LSST Verification & Validation Process & MBSE Methodology

Brian Selvy
Kathryn Wesson, George Angeli
Project Systems Engineering

Telescope MBSE SIG
November 2, 2016
Agenda

- LSST’s Verification Process
  - Verification Planning
  - Compliance Assessments
  - Developing Verification Events and logical sequences
- Verification to Commissioning Process Implementation in an MBSE Environment
- V&V integrated into Assembly, Integration, and Verification (AIV) in an MBSE Environment
- End-to-End Requirements-Verification Traceability in an MBSE Environment
- Update on OMG v2 Verification metamodel
Verification & Validation on LSST

- LSE-160 Verification and Validation Process is the governing document for V&V on LSST
  - Establishes a consistent, project-wide process for the development of V&V plans, compliance assessments, V&V reporting, and deliverables
- Defines steps in the verification process
- Defines requirements for developing verification plans for each project-controlled requirement
LSE-160 Applicability

- Applies to all Project-Controlled requirements:
  - Specifications
  - Requirements Documents
  - Interface Control Documents (ICDs)
- Each “shall” statement in each of these documents must be formally verified
LSST Verification & Validation Process

1. Identify all requirements
2. Ensure requirements are verifiable
3. Define verification method, write verification requirement, define success criteria
4. Identify task interdependency
5. Schedule verification events
6. Conduct verification events
7. Prepare verification reports
8. Prepare final verification matrices
9. Final acceptance review

VM-P
Verification Planning

VM-C
Compliance Assessments
For each requirement ("shall statement") a Verification Plan will be created that includes the following:

- **Verification Owner** – the subsystem team that is responsible for verification

- **Responsible Technical Authority (RTA)** – the point-of-contact assigned responsibility for the verification of the requirement from the responsible subsystem. The RTA, along with the responsible QA individual, has responsibility for overseeing all associated verification events.

- **Verification Method(s)** – Test, Analysis, Inspection, Demonstration

- **Verification Level** – Same Level, Higher Level, Lower Level

- **Verification Requirement** – A statement that defines precisely what will be done to verify the requirement. If there is any vagueness in the requirement, the Verification Requirement should clearly address the noted issues and define what precisely will be verified and any limitations. The statement should define what will be done, where it will be done, what special test equipment (SPE) is needed, and what project hardware/software is needed.

- **Success Criteria** - A statement that defines the explicit pass/fail criteria. This statement should be clear enough that an independent third party observer should be able to determine if the verification event was successful or not.
Compliance Assessment Requirements

- Compliance is defined as the ability of the current (any point in time) “as-designed” system to meet its associated requirements.
- The difference between compliance and verification is that verification is conducted on the final designed and built system, whereas compliance can be done at any earlier time and is an early step in the overall verification process.
- Compliance Assessments are required at each major subsystem and component design review.
- Required documentation:
  - Compliance Method(s) – Analysis, Test, Demonstration, Inspection
  - Verification Requirement
  - Success Criteria
  - Compliance Status (Y/N)
  - References to any additional documentation that further justifies the assessment, if available.
A final Verification Record is compiled after all requirements within a specification / ICD have been verified.

A Verification Matrix for Final Verification (VM-V) serves as the final record and summary of the verification process.

For each requirement, summary information from the Verification Plan is included along with:

- **Responsible Technical Authority (RTA)**
- **Verification Successful (Y/N)**
- **Verification Result Summary** – a concise summary narrative explaining why the verification activities were successful or not.
- **Verification Report** – A reference to the Verification Report that contains the details of the results of the verification activities.

For each requirement, an RTA will be identified. These individuals are responsible for vouching that the requirement has been verified and generating initial responses to Non-Conformances.
Verification Process Steps 4 and 5

- After Verification Plans are generated, PSE uses this information, along with additional input from the subsystem teams, to develop a comprehensive system-level Verification model:

- Identify Task Interdependency (Step 4)
  - Some Verification Activities can be naturally grouped and conducted at the same time
  - These Verification Activities are then grouped into a single Verification Event.
    - Can result in cost and schedule savings from eliminating redundant or nearly redundant V&V activities

- Schedule Verification Events (Step 5)
  - Events are scheduled, identifying predecessor/successor relationships and other schedule constraints
End-to-End Verification Implementation Process

- Different Tools Utilized for Various Strengths
  - Enterprise Architect
    - Manage Requirements
    - Traceability
    - Self Consistent Plan
    - Documentation from Model
  - SysML
  - PMCS (Primavera)
    - Integrated Master Schedule
    - EVMS
  - JIRA
    - Agile / Ability to Adapt quickly
    - Connectivity back to EA for Verification Closeout
End-to-End Verification Implementation Process
End-to-End Verification Implementation Process
End-to-End Verification Implementation Process

DRAFT IN WORK

Telescope MBSE SIG • Pasadena, CA • 11/02/2016
End-to-End Verification Implementation Process

DRAFT IN WORK

Telescope MBSE SIG • Pasadena, CA • 11/02/2016
End-to-End Verification Implementation Process

**Diagram Description:**
- **Activity:** Commissioning Image Quality Test Activity
- **Flow:**
  - Reverify Created Image quality
  - Take measurements using whole focal plane at WFS
  - WFS reconstruction analysis
  - Aggregate telescope element errors
  - Field Dependent Telescope Image Quality Test
  - Aggregate camera element errors
  - Take Field Dependent Camera image Quality Test

**Legend:**
- Action
- Data Store
- Activity
- Decision
- Control Flow
- Object/Data Flow
End-to-End Verification Implementation Process
SysML does not have a predefined element capable of capturing LSST’s Verification Planning information.

