LSST Verification & Validation Process & MBSE Methodology

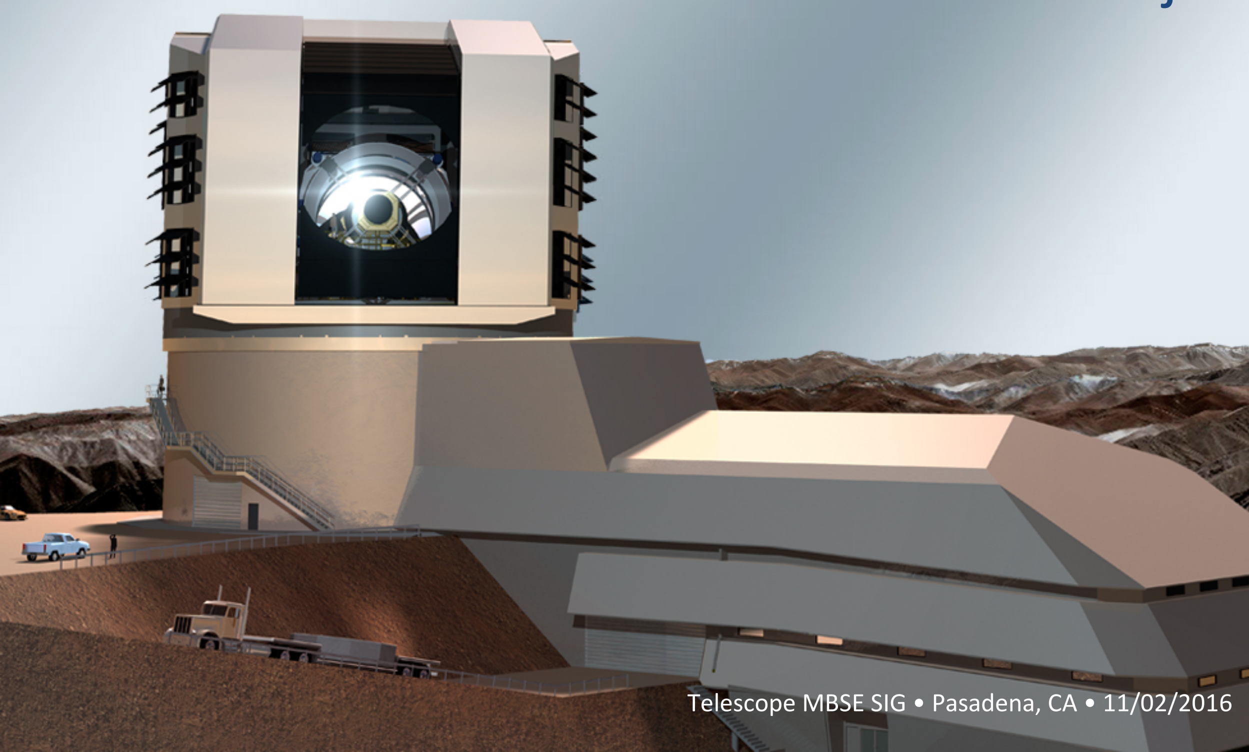
Brian Selvy

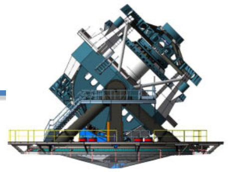
Kathryn Wesson, George Angeli

Project Systems Engineering

Telescope MBSE SIG

November 2, 2016



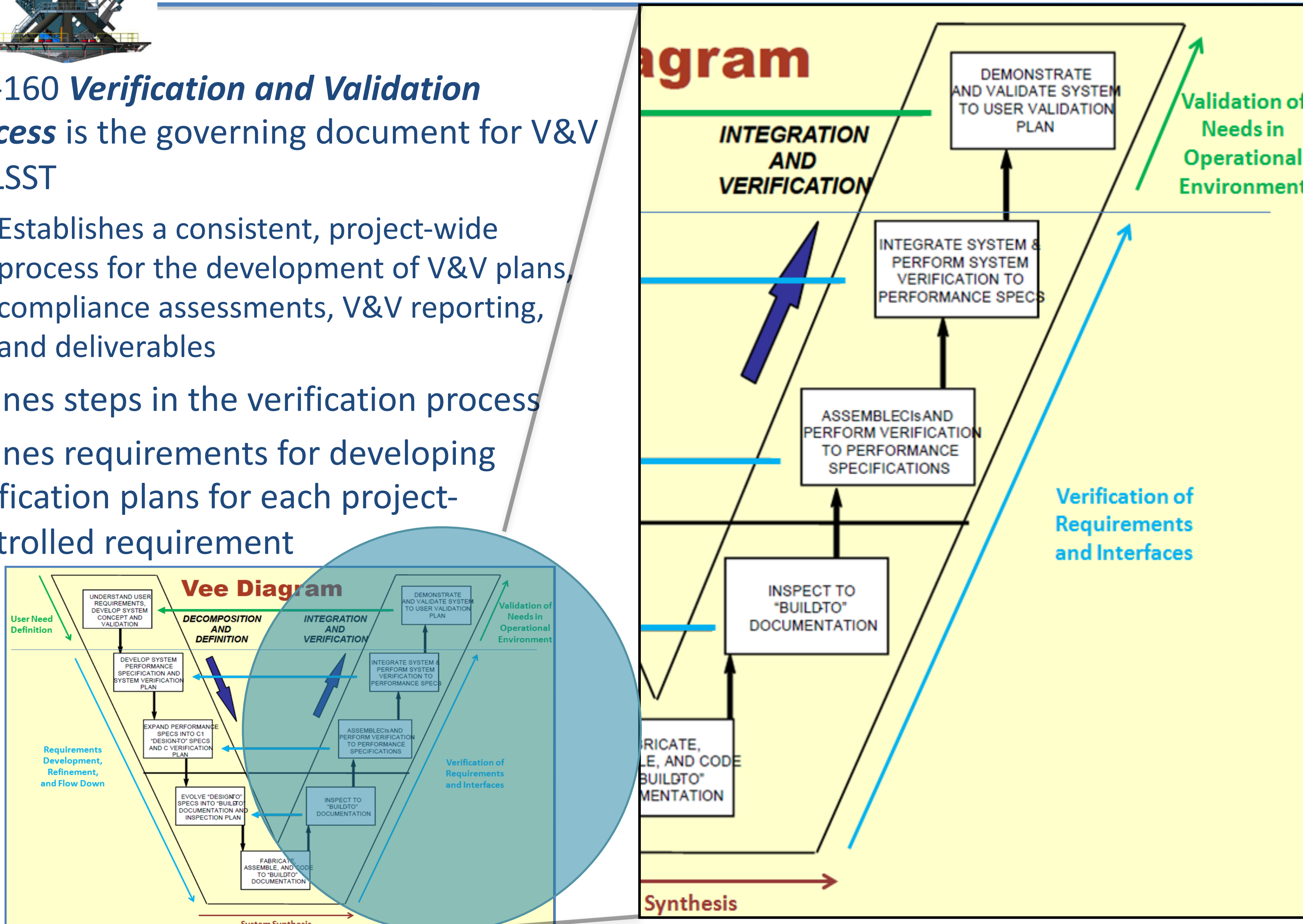


Agenda

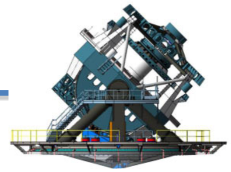
- LSST's Verification Process
  - Verification Planning
  - Compliance Assessments
  - Developing Verification Events and logical sequences
- Verification to Commissioning Process Implementation in an MBSE Environment
- V&V integrated into Assembly, Integration, and Verification (AIV) in an MBSE Environment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

Verification & Validation on LSST

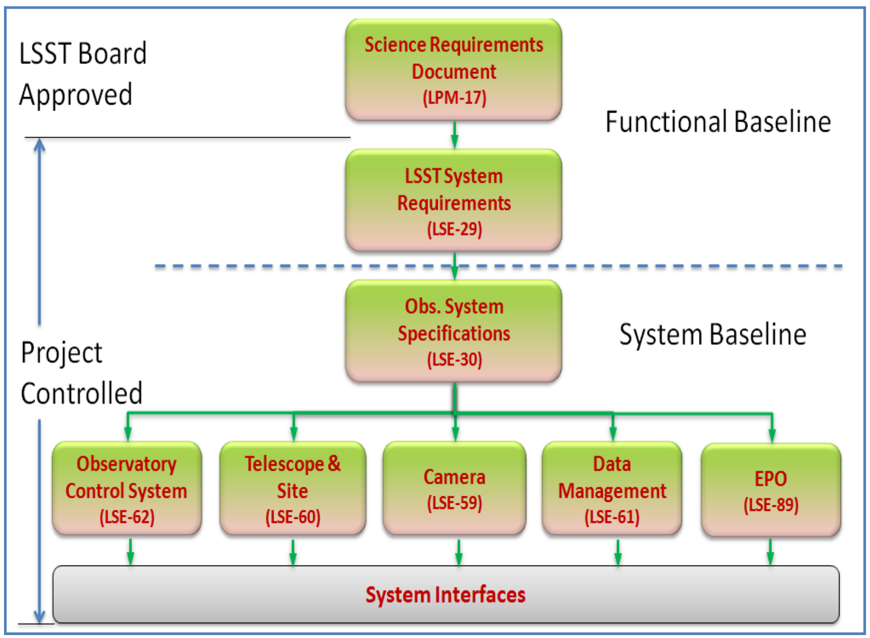
- LSE-160 ***Verification and Validation Process*** is the governing document for V&V on LSST
  - Establishes a consistent, project-wide process for the development of V&V plans, compliance assessments, V&V reporting, and deliverables
- Defines steps in the verification process
- Defines requirements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



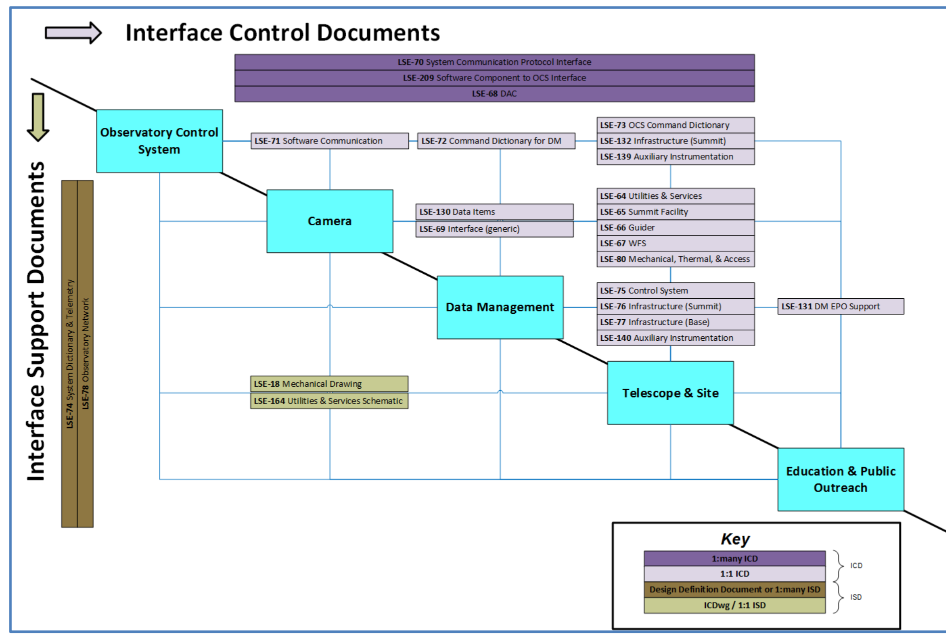
LSE-160 Applicability



- 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

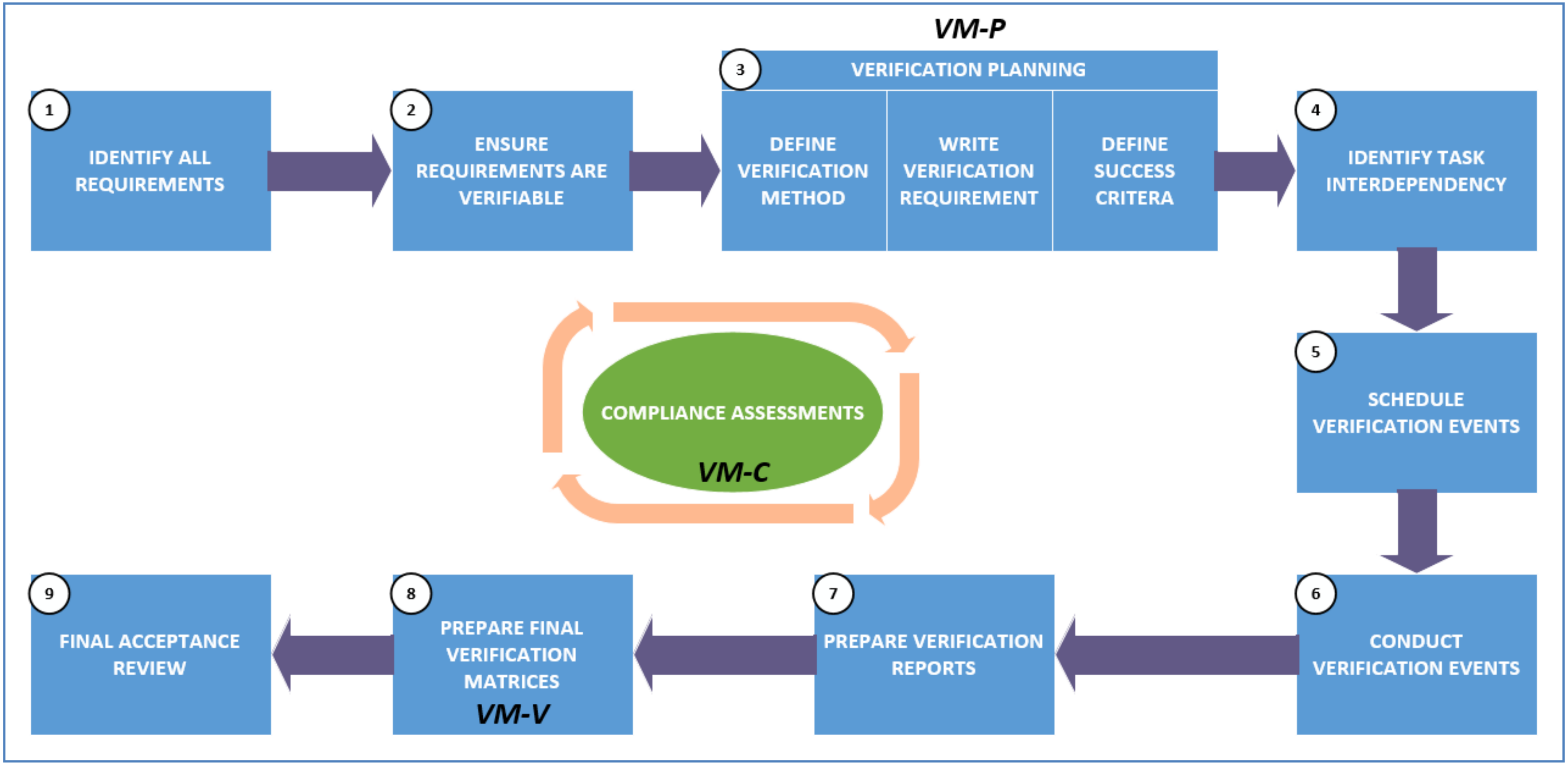


Project-Controlled Specifications

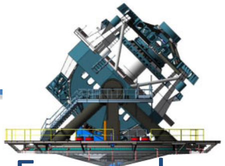


Project-Controlled ICDs

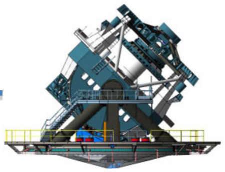
LSST Verification & Validation Process



Verification Planning Requirements



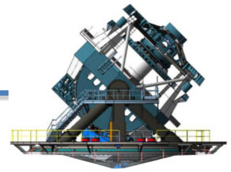
- For each requirement ("shall statement") a Verification Plan will be created that includes the following:
  - ***Verification Owner*** – the subsystem team that is responsible for verification
  - ***Responsible Technical Authority (RTA)*** – the point-of-contact assigned responsibility for the verification 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.

