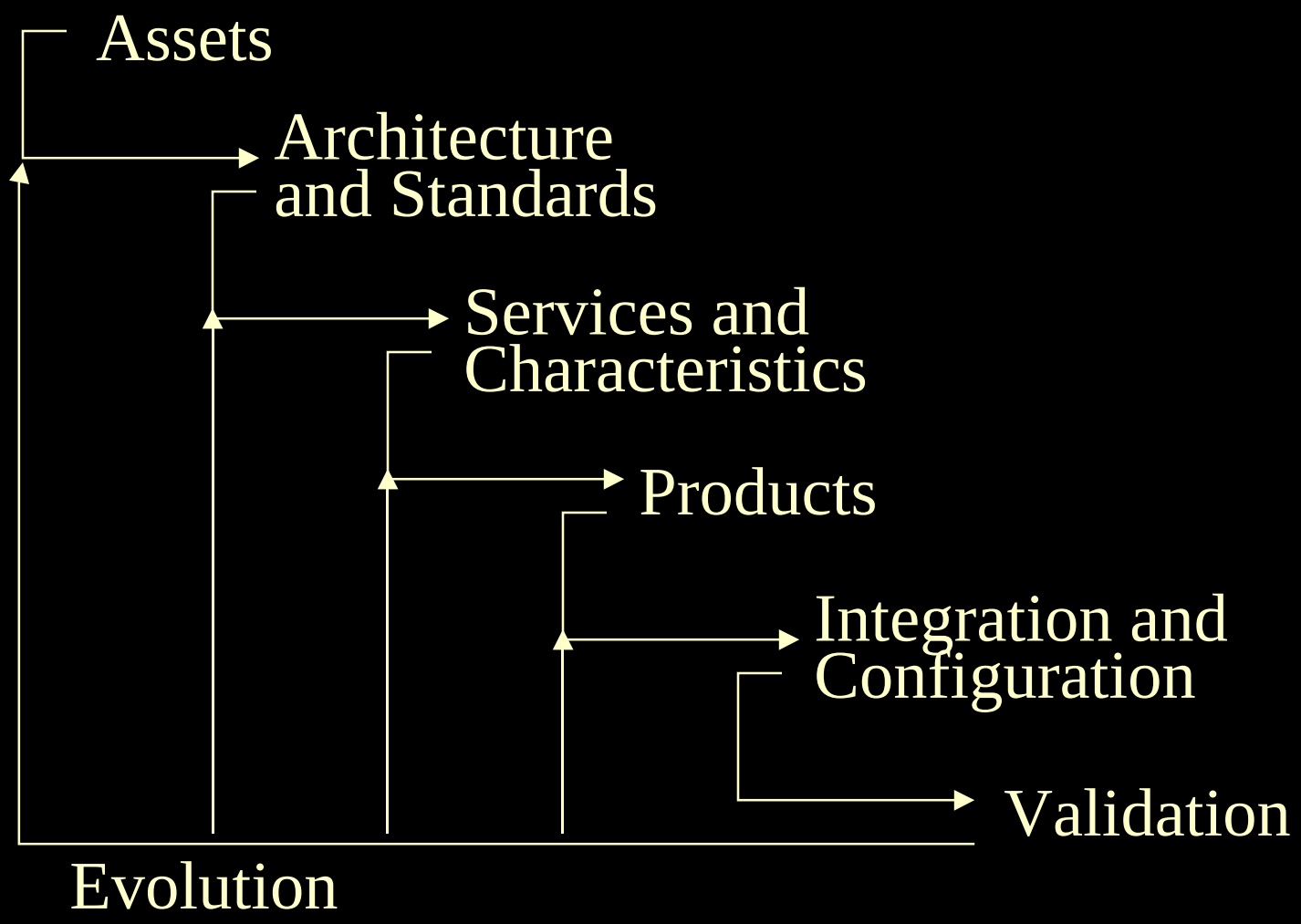
Advantages

- Improves spec usability for experienced operators and maintainers
- Ensures coverage of requirements types for each security domain
- Improves modularity for service upgrades

Disadvantages

- Structure can mask efficient application of security services and products
- Reduces coordination with other functional requirements
- Spec may be difficult to use and maintain due to redundancies

Stages of Security Definition



Evolutionary Spec Considerations

- Cluster standards or point to an appendix.
- Include placeholders for services, extensions and adaptations.
- Leave sections for unanticipated product features.
- Don’t delete; redline and reserve.
- Tag volatile requirements.

Part V

What's a Good Requirement?

Syntax and Semantics

Requirements statements are subject to many more misinterpretations than conversation!

- ⊗ Context is absent
- ⊗ History is lost
- ⊗ Audience is diverse
- ⊗ Information is dense
- ⊗ Concepts are complex

As I Was Saying...

