ESD.33 -- Systems Engineering

Session #4
Requirements Engineering

Pat Hale
As stated in ISO/IEC 15288:2008: The purpose of the Stakeholder Requirements Definition Process is to define the requirements for a system that can provide the services needed by users and other stakeholders in a defined environment. It identifies stakeholders, or stakeholder classes, involved with the system throughout its life cycle, and their needs, expectations, and desires. It analyzes and transforms these into a common set of stakeholder requirements that express the intended interaction the system will have with its operational environment and that are the reference against which each resulting operational service is validated.
What is the RE Process?

**Inputs**
- Source documents
- Stakeholders needs
- Project constraints

**Enablers**
- Organization/Enterprise policies, procedures, and standards
- Organization/Enterprise infrastructure
- Project infrastructure

**Activities**
- Elicit stakeholder requirements
- Define stakeholder requirements
- Analyze and maintain stakeholder requirements

**Outputs**
- Concept documents
- Stakeholder requirements
- Measures of effectiveness needs
- Measures of effectiveness data
- Validation criteria
- Initial RVTM
- Stakeholder requirements traceability

**Controls**
- Applicable laws and regulations
- Industry standards
- Agreements
- Project procedures and standards
- Project directives

Image by MIT OpenCourseWare.
Why ‘Engineer’ Requirements?

• Provide early assurance that all top-level requirements are fully satisfied in the product, with traceability to where they are satisfied.
• Prevent unintentional addition of cost (avoid ‘gold plating’).
• Establish clear responsibility for requirements implementation.
• Provide clear visibility across teams into requirements allocation and cross-functional interactions.
• Easily and quickly assess the impact of changes to requirements.
• Provide data for early and thorough validation and verification of requirements and design artifacts.
• Avoid unpleasant downstream surprises!
Requirements Management

Requirements Management is the process by which:

- System/Product needs are captured from the market, business strategy, customers and other sources.
- Requirements are synthesized from captured needs and the meanings and interpretations are validated with the original sources.
- Commitment to requirements is obtained from management and project personnel.
- A history of changes and change rationale is maintained as requirements evolve.
- Bi-directional traceability is maintained among requirements, project plans and work products.
- Inconsistencies are identified and resolved among requirements, project plans and work products.
What is a Requirement?

• Well-stated requirements exhibit the following attributes:
  – The requirement is **Necessary**
    • What would happen if you didn’t include this requirement?
  – The requirement is **Verifiable**
    • How will you know you have met the requirement?
  – The requirement is **Attainable**

• Requirements are concise and unambiguous
• Good requirements are solution-neutral
• Requirements are consistent (non-contradictory)
Requirements Semantics

• Requirements use ‘shall’
  – The system shall provide…
  – The system shall be capable of…
  – The system shall weigh…

• Statements of fact use ‘will’
  – Often part of a lead-in scenario prefacing one or more requirements

• Goals use ‘should’
  – Used for design goals where quantifiable requirements cannot be applied to a desired attribute
Summary: Attributes of Good Requirements

• What...not how
• Unambiguous, clear and complete
• Verifiable by test or independent analysis
• Necessary and appropriate for system hierarchy level
• Feasible
• Consistent (traceable) with requirements above and below in the system hierarchy
Requirements Baselines

Off-core user discussions and approvals  
(In-process validation)  
"Are the proposed baselines acceptable?"

Approved baseline

Baseline being considered

Baseline to be considered

Baseline to be verified

Planned verification

Baseline Verification  
"How to prove it's built right"

"Time Now"  
(Vertical line)  
with upward and downward iterations as required

"Time and Baseline Maturity"

Off-Core opportunity & risk management  
investigations and actions  
"How are the opportunities and risks of the proposed baselines being resolved?"

Core of the Vee  
Plans, specifications, and products are under progressive configuration management

Image by MIT OpenCourseWare.
### Product Commercialization Phases 1-6:

**Requirements Baselines in CDOV**

<table>
<thead>
<tr>
<th>Concept</th>
<th>Design</th>
<th>Optimize Sub-Systems</th>
<th>Optimize System</th>
<th>Verify Product Design</th>
<th>Verify Production Process</th>
</tr>
</thead>
</table>

1- Initial baseline  
2- Validated baseline (Contract)  
3- System functional baseline (PFS)  
4- Allocated baseline (PAS)  
5- Component baseline  
6- Verified product baseline
## System Life Cycle