Final Verification Reporting Rqmts

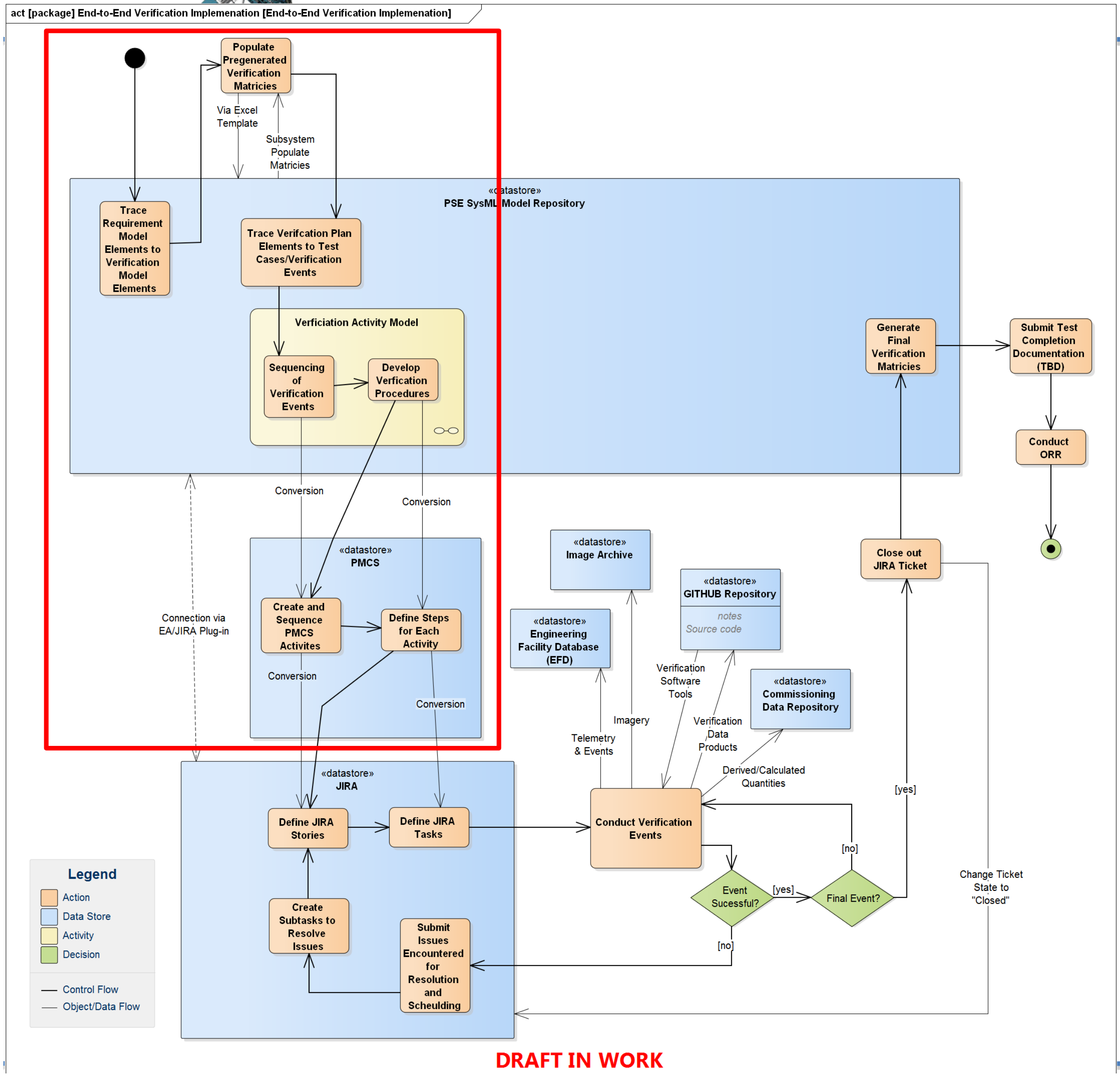


- A final Verification Record is compiled after all requirements within a specification / ICD have been verified.
- A Verification Matrix for Final Verification (VM-V) serves as the final record and summary of the verification process.
- For each requirement, summary information from the Verification 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

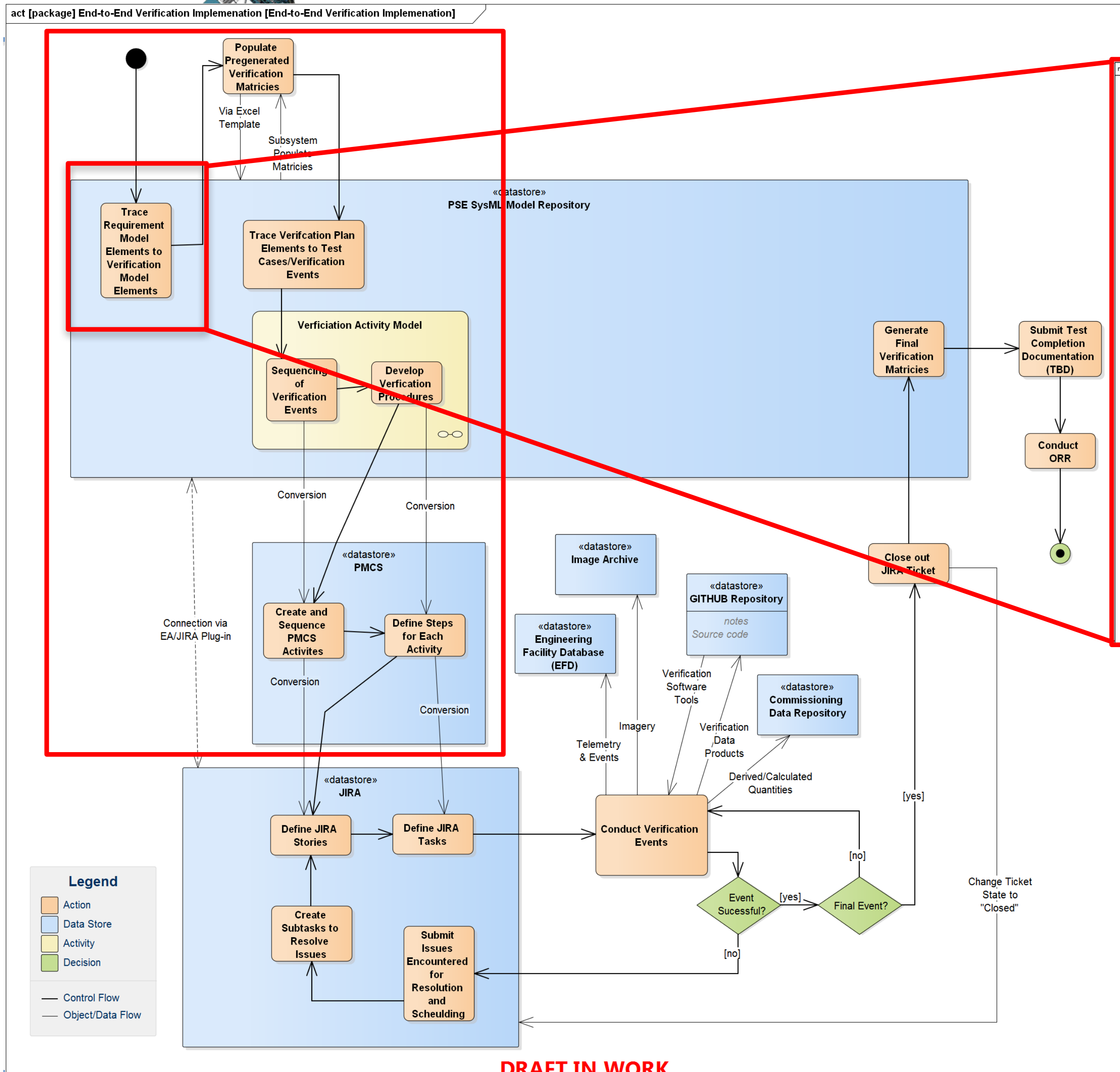
- After Verification Plans are generated, PSE uses this information, along with additional input from the subsystem teams, to develop a comprehensive system-level Verification model, :
- Identify Task Interdependency (Step 4)
  - Some ***Verification Activities*** can be naturally grouped and conducted 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

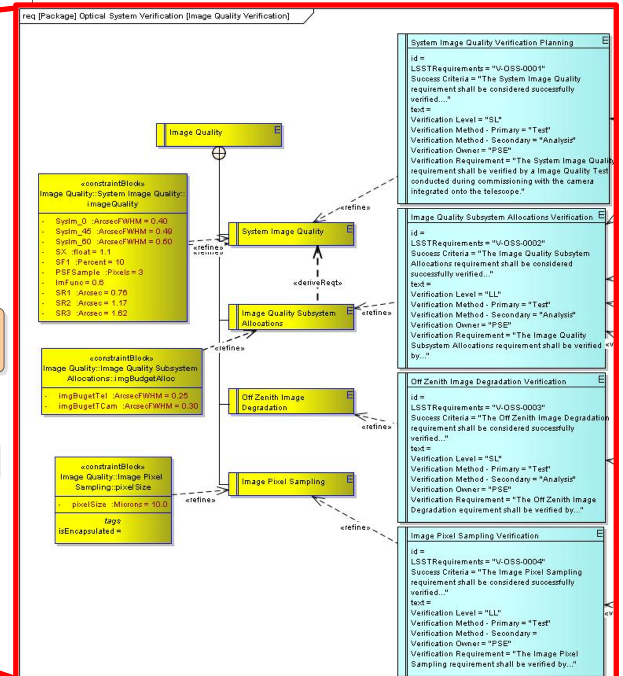
End-to-End Verification Implementation Process



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

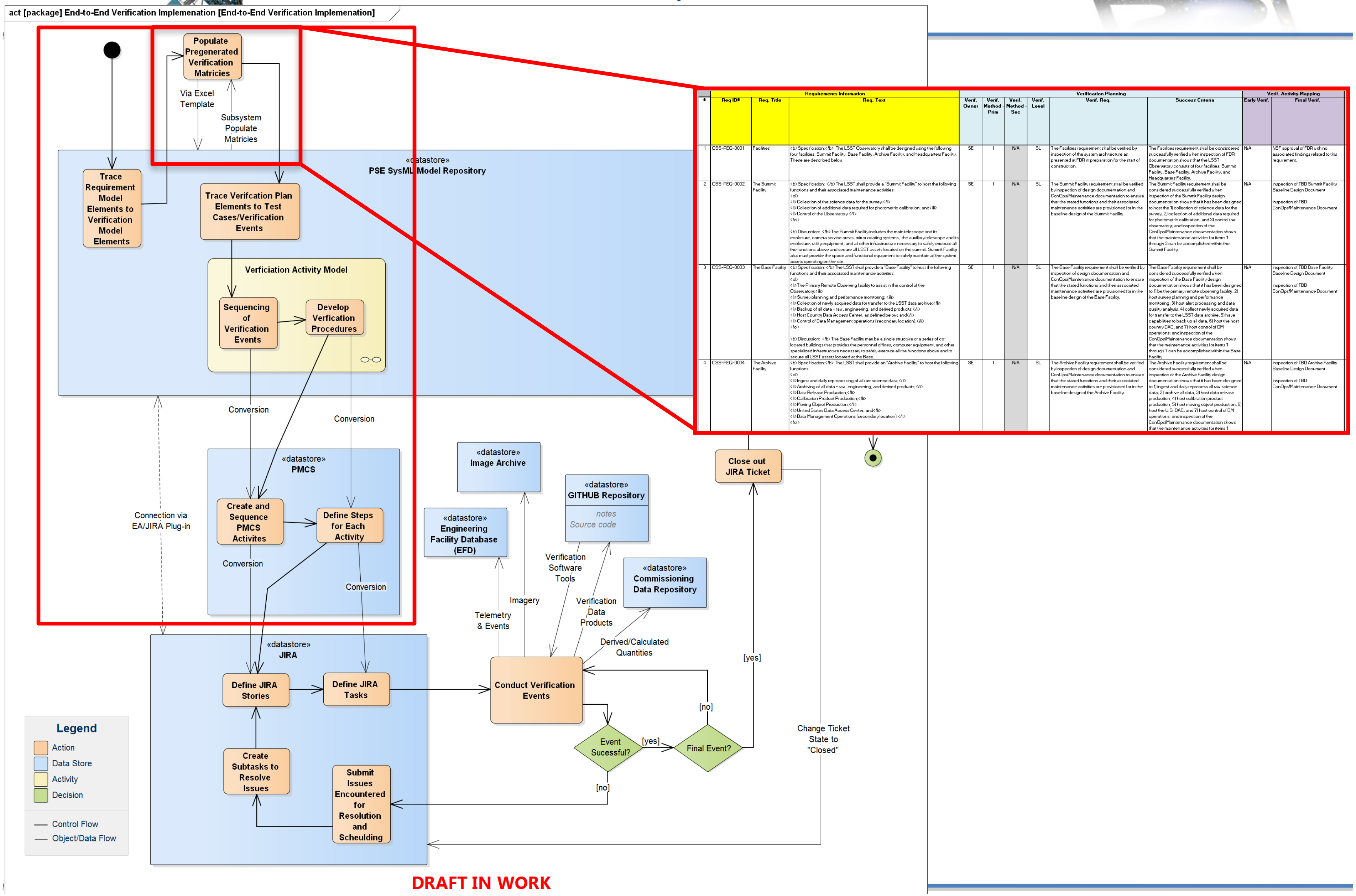
End-to-End Verification Implementation Process