- SysML is extensible, allowing for the definition of additional stereotypes.
- LSST created a `VerificationPlan` stereotype as an extension of the SysML1.3::requirement metaclass.
Creation of Verification Plans & Test Cases in the Model
Sequencing Test Cases (Verification Events)

- Test Cases (Verification Events) are sequenced on Activity Diagrams to show:
  - Predecessor/successor relationships
  - Events that are conducted in parallel/series
  - Outside constraints that must be met before a Verification Event can be executed

• Results can be used to validate or update the project’s schedule for the Commissioning period.
Refinement of Individual Test Cases (Verification Events)

- As plans mature, individual Verification Events can be further detailed via association with its own detailed behavior diagram
- Serves as refined and more detailed input to the commissioning planning effort
  - Can be used directly as inputs to writing detailed test & analysis procedures
Mapping Individual Test Case Steps to LSST’s PMCS

- Refined Test Case Actions mapped to associated Project Management Control System (PMCS) activity steps.
- Ensures Verification Activities are included in EVMS
Verification is one critical aspect of the broader manufacturing, assembly, integration, and verification set of activities.

- Project Systems Engineering needs to understand the early integration and verification activities being conducted by the subsystems that impact system level requirements, interfaces, assemblies, and verification activities.

- A general pattern has been defined that PSE will use to document these activities in Enterprise Architect using the SysML language (next slide).
AIV Pattern

Input Objects

- Hardware Component(s)
- Software Component(s)

Output Objects

- Assembly
- Manufacturing / Value Added Process

Transformation Processes

- Verification
  - To Verification & Acceptance
  - From Verification & Acceptance
  - Verify Form, Fit, and Function

Integration Processes

- Integration Activity
  - To Verification & Acceptance
  - Verify Interface Requirements
  - From Verification & Acceptance

From Previous Step

To Next Manufacturing Step

In Process QA & Verification (as needed)

Verify Form, Fit, and Function

Verification

Input Objects

- Hardware Component(s)
- Software Component(s)

Output Objects

- Assembly
- Manufacturing / Value Added Process

Transformation Processes

- Verification
  - To Verification & Acceptance
  - From Verification & Acceptance
  - Verify Form, Fit, and Function

Integration Processes

- Integration Activity
  - To Verification & Acceptance
  - Verify Interface Requirements
  - From Verification & Acceptance

From Previous Step

To Next Manufacturing Step

In Process QA & Verification (as needed)

Verify Form, Fit, and Function

Verification

Input Objects

- Hardware Component(s)
- Software Component(s)

Output Objects

- Assembly
- Manufacturing / Value Added Process

Transformation Processes

- Verification
  - To Verification & Acceptance
  - From Verification & Acceptance
  - Verify Form, Fit, and Function

Integration Processes

- Integration Activity
  - To Verification & Acceptance
  - Verify Interface Requirements
  - From Verification & Acceptance

From Previous Step

To Next Manufacturing Step

In Process QA & Verification (as needed)

Verify Form, Fit, and Function

Verification
Ongoing work – Initial AIV Models

- Partial Camera AIV
Ongoing work – Initial AIV Models

- T&S Mirror Systems Integration and Test Phase 1
Ongoing work – Initial AIV Models

− TMA Integration and Test
A Requirements to Verification Plan Example
Requirements Derivation
A Requirements to Verification Plan Example
The Object Management Group (OMG) currently has a team working on requirements for a major revision to SysML (notionally referred to as SysML v2)

- Brian Selvy (LSST) and David Haines (Boeing) are developing Verification Concepts

- Feedback welcome
Backup Slides
Requirements Engineering
- All project-controlled requirements are captured as elements in the EA SysML model

- Each specification from the LSST Specification Tree is modeled as a version-controlled package

- Requirements are modeled as Requirement elements under the applicable package.
Packages are used to manage our requirements for version control and document generation.

All of the LSST’s system level requirements documents are generated from the model.

8 System level documents contain ~1000 requirements
SysML Implementation – Definition of Rqmts Hierarchy

- Requirements Diagrams used to show:
  - Model hierarchy (using *Containment* relationship)
  - Requirements traceability via decomposition and allocation (using *Derived* relationship)
Extending the SysML Requirement Syntax

Requirement title
Tool extension enforces unique ID tag value

Requirement text
Clarifying discussion text (if needed)

SysML constraint blocks are used for quantitative attributes refines the requirement

System Image Quality

- **tags**
  - LSSTRequirements = OSS-REQ-0228

- **notes**
  - **Specification:** The delivered image quality of isolated bright unresolved point sources in images from a single visit shall have the properties specified in the table **ImageQuality**.