### System/Program Milestones

<table>
<thead>
<tr>
<th>Milestone I</th>
<th>Milestone II</th>
<th>Milestone III</th>
<th>Milestone IV</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Functional Baseline</strong></td>
<td><strong>Allocated Baseline</strong></td>
<td><strong>Product Baseline</strong></td>
<td><strong>Updated Product Baseline</strong></td>
</tr>
</tbody>
</table>

- **System Management Plan**
- **System Specification** (TYPE A)
- System Engineering Management Plan (SEMP)
- Conceptual Design Review (System Requirements Review)
- System Design Review
- Equipment/Software Design Reviews
- Critical Design Review

### System Engineering Requirements

- **System Level**
  - Requirements Analysis
  - Functional Analysis
  - Requirements Allocation
  - Trade-Off Studies
  - Synthesis
  - Evaluation
  - Type A Specification
  - Design Review’s

- **Subsystem Level**
  - Refined Functional Analysis
  - Refined Requirements Allocation
  - Detailed Trade-Off Studies
  - Synthesis (CI)
  - Evaluation (Engineering Models)
  - Type A Specification
  - Design Review’s

- **Component Level**
  - Detailed Design
  - Detailed Synthesis
  - Evaluation (Prototype Models)
  - Types B to E Specifications
  - Design Review’s

### Modifications for Improvement

- **System/Component**
  - Proposed Design Modification(s)
  - Synthesis of Modification
  - Prototype Modifications
  - Test and Evaluation (Production Model)
  - Incorporation of Modification(s)
  - Configuration Item Reviews
  - System Evaluation (Field Assessment)

- **Subsystem**
  - Proposed Design Modification(s)
  - Synthesis of Modification
  - Prototype Modifications
  - Test and Evaluation (Operational Model)
  - Incorporation of Modification(s)

Adapted from Systems Engineering lecture slides at University of Witwatersrand, Johannesburg, South Africa (Dr. R. Siriram)
Requirements Hierarchy & Traceability

VOC
- Requirements Analysis & Concept Trades
- System Functional Design
- PFS Allocation
- PAS Subsystem & Component Design
  - Level 4
  - :
  - :

Source Needs
- incorporated by
- documents

Originating Requirement
- decomposed from
- satisfied by

Function
- incorporated

Subsystem
- integrates to
- built from

Assembly
- built from

Component
- integrates to

Validation Requirement
- generates
- traces to

Critical Issue
- traces to

Decision
- traces to

Test Method & Criteria
- traces to

Detailed Design
- built from

Analysis Cycle

Cycle
The requirement is **Verifiable**

**Existing Documents**

- VOC/Business Plan
- System HOQ
- System Requirements

**DOORS (Requirements)**

- System Requirements Spec
- Allocated System Req’ts

**DOORS (Test)**

- System Master Test Plan
- System Integration Test Cases

**Subsystem Allocation**

- ss1
- ss2
- ss(n)

**Subsystem Design Req’ts**

- Design Artifacts

- Test Procedures

**Primary Requirements Flow**

- Report Generation
- Test Linkages
Requirements Change Authority

**VOC**
Baseline
Requirements Analysis & Concept Trades

**Contract**
System Functional Design

**PFS**
Allocation

**PAS**
Subsystem & Component Design

**Level 4**
Detailed Component/Part Specifications

**Pre-PAS:**
- Changes/new requirements proposed by any development team member
- Review by Systems team and other Stakeholders (e.g. PMT members) as required
- Approval by Systems team/Chief Systems Engineer

**Post-PAS:**
- Changes/new requirements proposed by any development team member
- Review by Subsystem team with appropriate representation from other affected teams
- Decision by Subsystem team leader with primary design responsibility
- Weekly review of changes and change proposals by Systems team
# Implementation Plan

<table>
<thead>
<tr>
<th>Task Description</th>
<th>Who?₁</th>
<th>Target Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Propose Draft RM process</td>
<td>JW, Cons.</td>
<td>10/31/2003</td>
</tr>
<tr>
<td>Select DOORS versions, obtain training</td>
<td>LD</td>
<td>11/3/2003</td>
</tr>
<tr>
<td>Develop requirements schema for RM &amp; CPM (link, no common schema)</td>
<td>JW, SS, Cons.</td>
<td>11/12/2003</td>
</tr>
<tr>
<td>Review/Import existing requirements, refine to meet RM criteria</td>
<td>JW, SS, KH, Cons.</td>
<td>12/28/2003</td>
</tr>
<tr>
<td>Certify PFS baseline in DOORS</td>
<td>JW, DD, Mkt, Test</td>
<td>1/10/2004</td>
</tr>
<tr>
<td>Full process implementation</td>
<td>NGD Team</td>
<td>1Q04</td>
</tr>
</tbody>
</table>
System-level Implementation
(Example from a backup tape drive project)
Documents to DOORS Modules