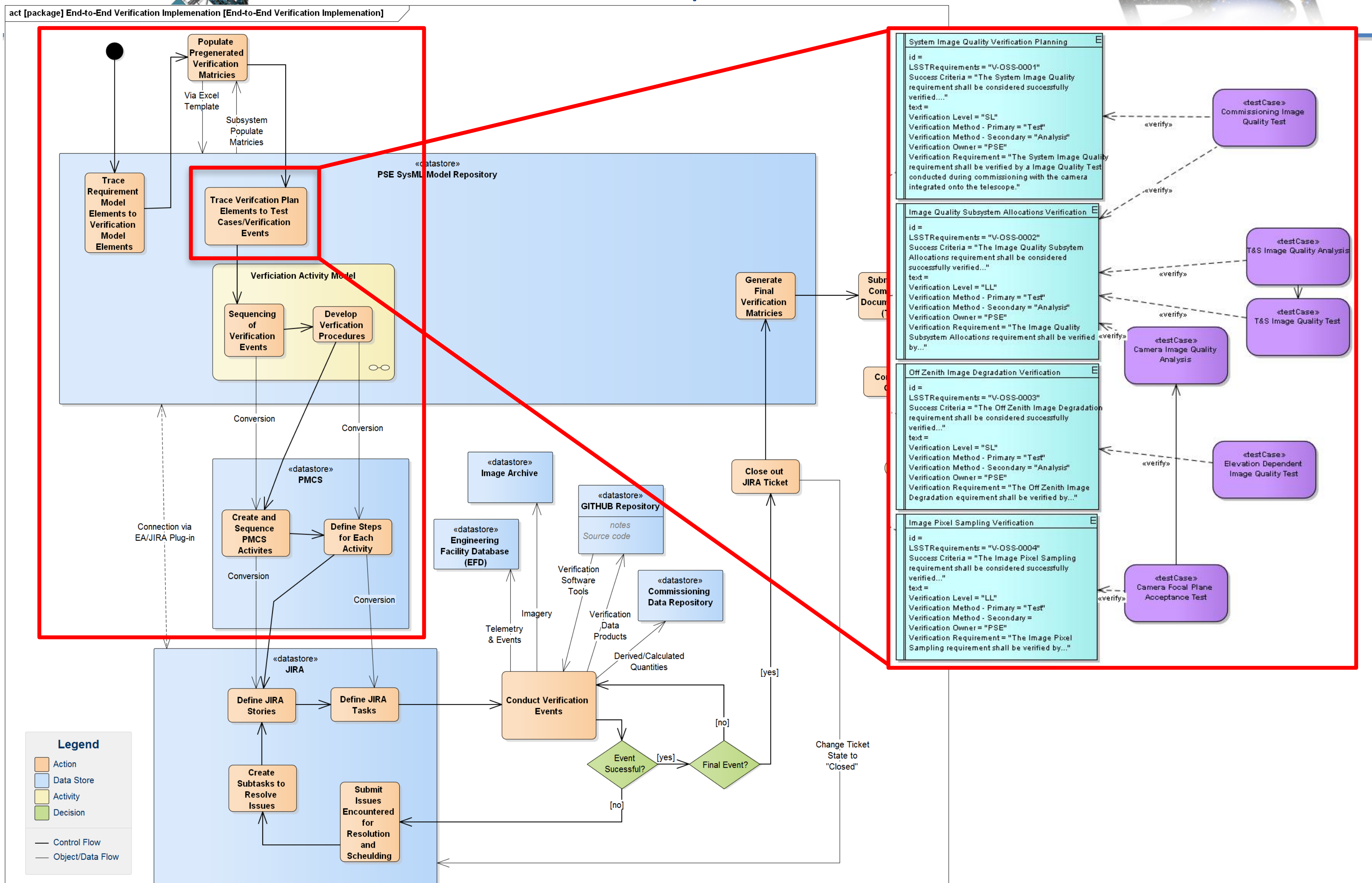
DRAFT IN WORK

End-to-End Verification Implementation Process



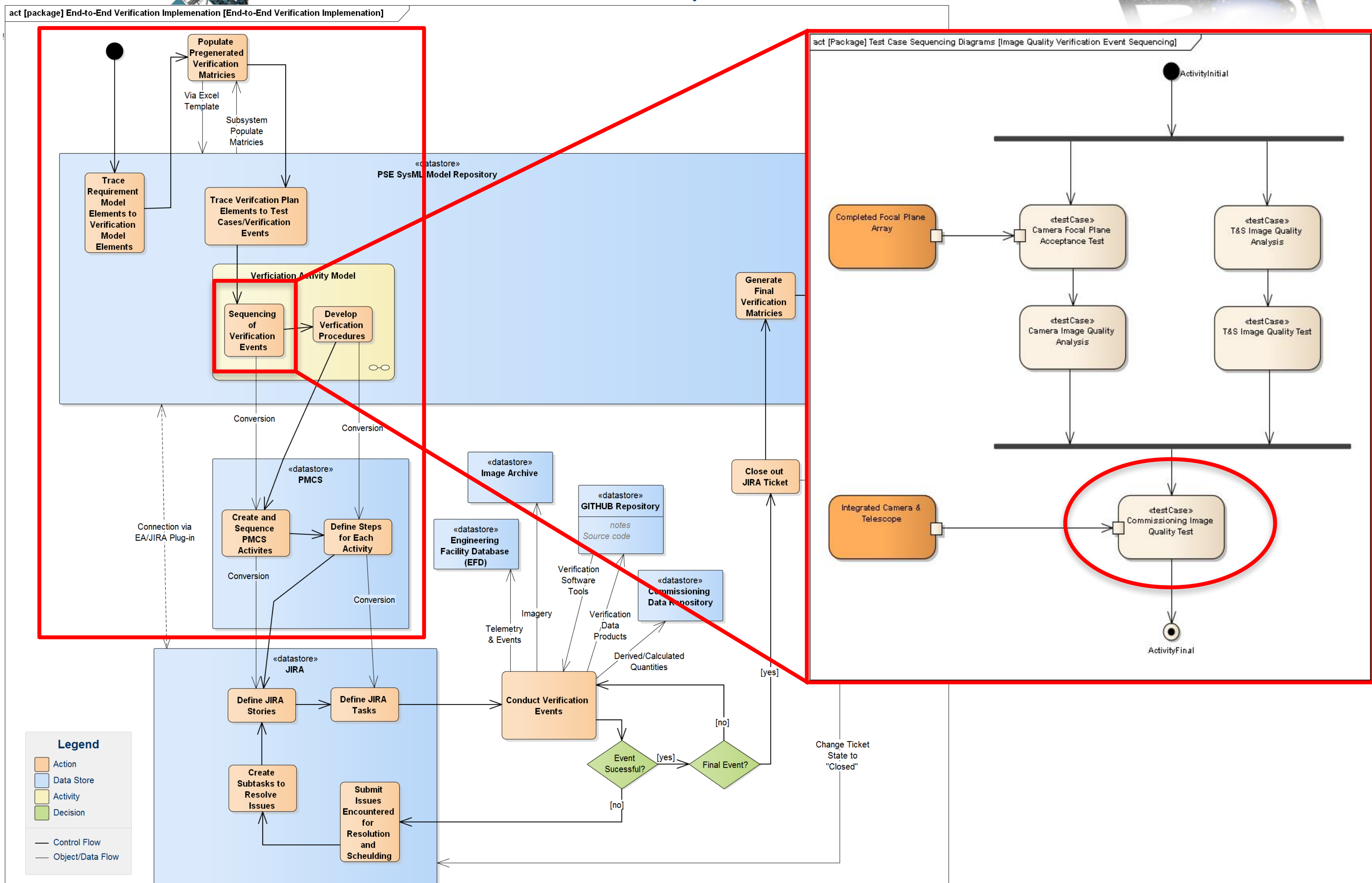
DRAFT IN WORK

End-to-End Verification Implementation Process



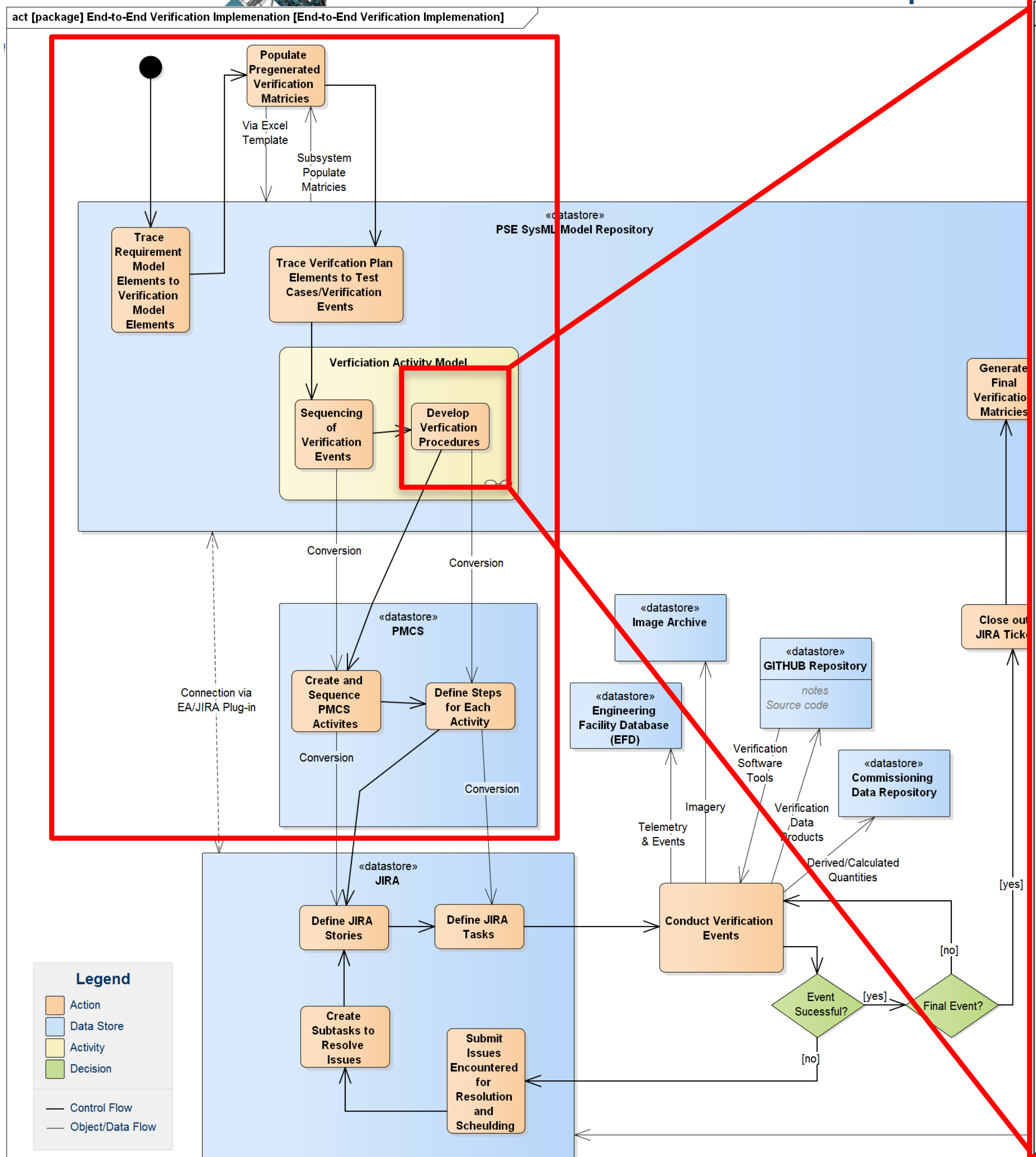
DRAFT IN WORK

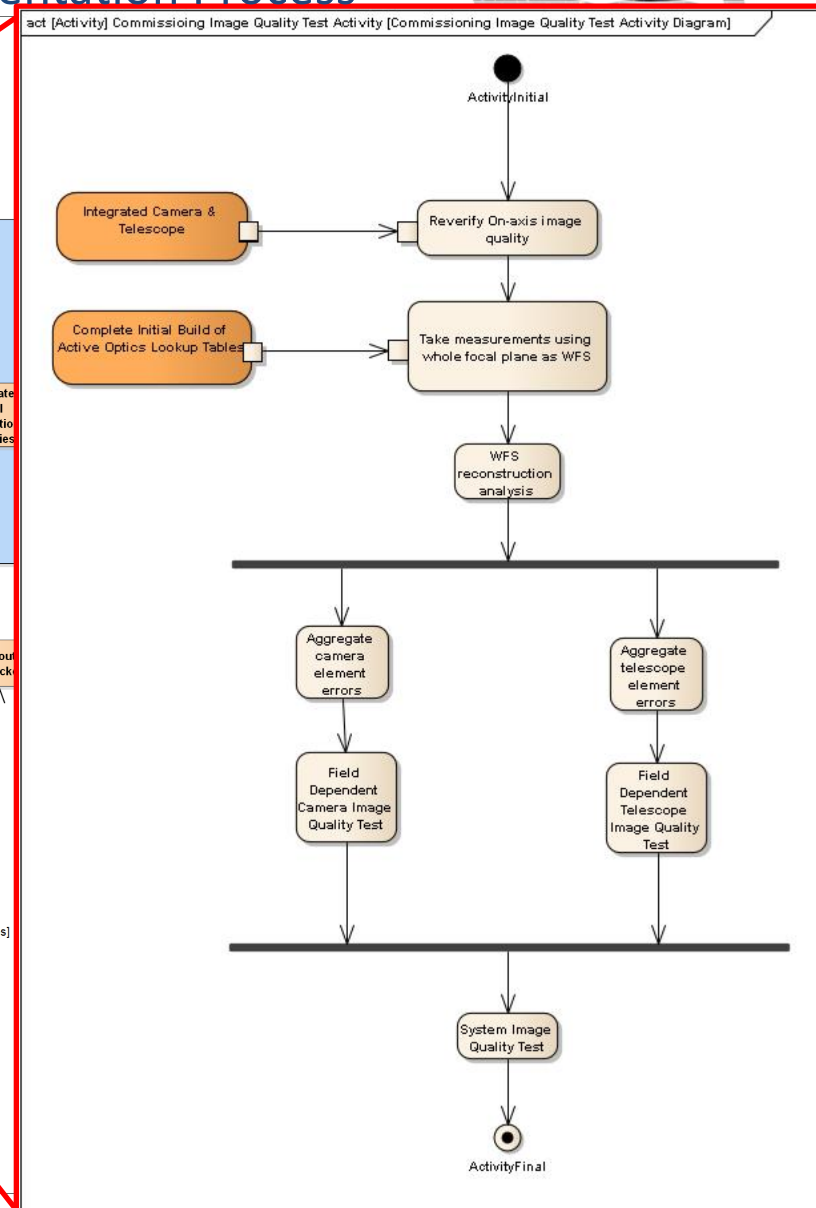
End-to-End Verification Implementation Process



DRAFT IN WORK

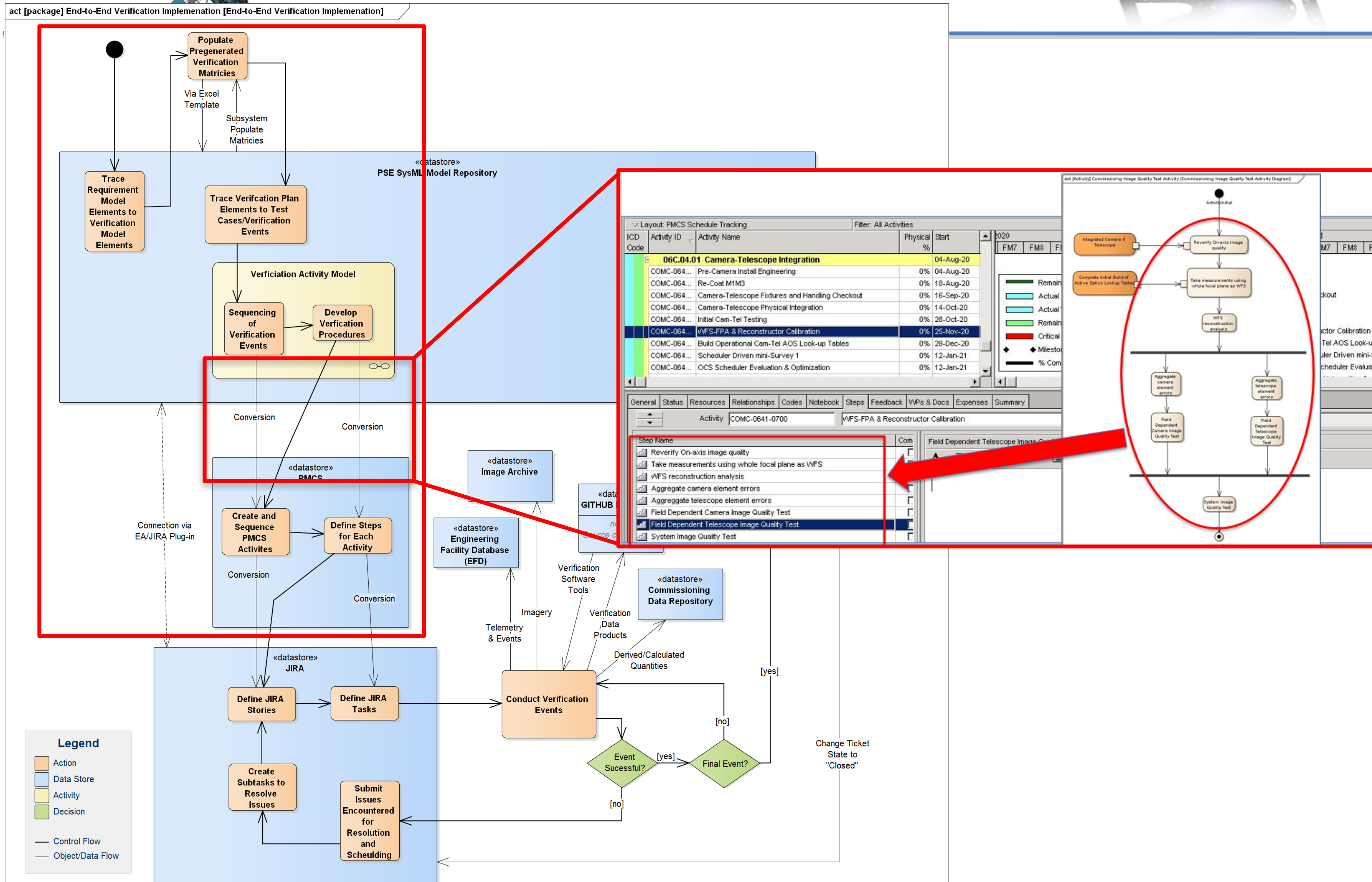
End-to-End Verification Implementation Process





DRAFT IN WORK

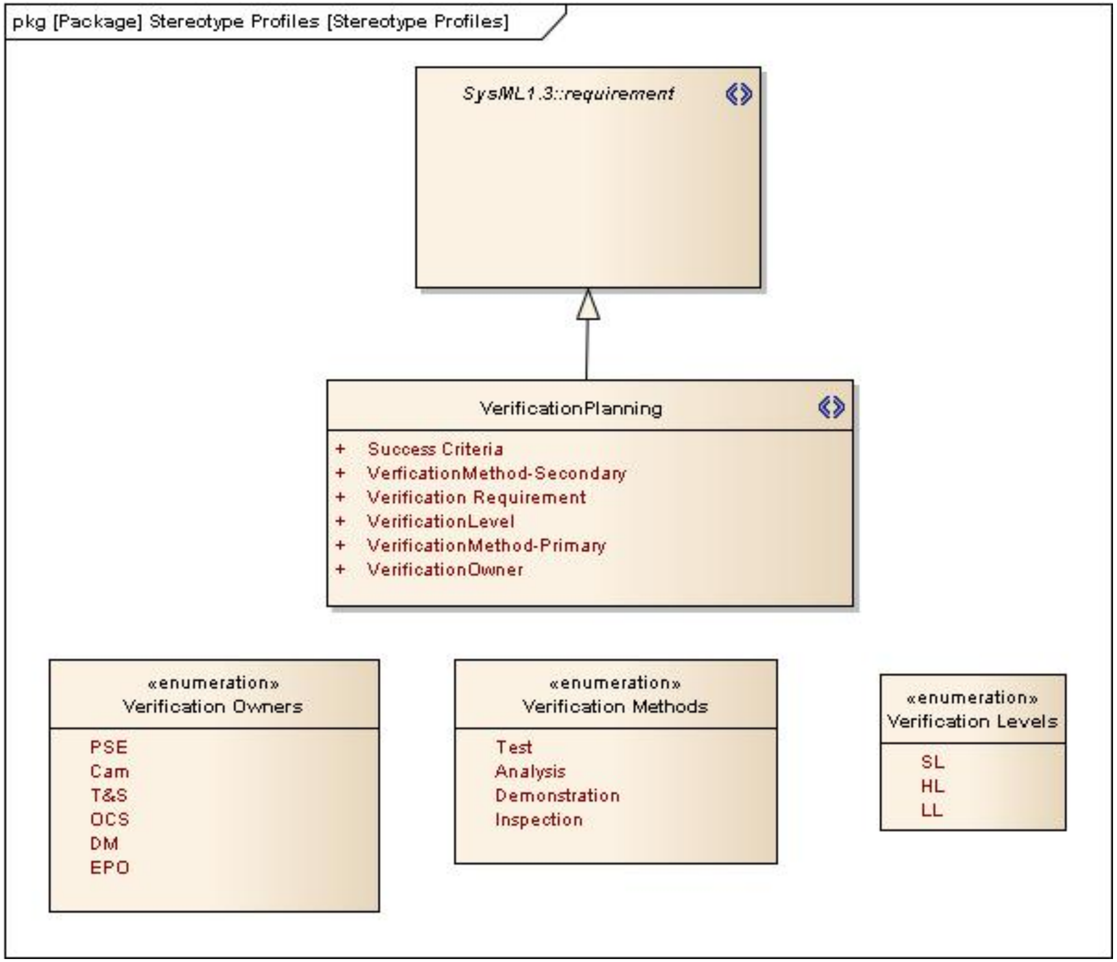
End-to-End Verification Implementation Process