The best requirements are*:

- Correct
- Unambiguous
- Complete
- Verifiable
- Consistent
- Traceable
- Maintainable

* *Source: Davis, Software Requirements: Analysis and Specification. Prentice-Hall, Englewood Cliffs NJ, 1990.*

Basic Syntax Goals

a. Use one subject, one verb and one object per statement.
b. If multiple objects are necessary, enumerate.
c. Employ active verbs.
d. Limit the use of clauses.
e. Avoid orphan phrases.
f. Check the grammar.

Syntax Nightmare #1

- ☹ A system administrators shall be authorized and provided with the capability to control security, except at system startup or recovery.

A. Two verbs.
B. Missing enumeration.
C. Passive verbs.
D. Conditional clause.
F. Article/noun singular/plural mismatch.

Syntax Fantasy #1

- ☺ Security auditing shall begin and end at the command of a system administrator.
- ☺ Security auditing shall begin within 1 second of an auditing startup command.
- ☺ Security auditing shall automatically begin at system startup and recovery.
- ☺ System administrators shall be authorized to perform the following functions:
    a. Audit log control,
    b. ...

Syntax Nightmare #2

- ☹ Accessing the system remotely from any host outside the system security perimeter, accounts will be used by identification and authentication in the same manner as local identification and authentication -- from inside the security perimeter.

C. Passive verb.
D. Complex clause.
E. Orphan phrase.
F. Dangling clause.

Syntax Fantasy #2

- ☺ Local -- Inside of the defined system security perimeter, including the security components of the perimeter (see ref. …).
- ☺ Remote -- Outside of the defined system security perimeter (see ref. …).

- ☺ Identification and authentication for remote access will use the same accounts as local identification and authentication.

Basic Semantic Goals

a. Define your terms!
b. Beware overloaded words.
c. Select accurate adjectives.
d. Check for internal consistency.
e. Include references to details where needed.
f. Avoid duplication among requirements.
g. Apply the “trainee” test.

Semantic Nightmare #1

- ☹ The Security Functions shall ensure that 600 audit records will be maintained when the following conditions occur: storage exhaustion or failure. *[FAU_STG.2.3]*

B. Overloaded words.
C. Inaccurate or missing adjectives.
G. Fails "trainee test".

Semantic Fantasy #1

- ☺ The 600 most recent audit records will be retained when any of the following conditions occur:
    a. Exhaustion of audit log storage,
    b. Exhaustion of all local storage,
    c. Auditing failure,
    d. Failure of the audit host.

Semantic Nightmare #2

- ☹ Remote users shall be limited to electronic mail and file transfers.

. . .

- ☹ The firewall shall filter out all services from external sources except cc:Mail and ftp.

A. Undefined terms.
B. Overloaded words.
D. Internal inconsistency.
E. Useful reference missing.
F. Duplication among requirements.

Semantic Fantasy #2

- ☺ Services available to remote users shall be limited to the following:
  a. sendmail [smtp],
  b. file transfer protocol [ftp],
  c. Lotus notes [lotusnotes], and
  d. Lotus cc:Mail [ccmail].

- ☺ Remotely-available services shall use standard IP port numbers as specified in RFC 1700.

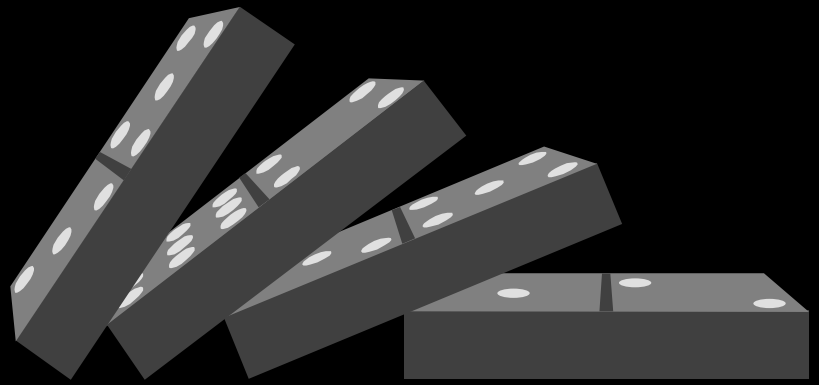
Before You Sign

- Perform an independent syntactic and semantic review.
- Perform a testability review.
- Listen to your reviewers!
- Trace back to policy and ensure congruence.
- Gain agreement on priority and volatility.
- Document and agree on a change process.
- Ensure requirements maintenance tools are in place.

Time to Stem the Tide

The most expensive errors
to fix are those that appear
at the requirements
definition phase.

Do your part!



Fight requirements illiteracy.

For additional clarification, discussion or insights:
paula.moore@earthlink.net