[Diagram showing the process of converting documents to DOORS modules, including steps like Rich Text (.rtf), Word, Edit, Prepare for Export, Framemaker, Create Link Module, Baseline PAS, Assign Attributes, and Export & Create Doors Module.]

Image by MIT OpenCourseWare.
Formatting PAS in Word for Export into Doors

Keep heading styles in the following form: Heading 1, Heading 2, Heading 3, ..., etc.

Notes:
- Framemaker documents use Sec-#, Sec-#.#, Sec-#.#.#. Export Framemaker documents to Word and convert them to Heading 1, Heading 2, etc. before exporting into Doors.
- Keep all other styles consistent throughout (e.g. figure heading, table heading, etc.)
The System Engineering Process

**Phased Process Outputs**
- Decision support data
- System architecture
- Specification
- Baselines

**System Analysis & Control (Balance)**
- Modelling, Analysis Simulation
- Design for (x)
- Trade studies/Trade offs
- Risk management
- Configuration & Data management
- Interface management
- Work breakdown
- Critical parameter management (CPM)
- Technical reviews
- Lessons learned

**Process Inputs**
- Business objectives
- Market/competitive requirements
- Customer needs/objectives
- Technology base
- Environments
- Constraints
- Specifications, codes, standards
- Program decision requirements
- Measures of effectiveness

**Requirements Management Sub-process Region**

**Requirements Loop**
- **Requirements Analysis**
  - Analyze markets and environments
  - Identify functional/requirements
  - Define/refine performance requirements
  - Identify constraint requirements

- **Functional Analysis/ Allocation**
  - Decompose functional requirements
  - Allocate functionality and performance to lower levels
  - Define/refine functional interfaces and architecture

**Design Loop**
- **Synthesis**
  - Transform architecture (Functional to physical)
  - Define configuration items & elements
  - Define/refine physical interfaces
  - Select preferred product/process solutions

**Verification Loop**

**Illustration of a Single Iteration**

Image by MIT OpenCourseWare.
Another View of the ‘V’ Model

Requirements

- VOC
- System Requirements
- Architectural Design
- Detailed Design
- Assembly Integration & Verification

Verified By

Test Phase

- Acceptance Test
- System Test
- Integration Test
- Subsystem Test
- Unit Tests

Time

Embodiment/Code

Function Requirements

Another View of the ‘V’ Model
NGD Requirements Structure & Links

Design

Product Contract

PFS

PAS

Level 4

NGD Test

Integrated Test Plan

Detailed Test Plans

Test Suites

Test Cases

Library Test

Library Test Plans

Test Suites

3rd Party Certs

Host Applications

Network Infrastructure

Requirements Links

Test Links

Req’ts-Test Links
## Module Schema

### Test Suite Module

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Req'd?</th>
<th>Inherit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object_ID</td>
<td>Index</td>
<td>Auto</td>
<td>N/A</td>
<td>Prefix w/number assigned by DOORS</td>
</tr>
<tr>
<td>Test_Name</td>
<td>Text</td>
<td>Yes</td>
<td>No</td>
<td>Shorthand nomenclature for test suite</td>
</tr>
<tr>
<td>Req_Source</td>
<td>Enum/1</td>
<td>Opt.</td>
<td>Yes?</td>
<td>Values= PC, PFS, PAS(RO...)</td>
</tr>
<tr>
<td>Document</td>
<td>Text</td>
<td>Yes</td>
<td>No</td>
<td>External test reference document</td>
</tr>
<tr>
<td>Test_Purpose</td>
<td>Enum/M</td>
<td>Yes</td>
<td>Yes</td>
<td>Values= Reliability, Performance, Standards, Regulatory, Interface, Functional</td>
</tr>
<tr>
<td>Test_Environ</td>
<td>Text</td>
<td>Yes</td>
<td>No</td>
<td>Description of conditions and equipment, software environment necessary for test suite conduct</td>
</tr>
</tbody>
</table>