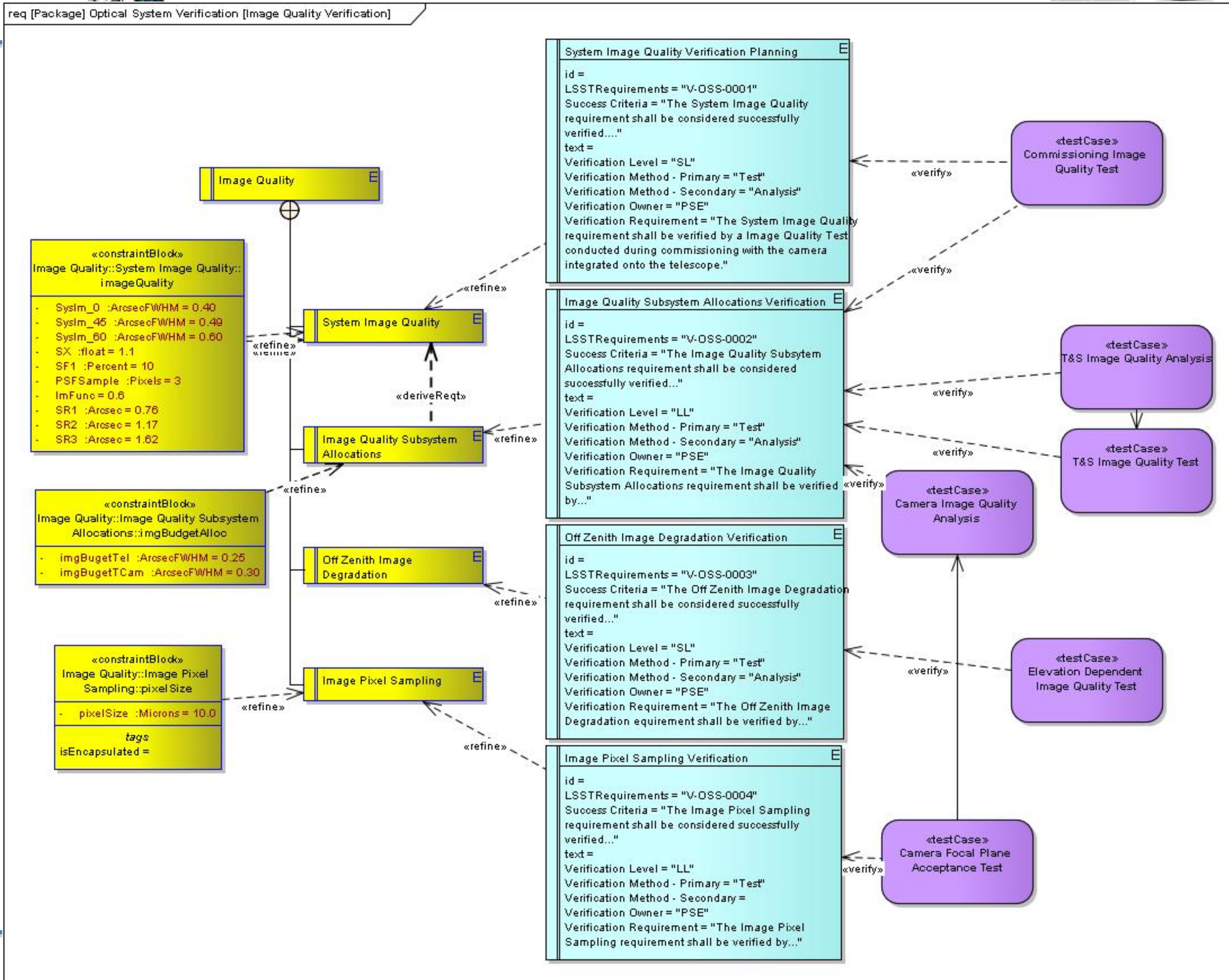
DRAFT IN WORK

– 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

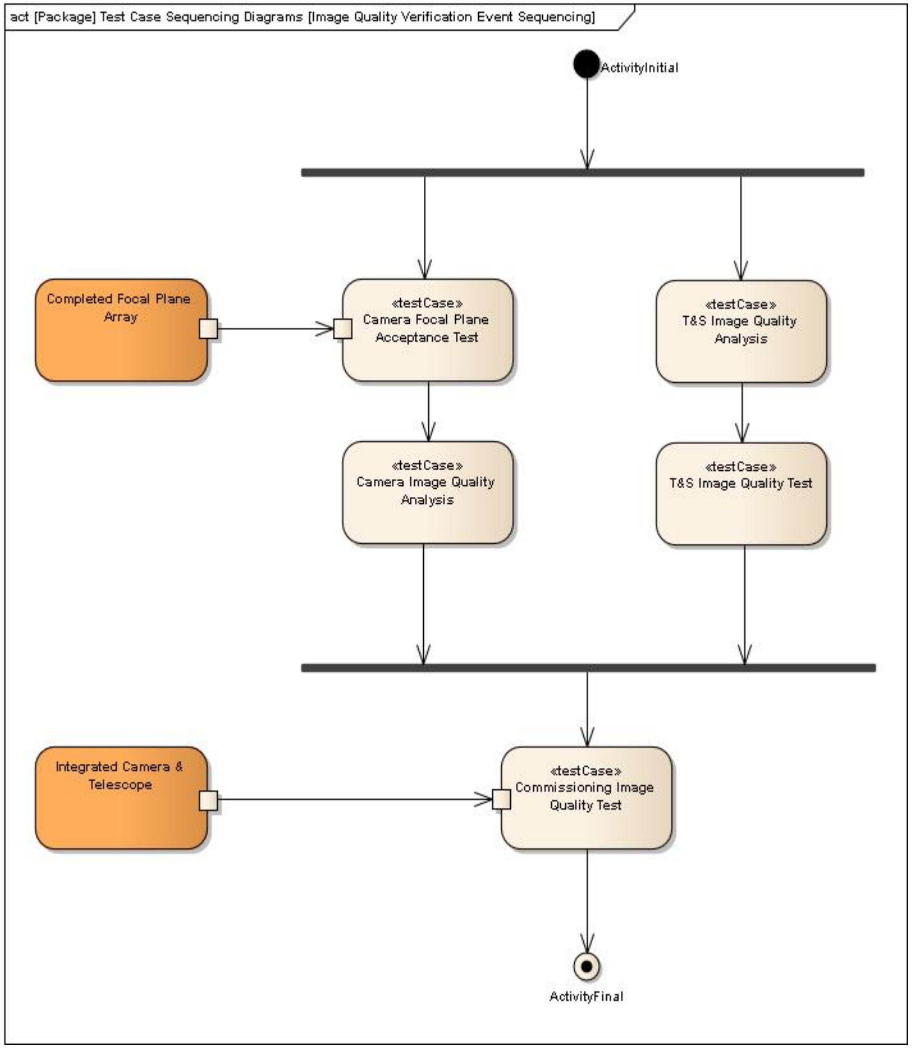


Creation of Verification Plans & Test Cases in the Model

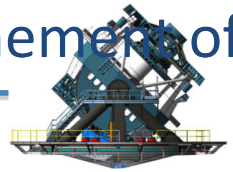


Sequencing Test Cases (Verification Events)

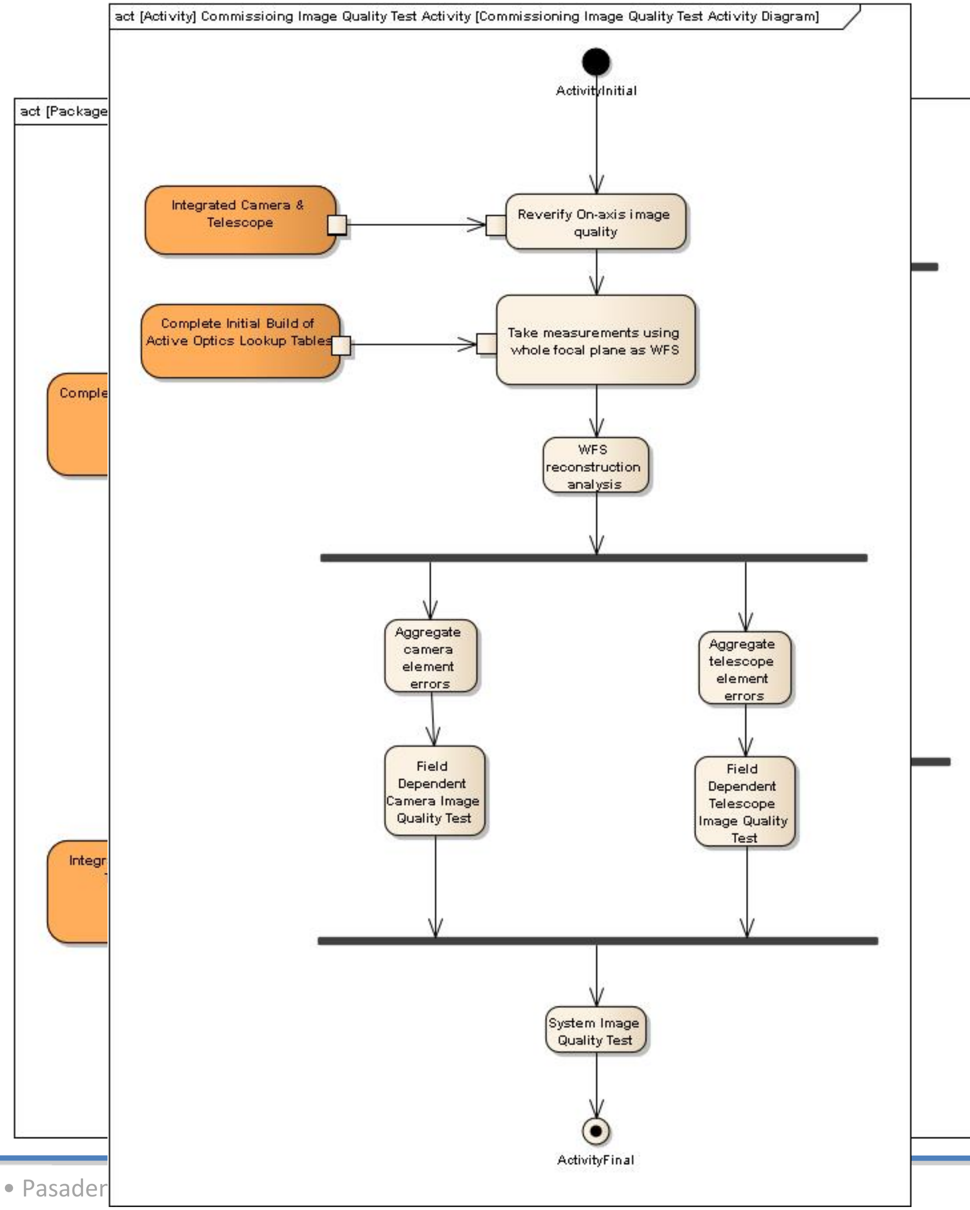
– Test Cases (Verification Events) are sequenced on Activity Diagrams to show:
  - Predecessor/ successor relationships
  - Events that are conducted in parallel/ series
  - Outside constraints that must be met before a Verification Event can be executed
- Results can be used to validate or update the project's schedule for the Commissioning period.



Refinement of Individual Test Cases (Verification Events)

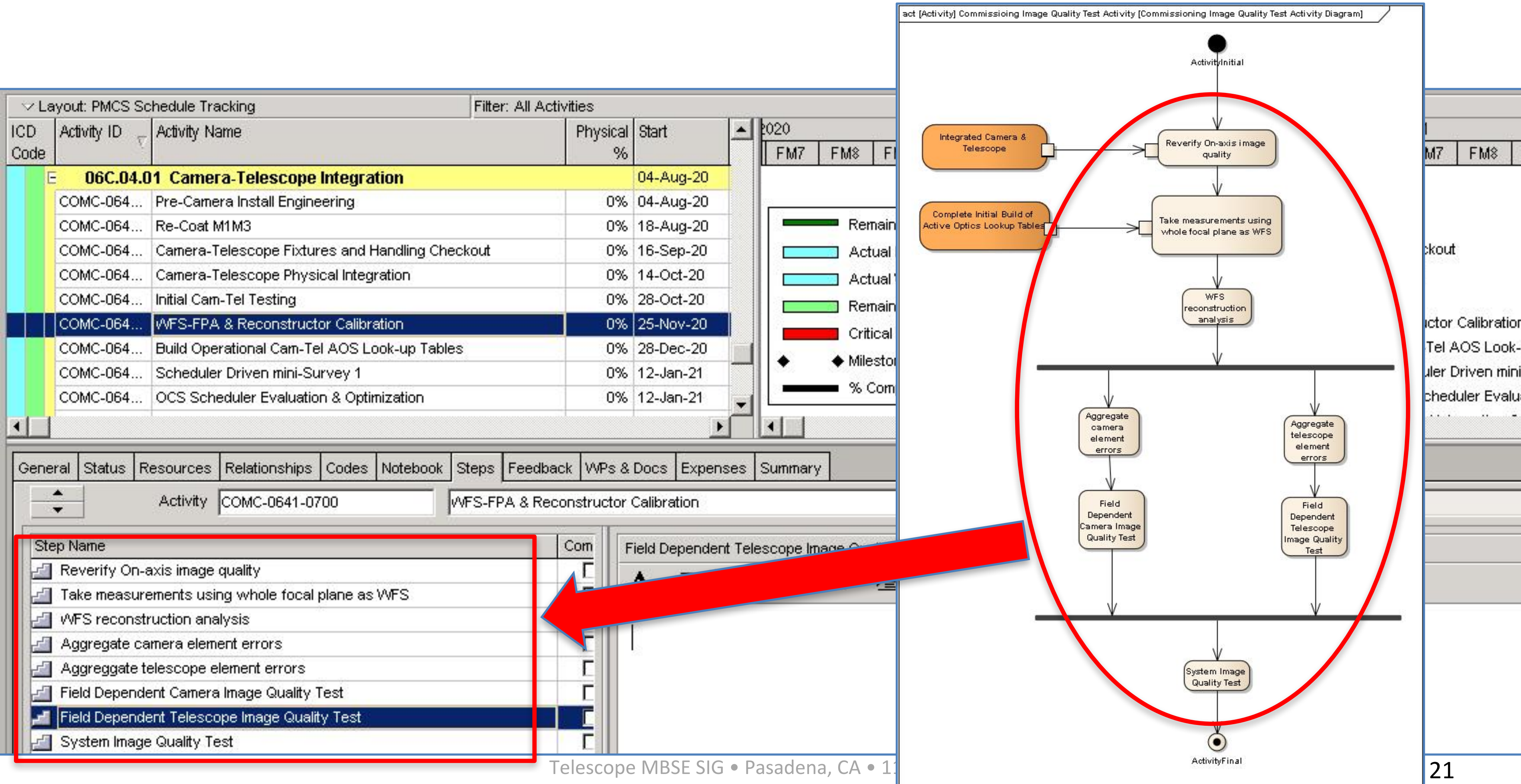


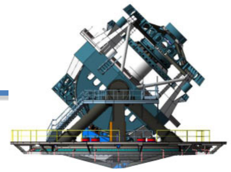
- As plans mature, individual Verification Events can be further detailed via association with its own detailed behavior diagram
- Serves as refined and more detailed input to the commissioning planning effort
  - Can be used directly as inputs to writing detailed test & analysis procedures