  - **Discussion:** The design point specified here deviates from the SRD design specification due to the conflict between image quality and charge spreading in thick detectors, needed to achieve the desired z-band and y-band sensitivities. The adopted base system image quality of 0.4 arcsec FWHM is within the allowed value set by the SRD minimum specification.

- **constraintBlock**
  - **ImageQuality**
    - Syslm_0 : ArcsecFWHM = 0.40
    - Syslm_45 : ArcsecFWHM = 0.49
    - Syslm_60 : ArcsecFWHM = 0.60
    - SX : float = 1.1
    - SF1 : Percent = 10
    - PSFSample : Pixels = 3
    - ImFunc = 0.6
    - SR1 : Arcsec = 0.76
    - SR2 : Arcsec = 1.17
    - SR3 : Arcsec = 1.62
Nested requirements structure are used to further detail a parent requirement within the parent’s domain.
Requirements flow down and traceability example

SysML Relationships
- derive
- satisfy
- Trace
- Refine
- allocate
- generalize

Generalization relationship between constraint blocks allows attribute inheritance
Modeling tool provides means to analyze and manage flow down

“owns” and “needed by” provides downward traceability

“owned by” and “depends on” provides upward traceability

Model namespace also provides traceability
DocGen of Requirements

- The LSST Project generates traditional requirements specifications from the model.
  - Allows for dissemination beyond the core set of model users
## Subsystem Level Milestones

<table>
<thead>
<tr>
<th>Hardware Centric</th>
<th>Software Centric</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Review, Verification, and Acceptance Milestones to be identified for each Component:</strong></td>
<td><strong>Review, Verification, and Acceptance Milestones to be identified for each Component:</strong></td>
</tr>
<tr>
<td>Requirements Review</td>
<td>Release Objectives Review</td>
</tr>
<tr>
<td>Final Design Review</td>
<td>Verification Plan Review</td>
</tr>
<tr>
<td>Procurement Review</td>
<td>Unit Test</td>
</tr>
<tr>
<td>Manufacturing Readiness Review</td>
<td>Low Level Integration Test</td>
</tr>
<tr>
<td>Verification Plan Review</td>
<td>End to End Test</td>
</tr>
<tr>
<td>Start of Verification Activities (i.e. Tests)</td>
<td>Acceptance Test</td>
</tr>
<tr>
<td>Subsystem Pre-shipment Review (if applicable)</td>
<td>Acceptance Test Review</td>
</tr>
<tr>
<td>Subsystem Acceptance Review</td>
<td>.....</td>
</tr>
<tr>
<td>.....</td>
<td>.....</td>
</tr>
</tbody>
</table>
Verification vs. Validation

- **Verification:**
  - Ensures that the system, its elements, and its interfaces conform to their requirements.
  - “You built it right.”

- **Validation:**
  - Provides objective evidence that the services provided by a system when in use in an operational environment comply with the stakeholders’ needs.
  - “You built the right thing.”

![Vee Diagram](image-url)
Basis of Verification

- Statements of need, requirements, and constraints are written using one of three specific verbs that have a direct tie to verification:
  - **Will** – A statement of fact. Will statements document something that will occur through the course of normal design practice, project process, etc. These statements do not get formally verified.
  - **Should** – A goal. Should statements document a stretch goal. A should statement will be partnered with a shall statement. Should statements do not get formally verified.
  - **Shall** - A requirement that gets formally verified. Shall statements document critical requirements that must be verified through inspection, demonstration, analysis, or test during the verification phase of the project to ensure objectively that the as-built design meets the requirement.
- As noted by these definitions, only “shall” statements are formally verified.
**Inspection**: An examination of the item against applicable documentation to confirm compliance with requirements. Inspection is used to verify properties best determined by examination and observation (e.g., paint color, weight, etc.)

**Analysis**: Use of analytical data or simulations under defined conditions to show theoretical compliance. Analysis (including simulation) is used where verifying to realistic conditions cannot be achieved or is not cost-effective and when such means establish that the appropriate requirement, specification, or derived requirement is met by the proposed solution.

**Demonstration**: A qualitative exhibition of functional performance, usually accomplished with no or minimal instrumentation. Demonstration (a set of verification activities with system stimuli selected by the system developer) may be used to show that system or subsystem response to stimuli is suitable. Demonstration may also be appropriate when requirements or specifications are given in statistical terms (e.g., mean time to repair, average power consumption, etc.)

**Test**: An action by which the operability, supportability, or performance capability of an item is verified when subjected to controlled conditions that are real or simulated. These verifications often use special test equipment or instrumentation to obtain very accurate quantitative data for analysis. (Haskins, 127)