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

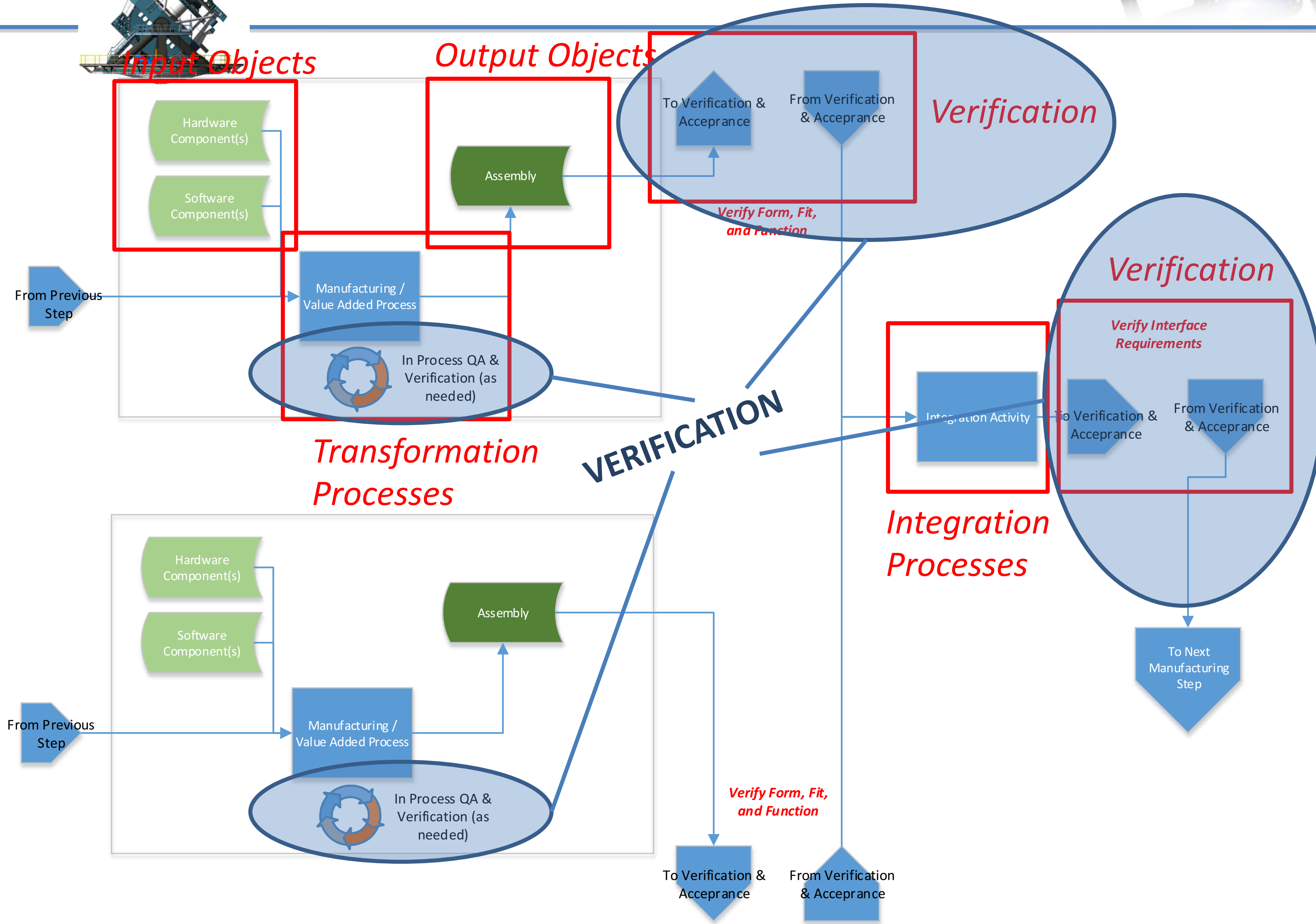




Verification as Part of AIV

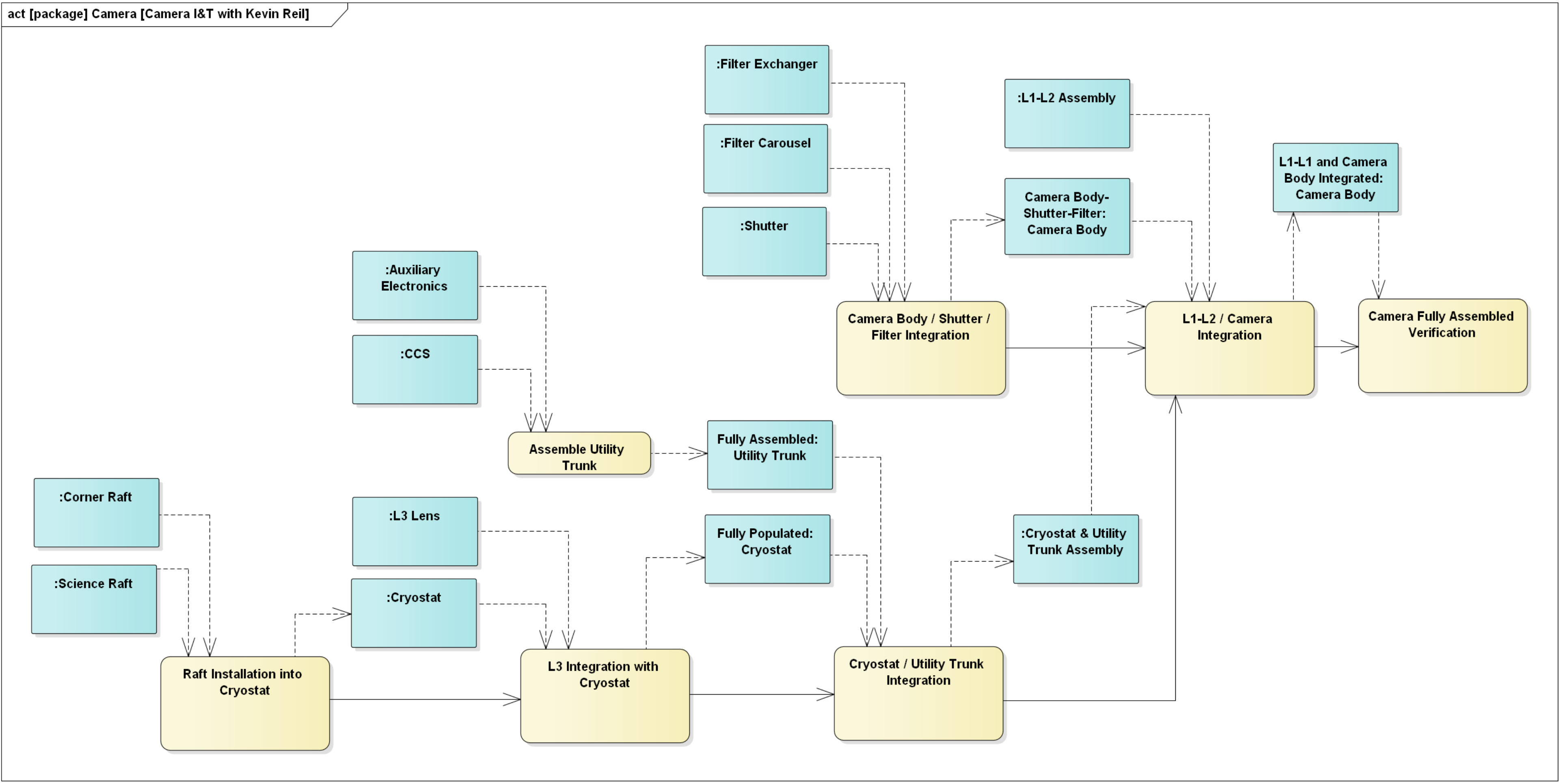
- Verification is one critical aspect of the broader manufacturing, assembly, integration, and verification set of activities
- Project Systems Engineering needs to understand the early integration and verification activities being conducted by the subsystems that impact system level requirements, interfaces, 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

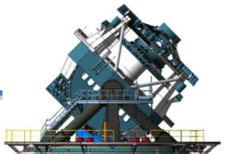

Input Objects
Output Objects
Hardware Component(s)
Software Component(s)
Assembly
To Verification & Accerprance
From Verification & Accerprance
Verification
Verify Form, Fit, and Function
From Previous Step
Manufacturing / Value Added Process
In Process QA & Verification (as needed)
Transformation Processes
VERIFICATION
Verification
Verify Interface Requirements
Integration Activity
To Verification & Accerprance
From Verification & Accerprance
Integration Processes
To Next Manufacturing Step
Hardware Component(s)
Software Component(s)
Assembly
From Previous Step
Manufacturing / Value Added Process
In Process QA & Verification (as needed)
Verify Form, Fit, and Function
To Verification & Accerprance
From Verification & Accerprance

Ongoing work – Initial AIV Models

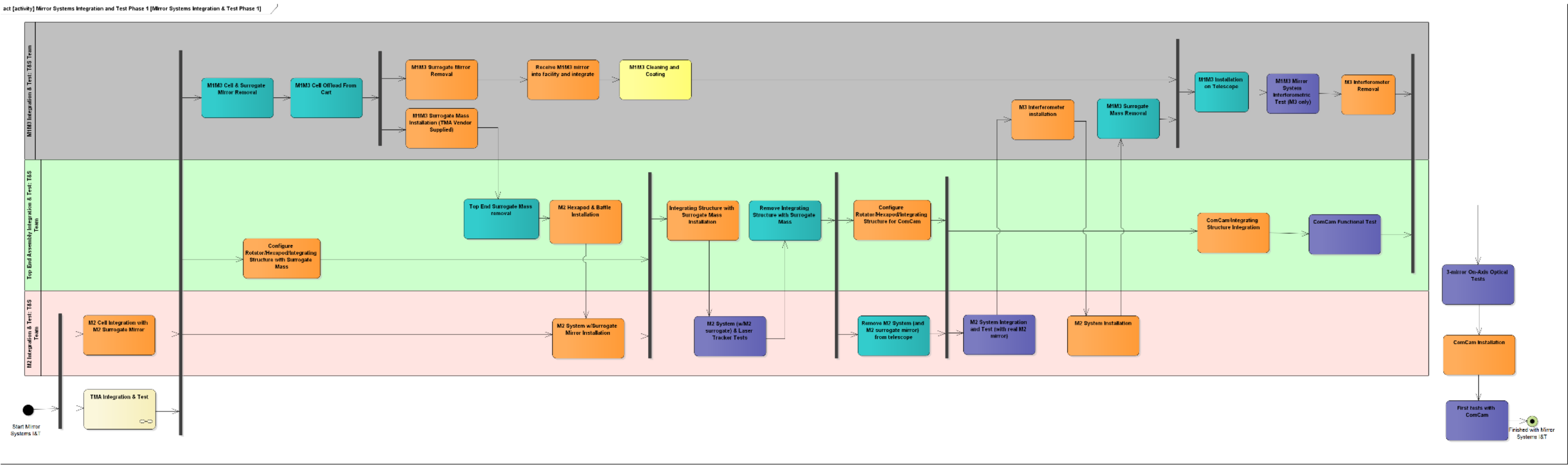
– Partial Camera AIV



Ongoing work – Initial AIV Models

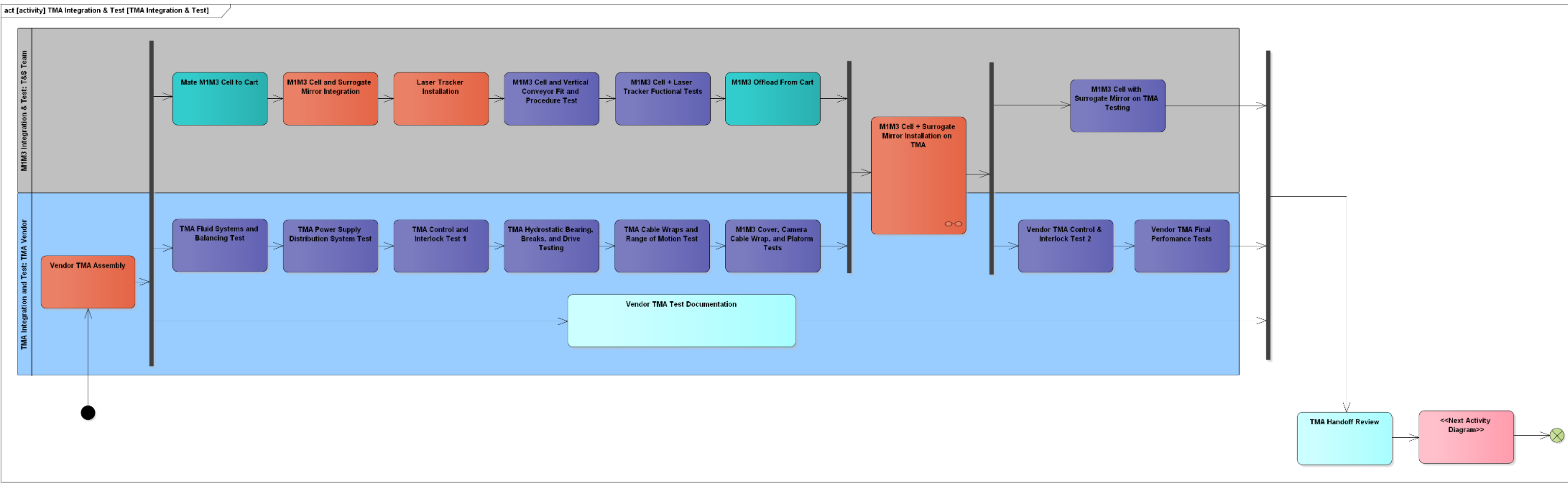


– T&S Mirror Systems Integration and Test Phase 1

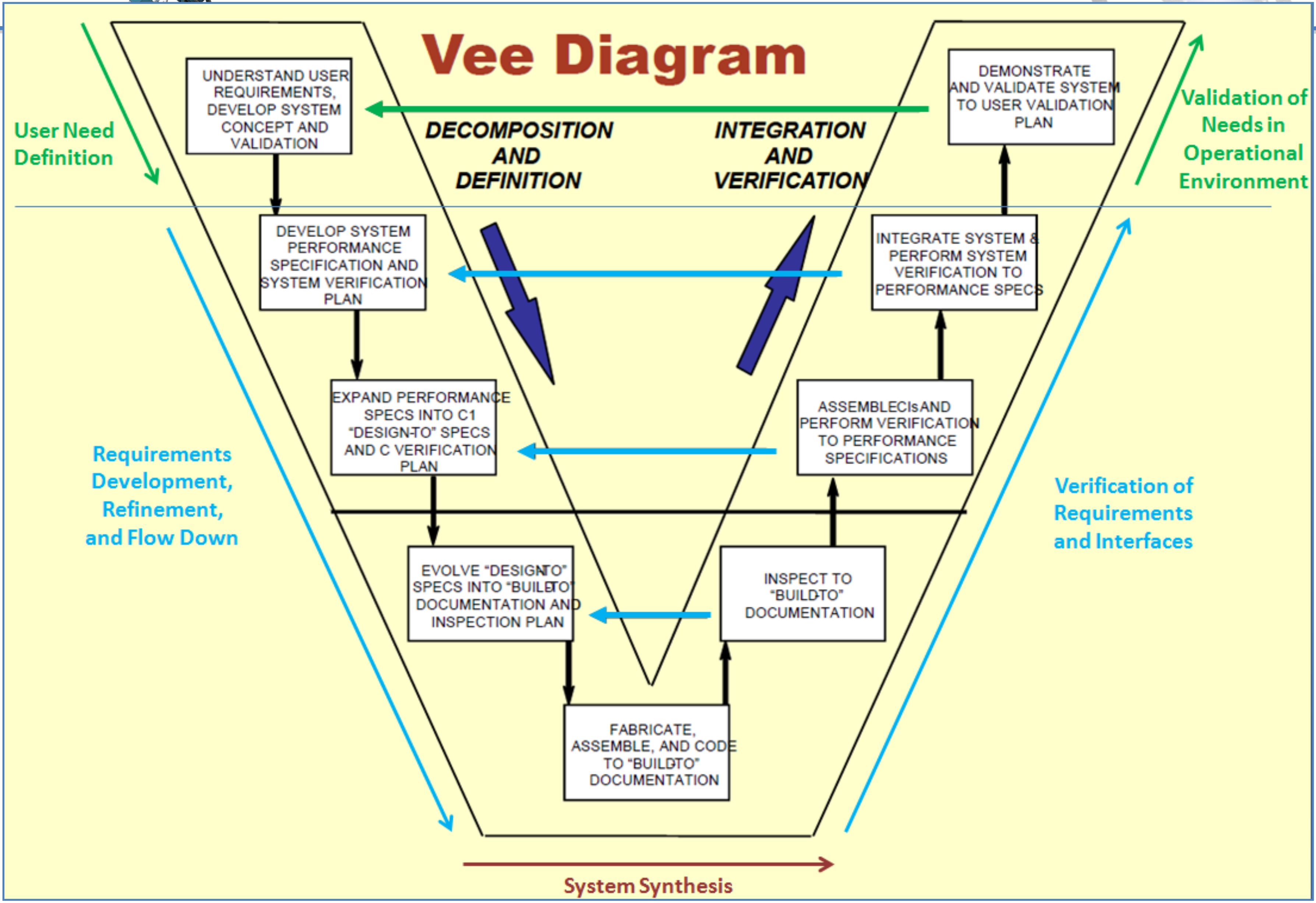


Ongoing work – Initial AIV Models

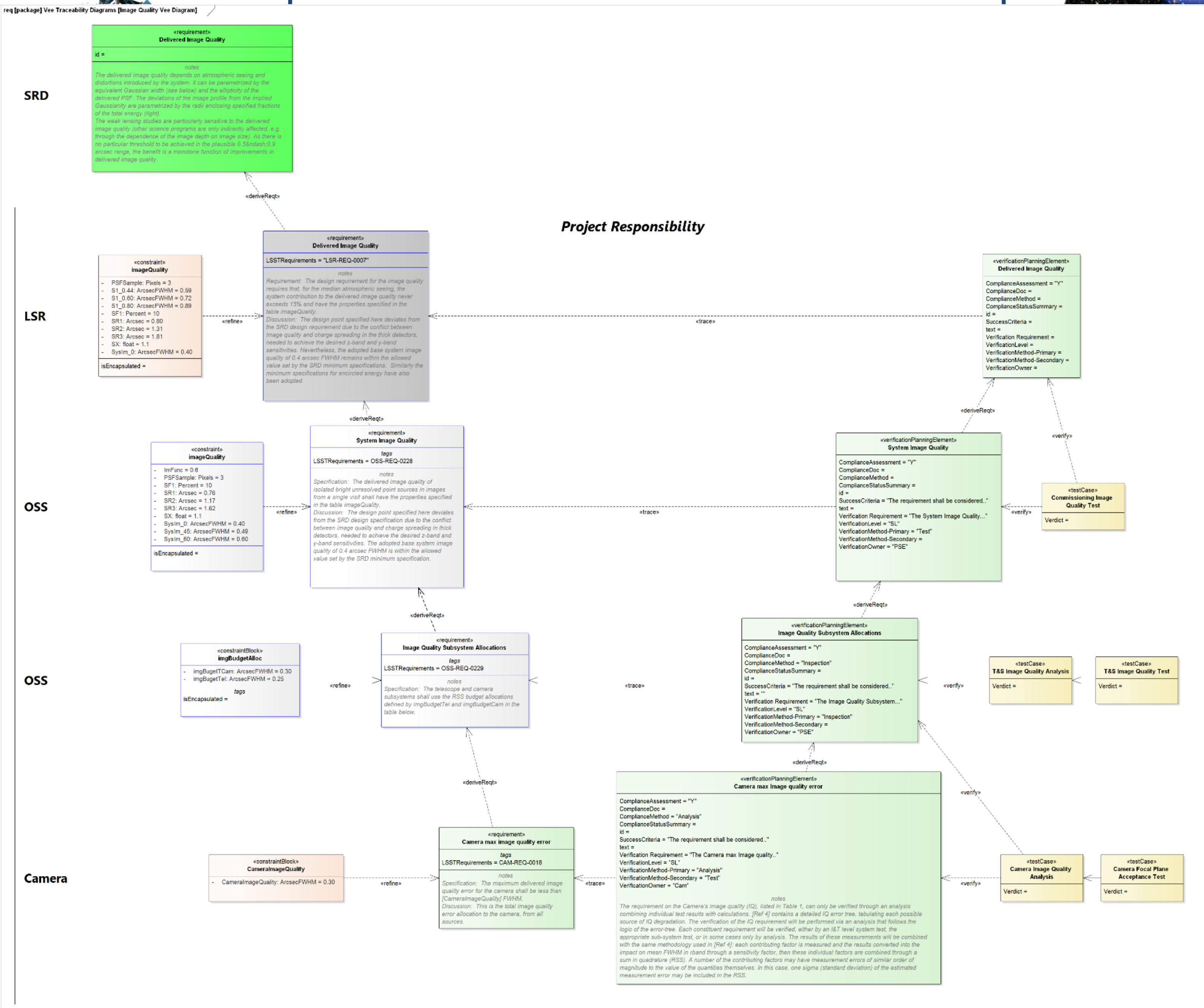
- TMA Integration and Test



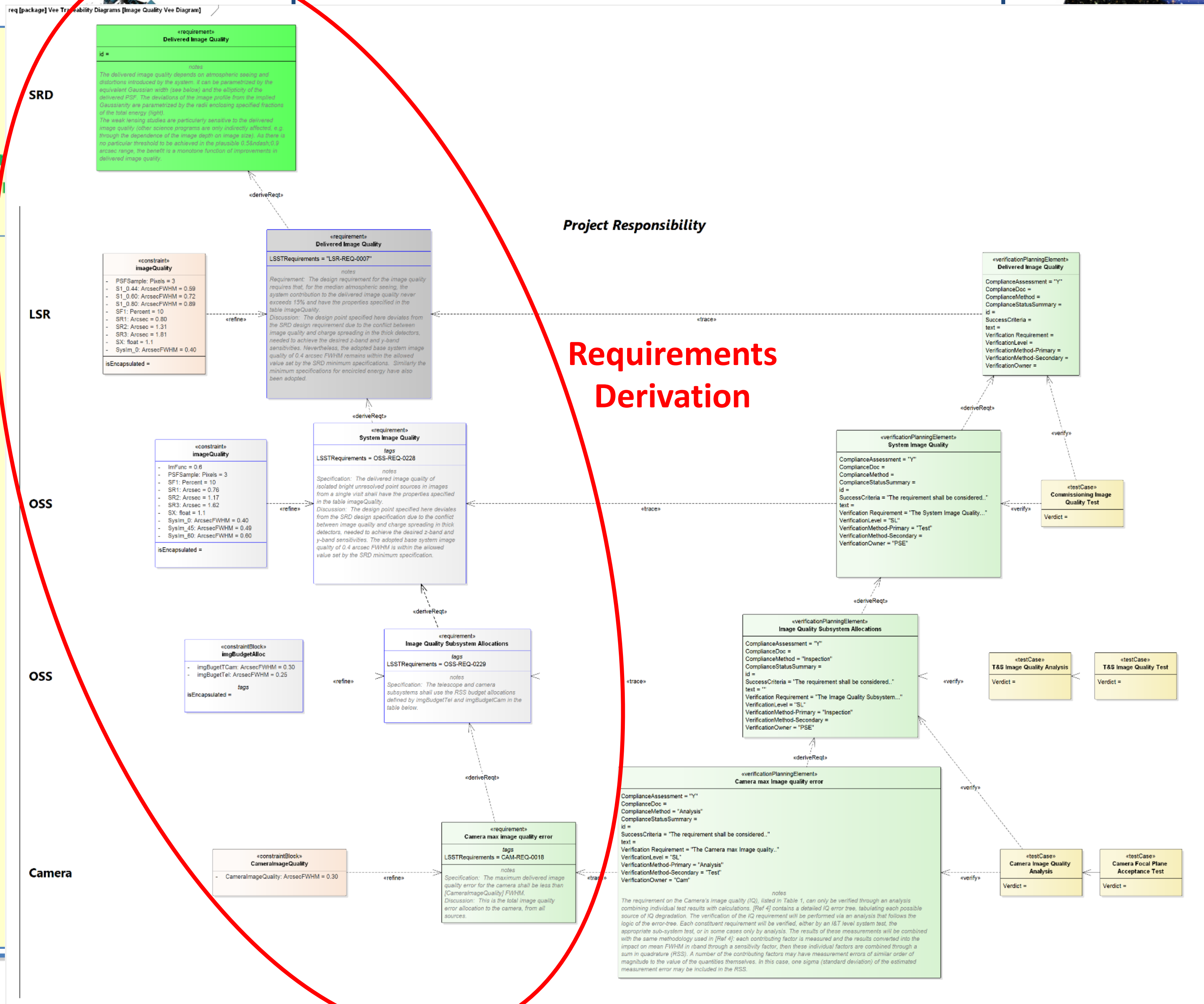
A Requirements to Verification Plan Example



A Requirements to Verification Plan Example



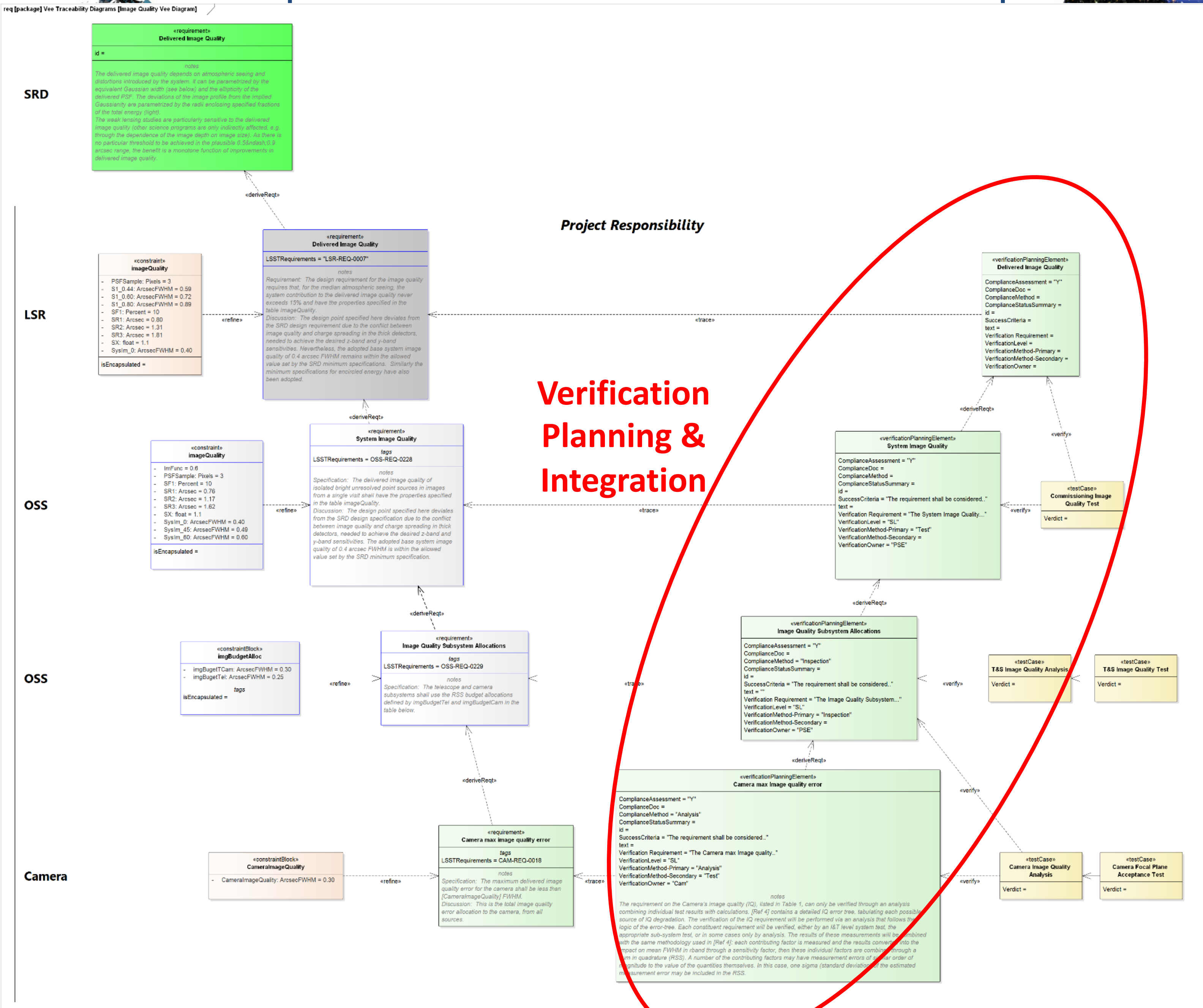
A Requirements to Verification Plan Example



A Requirements to Verification Plan Example

User
Defi

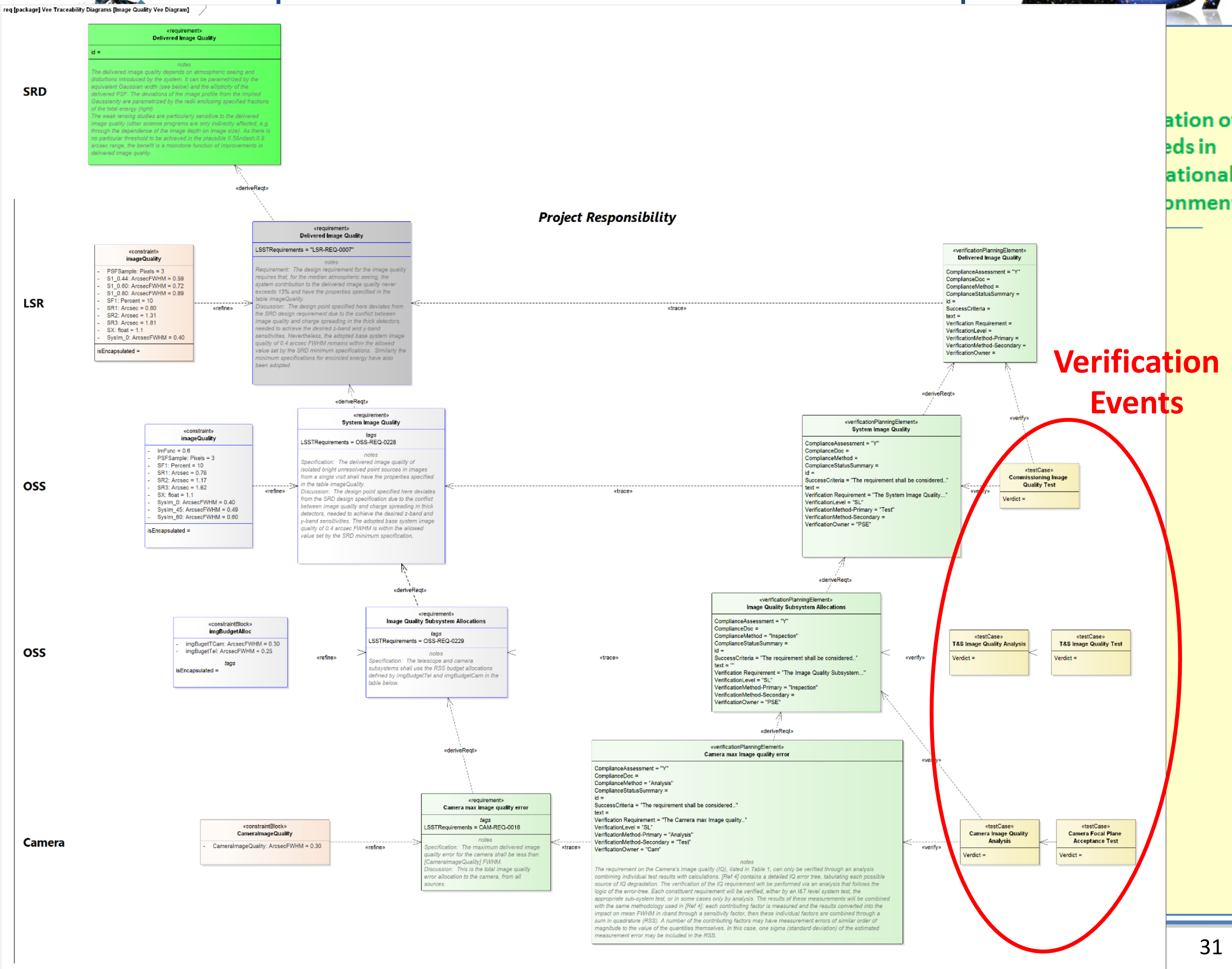
ation of
eds in
ational
onment



A Requirements to Verification Plan Example

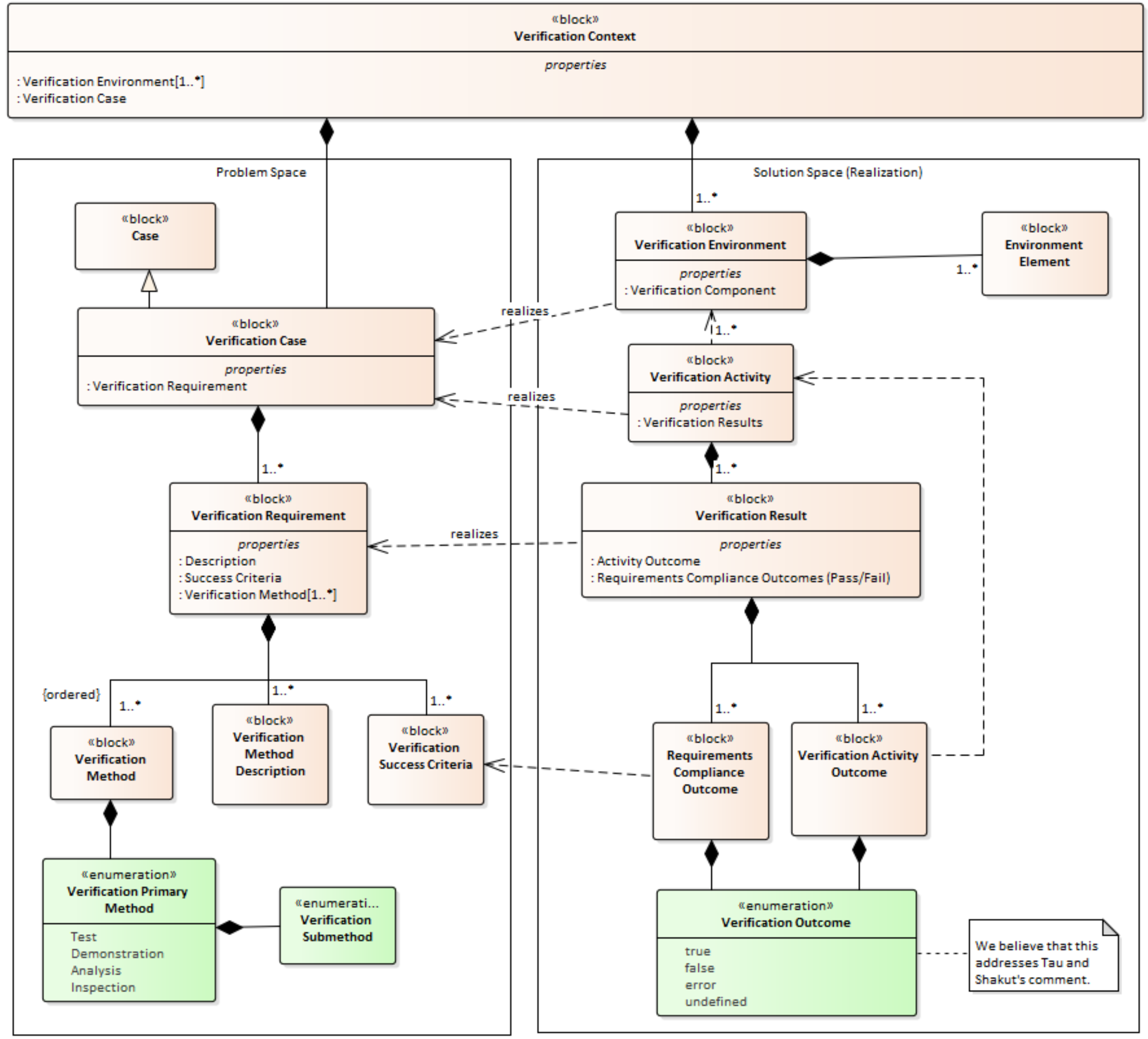
User
Defi

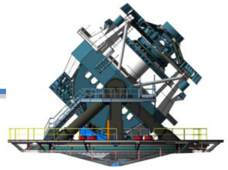
ation of
eds in
ational
onment



SysML v2 Development Work for Verification

- 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

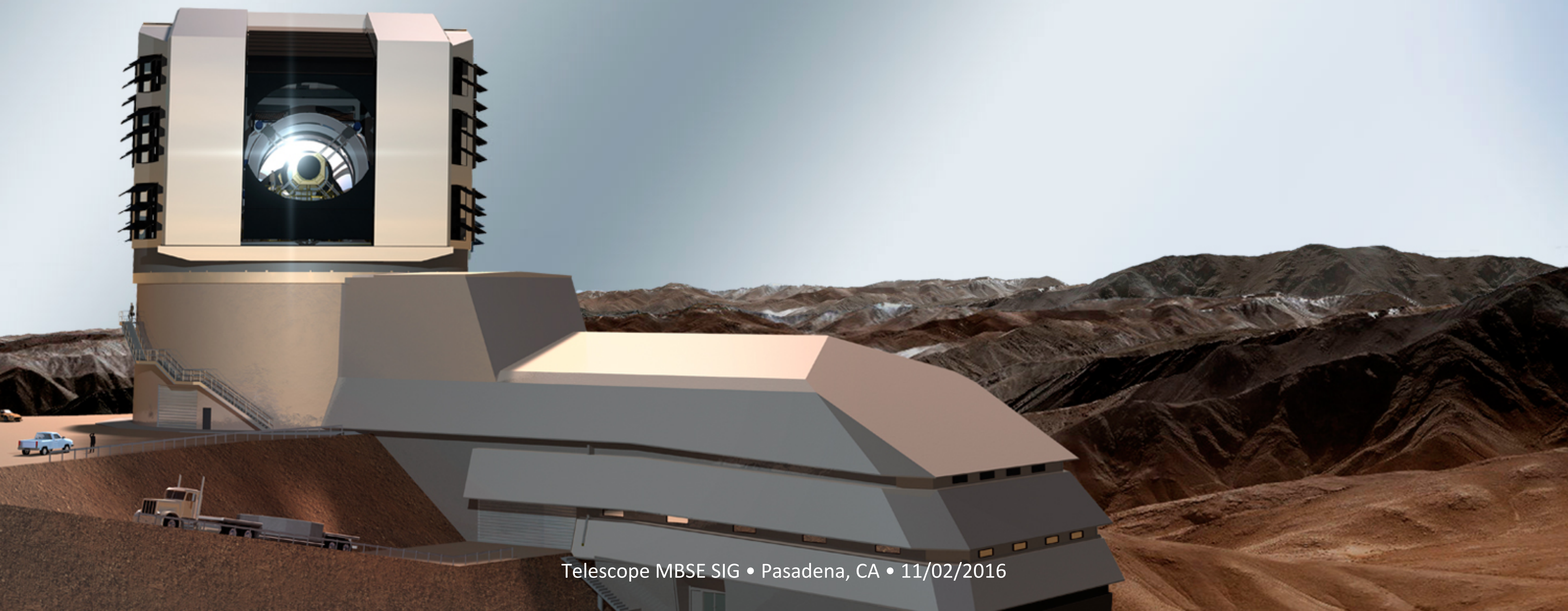




End

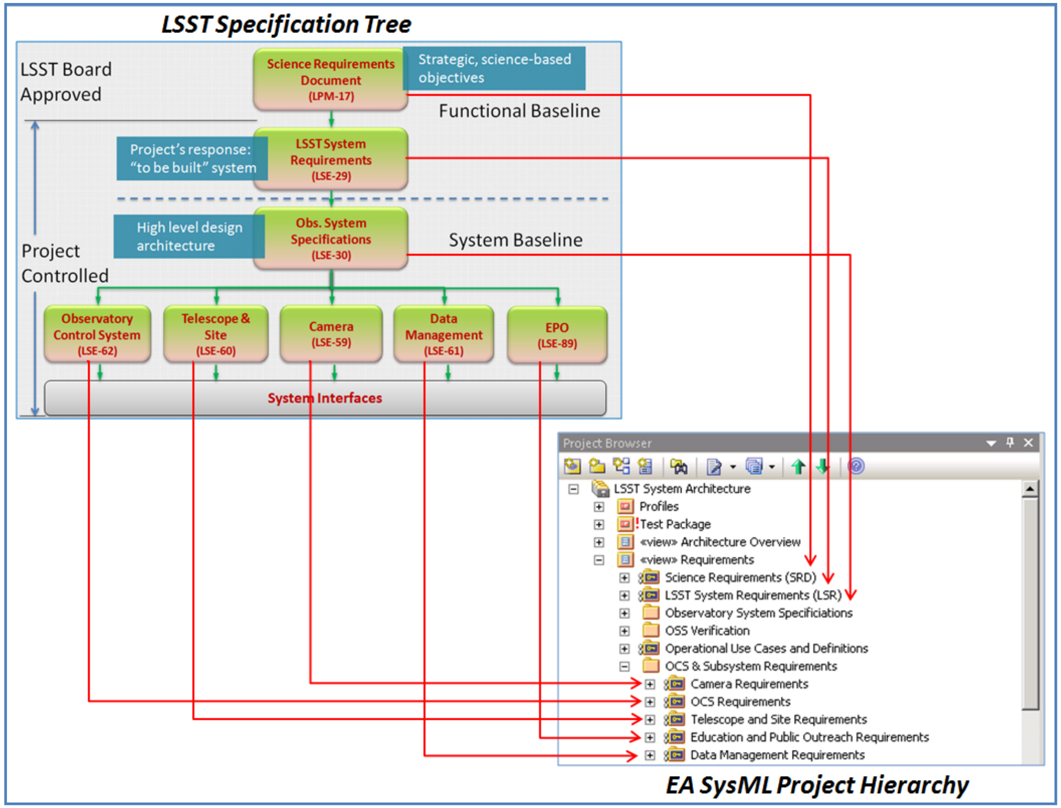
Backup Slides

Requirements Engineering



SysML Implementation – Defining Requirements

- 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.

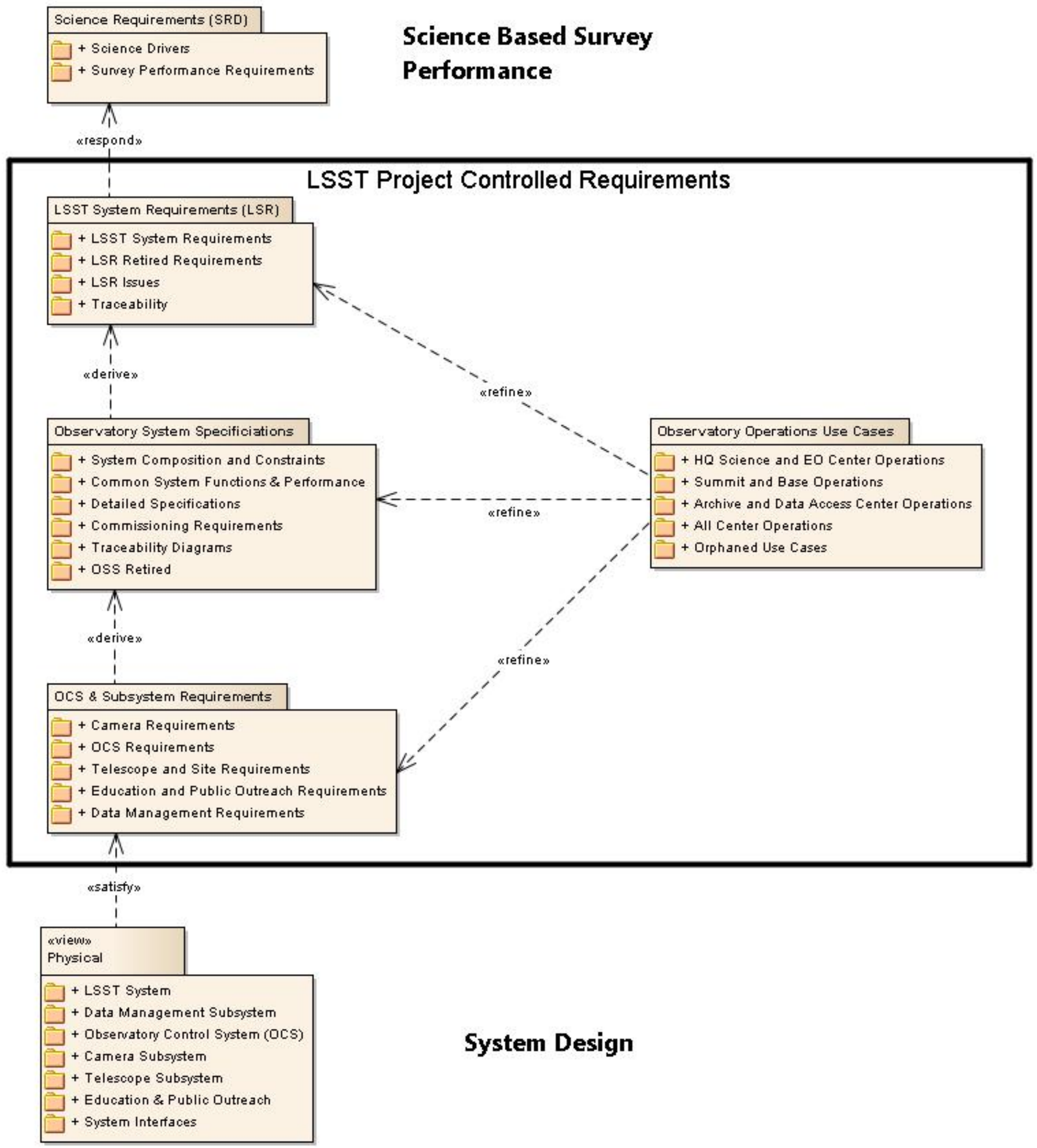


Requirements View Captures Flow Down and Traceability

Packages are used to manage our requirements for version control and document generation.

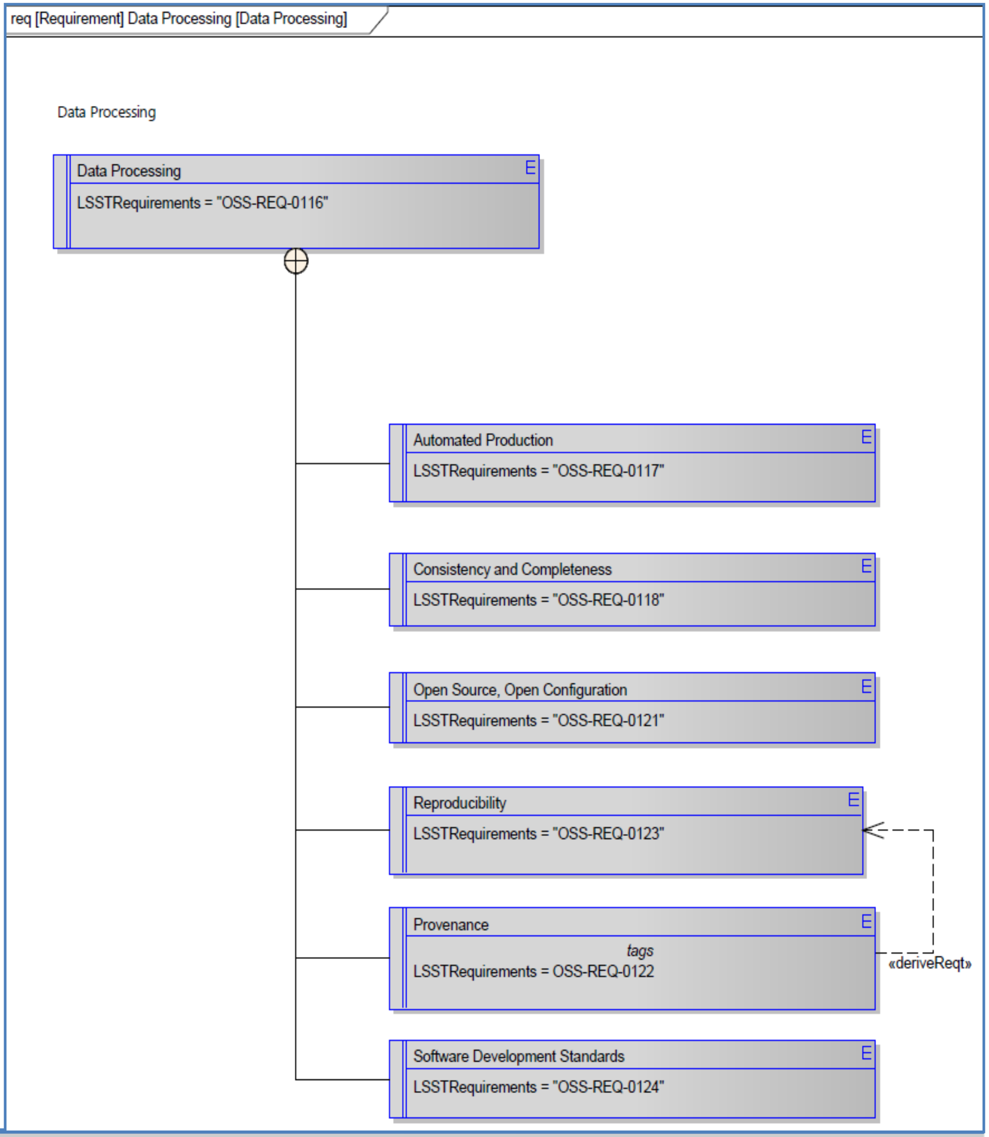
All of the LSST's system level requirements documents are generated from the model.

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

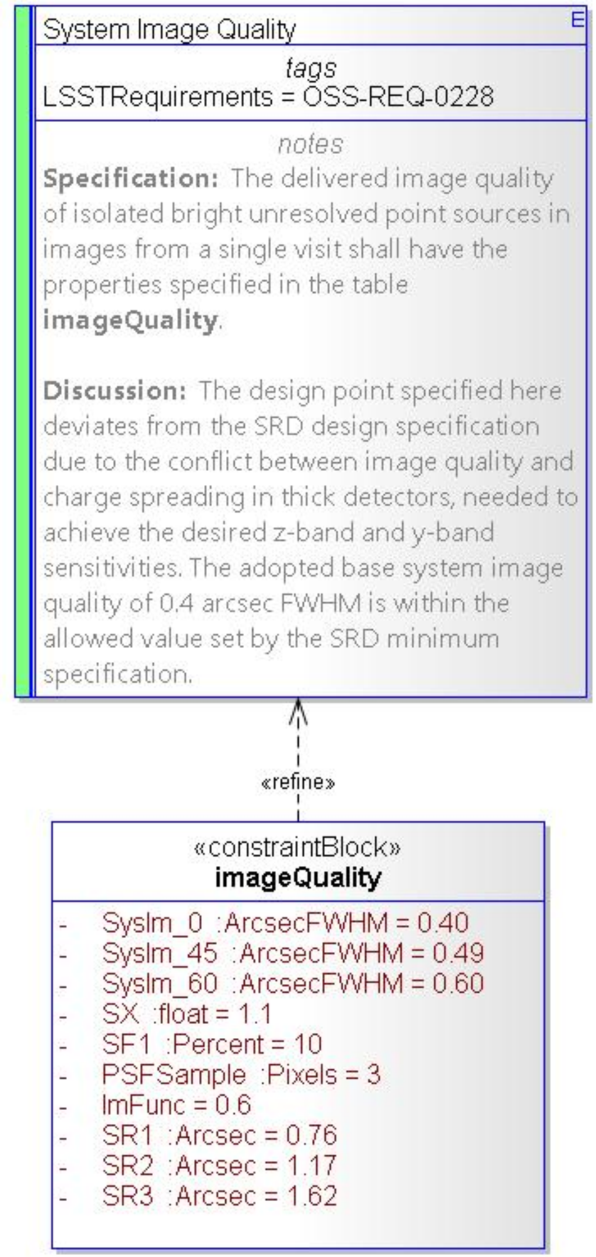
Requirement title

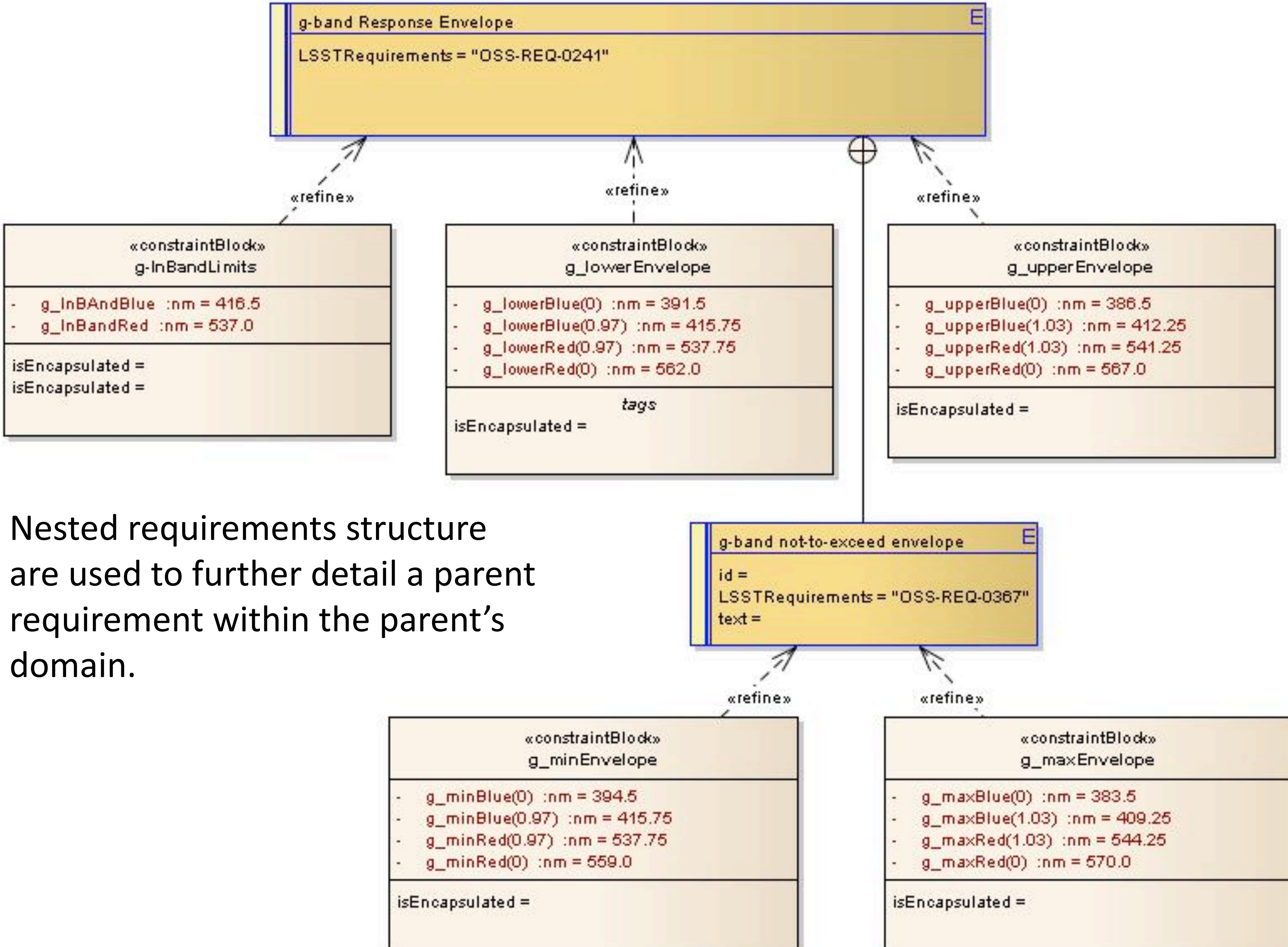
Tool extension enforces unique ID tag value

Requirement text

Clarifying discussion text (if needed)

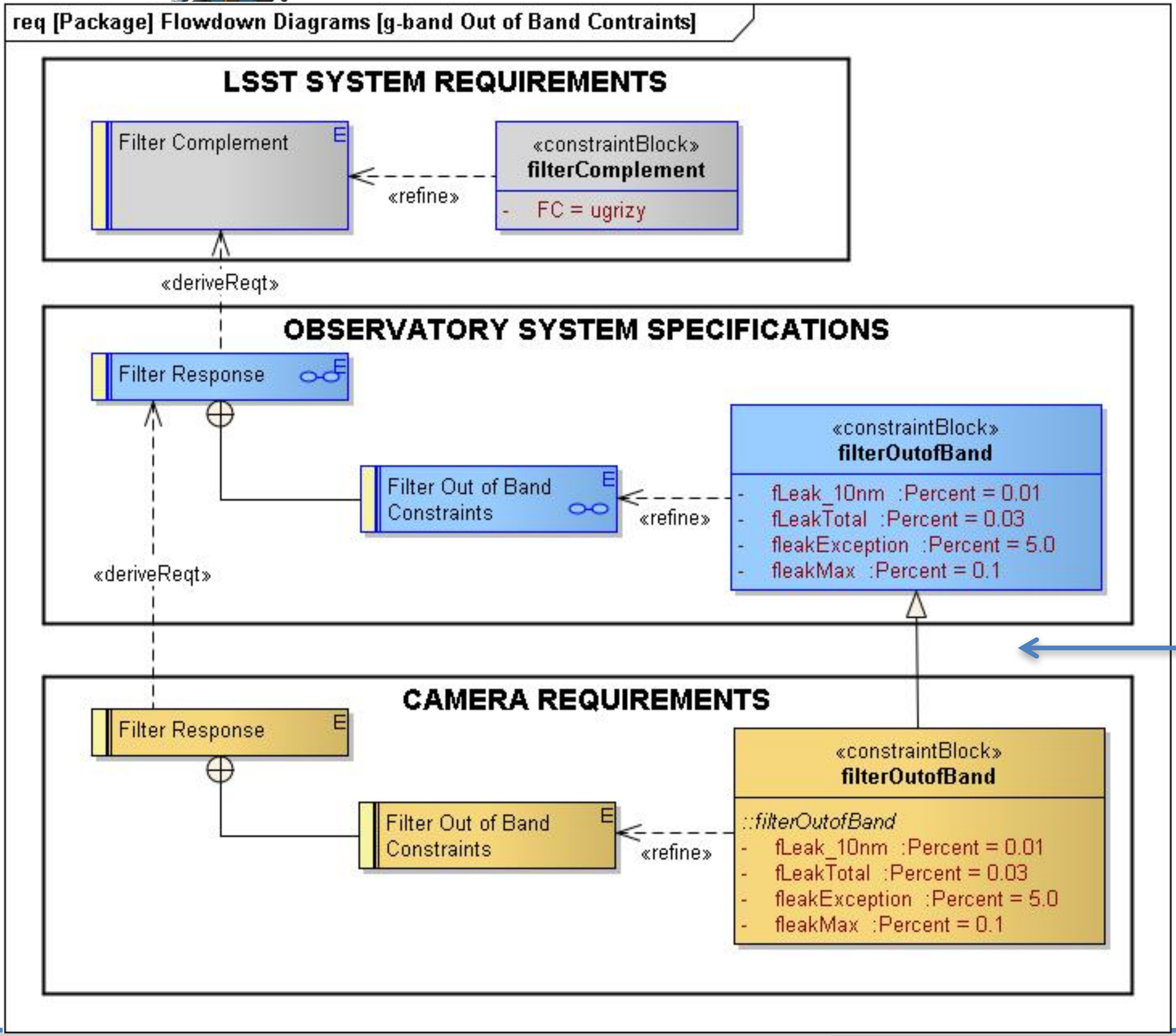
SysML constraint blocks are used for quantitative attributes refines the requirement





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

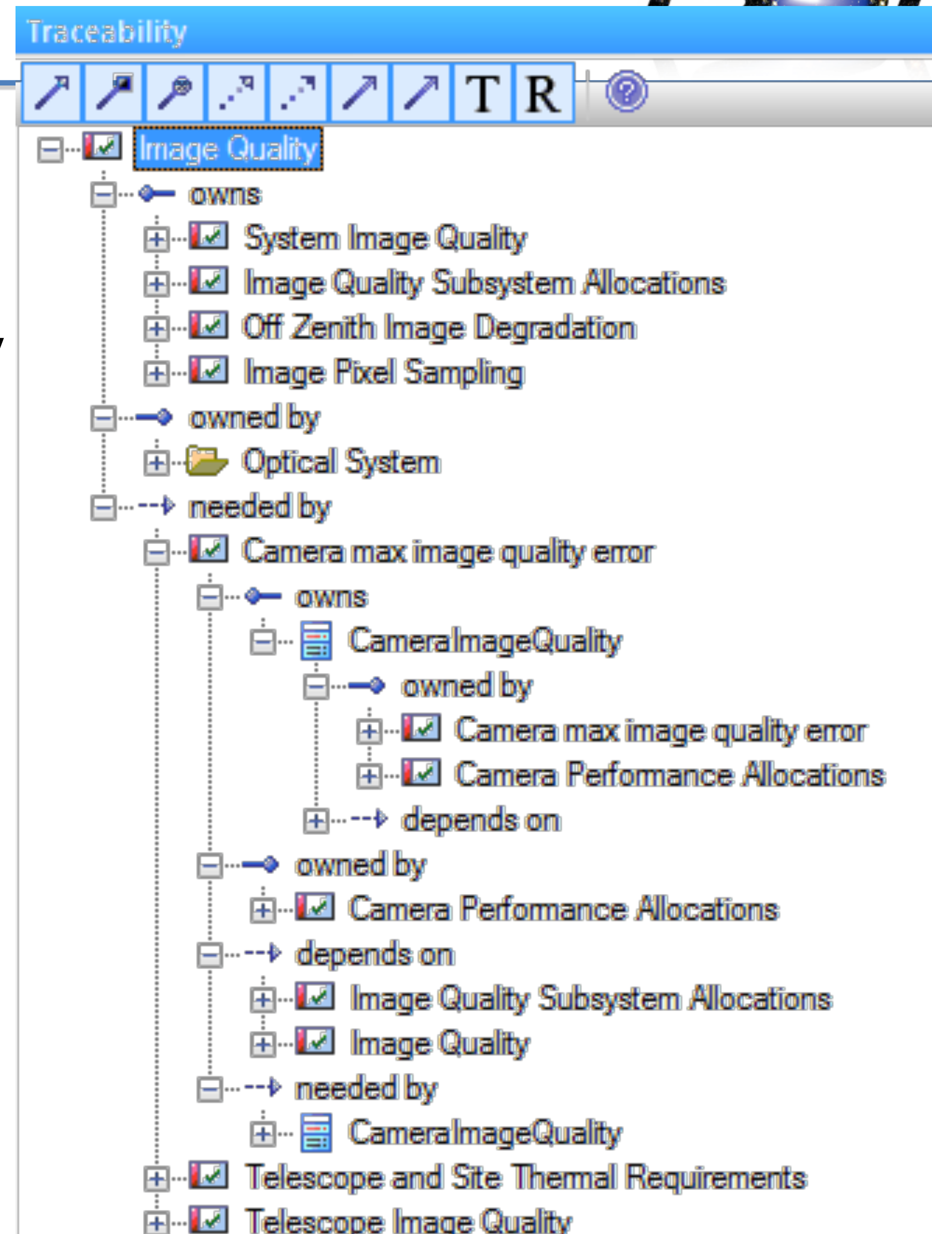
Generalization relationship between constraint blocks allows attribute inheritance

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

LARGE SYNOPTIC SURVEY TELESCOPE

LSST Observatory System Specifications LSE-30 Latest Revision 1/27/2015

3.3.1.5 g-band Response Envelope

ID: OSS-REQ-0241 **Last Modified: 6/19/2014**

Specification: The area weighted mean g-band filter response normalized to the in-band average (as measured between **g_inBandBlue** and **g_inBandRed**) shall lie between the upper and lower envelopes defined in the tables below:

g-InBandLimits

Description	Value	Unit	Name
The in-band blue limit for the g-band filter response normalization.	416.5	nm	g_InBAndBlue
The in-band red limit for the g-band filter response normalization.	537.0	nm	g_InBandRed

g_lowerEnvelope

Description	Value	Unit	Name
The blue side zero response wavelength of the g-band lower envelope.	391.5	nm	g_lowerBlue(0)
The blue side 0.97% response wavelength of the g-band lower envelope.	415.75	nm	g_lowerBlue(0.97)
The red side 0.97% response wavelength of the g-band lower envelope.	537.75	nm	g_lowerRed(0.97)
The red side zero response wavelength of the g-band lower envelope.	562.0	nm	g_lowerRed(0)

g_upperEnvelope

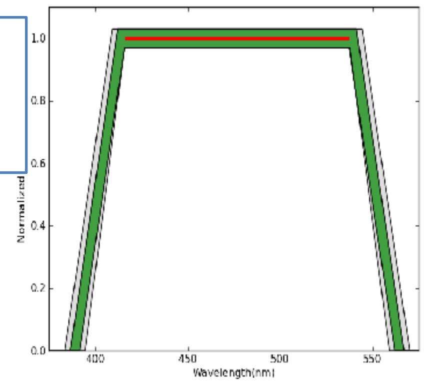
Description	Value	Unit	Name
The blue side zero response wavelength of the g-band upper envelope.	386.5	nm	g_upperBlue(0)
The blue side 103% response wavelength of the g-band upper envelope.	412.25	nm	g_upperBlue(1.03)
The red side 103% response wavelength of the g-band upper envelope.	541.25	nm	g_upperRed(1.03)
The red side zero response wavelength of the g-band upper envelope.	567.0	nm	g_upperRed(0)

Req. text references attribute names or Constraint Block

Constraint Blocks displayed as tables with each attribute as a row

LARGE SYNOPTIC SURVEY TELESCOPE

LSST Observatory System Specifications LSE-30 Latest Revision 1/27/2015



g-band not-to-exceed envelope

ID: OSS-REQ-0367 **Last Modified: 5/16/2014**

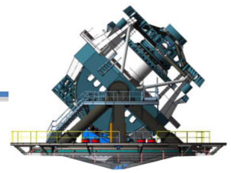
Specification: Over the wavelength range defined by the upper envelope - excluding the in-band range, 30% (by wavelength) of the area weighted average g-band filter response with may lie outside the nominal upper and lower envelope, but shall lie completely within the minimum and maximum envelopes defined below.

Discussion: Specific instances of non compliance to this specification will be evaluated by the project to assess acceptability.

g_minEnvelope

Description	Value	Unit	Name
The blue side zero response wavelength of the g-band minimum envelope.	394.5	nm	g_minBlue(0)
The blue side 97% response wavelength of the g-band minimum envelope.	415.75	nm	g_minBlue(0.97)
The red side 97% response wavelength of the g-band minimum envelope.	537.75	nm	g_minRed(0.97)
The red side zero response wavelength of the g-band minimum envelope.	559.0	nm	g_minRed(0)

Subsystem Level Milestones



Hardware Centric

Review, Verification, and Acceptance Milestones to be identified for each Component:
Requirements Review
Final Design Review
Procurement Review
Manufacturing Readiness Review
Verification Plan Review
Start of Verification Activities (i.e. Tests)
Subsystem Pre-shipment Review (if applicable)
Subsystem Acceptance Review
…..

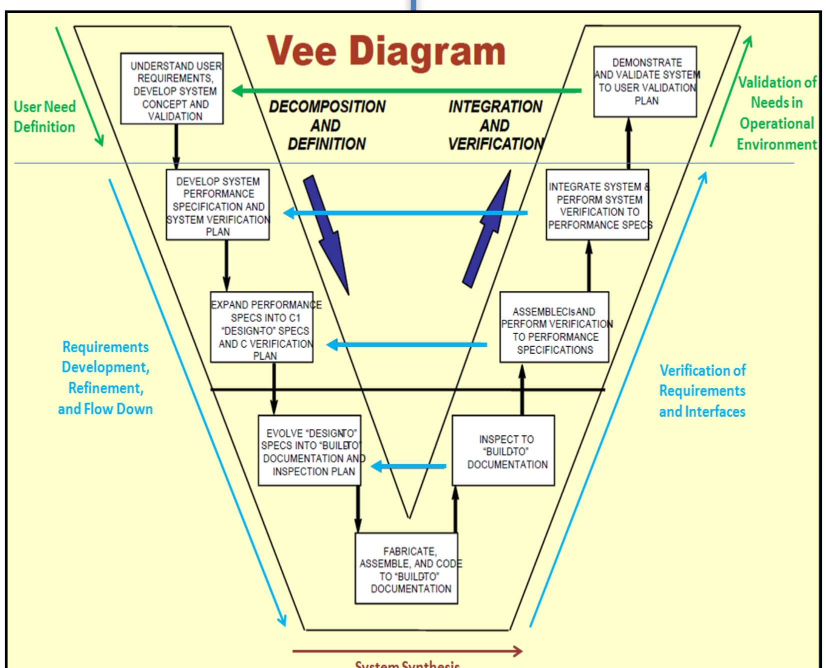
Software Centric

Review, Verification, and Acceptance Milestones to be identified for each Component:
Release Objectives Review
Verification Plan Review
Unit Test
Low Level Integration Test
End to End Test
Acceptance Test
Acceptance Test Review
…..

Verification vs. Validation

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

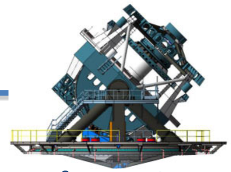
- **Validation**:
  - Provides objective evidence that the services provided by a system when in use in an operational environment comply with the stakeholders' needs.
  - "You built the right thing."



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.

Verification Methods



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)