System Engineering Workflow Use Cases

INCOSE Session Working Document

Table of Contents

[1. Context 4](#_Toc410632964)

[2. Life Cycle Workflow Use Cases for System Engineering 5](#_Toc410632965)

[2.1. Exploratory and Concept Stage Workflow Use Cases 6](#_Toc410632966)

[**2.1.1.** **Analyze Stakeholders Needs (Reviewed)** 6](#_Toc410632967)

[**2.1.2.** **Derive System Requirements *(Textual refinement)*** 9](#_Toc410632968)

[2.2. System Development Stage Workflow Use Cases 12](#_Toc410632969)

[**2.2.1.** **SE Domain Workflow Use Cases** 12](#_Toc410632970)

[***2.2.1.1.*** ***Derive Product Architecture – (Textual refinement)*** 14](#_Toc410632971)

[***2.2.1.2.*** ***Evaluate System Safety (Not refined)*** 16](#_Toc410632972)

[***2.2.1.3.*** ***Collaborate with Implementation Domain Team (Not Refined)*** 17](#_Toc410632973)

[2.3. Supporting Activity Behaviors 18](#_Toc410632974)

[**2.3.1.** **Measure a Change Impact (Not Reviewed)** 18](#_Toc410632975)

[**2.3.2.** **Conduct a Review – (Reviewed and Updated)** 19](#_Toc410632976)

[**2.3.2.1.** **Conduct a Review (Before Review)** 19](#_Toc410632977)

[**2.3.2.2.** **Conduct a Review (After Review)** 20](#_Toc410632978)

[**2.3.3.** **Import Reference Materiel into Model (Discussed with Ron Williams after the session)** 21](#_Toc410632979)

[**2.3.3.1.** **Before Changes** 21](#_Toc410632980)

[**2.3.3.2.** **After Changes** 21](#_Toc410632981)

[**2.3.4.** **Analyze Requirements** 22](#_Toc410632982)

[**2.3.4.1.** **Before Changes** 22](#_Toc410632983)

[**2.3.4.2.** **After Changes** 23](#_Toc410632984)

Table of Figures

[**Figure 1: Systems Engineering Workflow Use Case Context** 4](#_Toc410632985)

[**Figure 2: Exploratory and Concept Stage Workflow Use Cases** 6](#_Toc410632986)

[**Figure 3: SE Domain Workflow Use Cases** 12](#_Toc410632987)

[**Figure 4: Specialty Engineering** 13](#_Toc410632988)

[**Figure 5: Measure a Change Impact** 18](#_Toc410632989)

[**Figure 6: Conduct a Review** 19](#_Toc410632990)

[Figure 7 Conduct a Review AFTER REVIEW 20](#_Toc410632991)

[**Figure 8: Import Reference Materiel into Model** 21](#_Toc410632992)

[Figure 9 - Import Reference Material into Model AFTER REVIEW 21](#_Toc410632993)

[Figure 10 Analyze Requirements 22](#_Toc410632994)

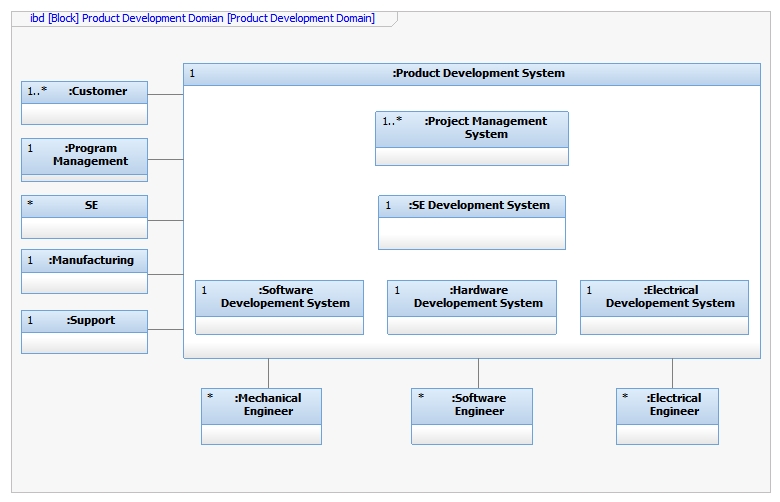
[Figure 11 Analyze Requirements AFTER REVIEW 23](#_Toc410632995)

1. **Context**

Notes:

Other source inputs that may be available, Requirements Working Group, Essential Unified Process, Essence standard within OMG

The diagram below defines a typical Product Domain structure. Within this domain is the SE Development System. Systems Engineering Development System (SEDS) is the context for the Systems Engineering Workflow Use Cases. It is the system used to provide an integrated environment of tools and capabilities required to perform Systems Engineering activities and tasks. This includes the environment to support system requirements flowdown, design, analysis, verification, validation activities. The SEDS provides interfaces to external domains such as the software, mechanical, electrical engineering domains and interfaces to manufacturing, support and product management domains.



**Figure 1: Systems Engineering Workflow Use Case Context**

1. **Life Cycle Workflow Use Cases for System Engineering**
   1. **Exploratory and Concept Stage Workflow Use Cases**

C:\Users\watsonjc\Documents\Work Folder\LM_ProfilesAndScripts_8\DocGen/figures/GUID ca74f3a3-84ae-4495-ab94-c9ffbd285ee3.emf

**Figure 2: Exploratory and Concept Stage Workflow Use Cases**

* + 1. **Analyze Stakeholders Needs (Reviewed)**

**Reviewed Text and activity – No changes, Sub-activities were also reviewed and enhanced.**

**Goal** - The goal of this workflow use case is to identify all stakeholders and better understand and capture their required needs expectations, goals, and objectives across the entire product life cycle.

**Primary Actor** - System Architect

**Secondary Actors** - Stakeholders

**Preconditions** -

**Activity -**

References: INCOSE Systems Engineering Handbook v. 3.2.2

INCOSE‐TP‐2003‐002‐03.2.2, October 2011, Section 4.1

1. Define Stakeholders and their Needs/Goals - In this task the primary actor identifies and captures the system-of-interest stakeholders from across all life cycle stages. The intent is to elicit and capture a set of stakeholder needs, expectations, goals, or objectives for a desired solution.
2. Define the system domain level model - This task is to evaluate and capture the existing domain structure with and without the system-of-interest included. In this task the definition of the appropriate domain entities and their relationship to other entities are captured. These definitions can include stakeholders, other systems, organizations or roles that participate with or influence the system-of-interest. Also captured are the primary interfaces for each.
3. Analyze mission level use cases - Identify the mission level use cases where the system-of-interest will be expected to participate. This task includes capturing the behavior of the system-of-interest and other participating entities to fully understand the needs of the system-of-interest, what constraints will be imposed because of this surrounding environment and the key system-of-interest interfaces.
4. Analyze system threats - Identify and capture any additional use cases or scenarios that may occur during the mission level use cases that can pose an external threat or unwanted system behavior.
5. Define Effective Measures - Elicit from the stakeholders a set of measurable properties and a means of demonstrating how the overall system goals and objectives are met.
6. Derive a set of stakeholder requirements relative to the system-of-interest needs analysis.
7. Capture any unknowns, risks and assumptions. Derive a method to manage these entities through the product life cycle.
8. Review results with internal stakeholders including appropriate internal development, manufacturing and management teams.
9. Review results with the customer and other appropriate external stakeholders.

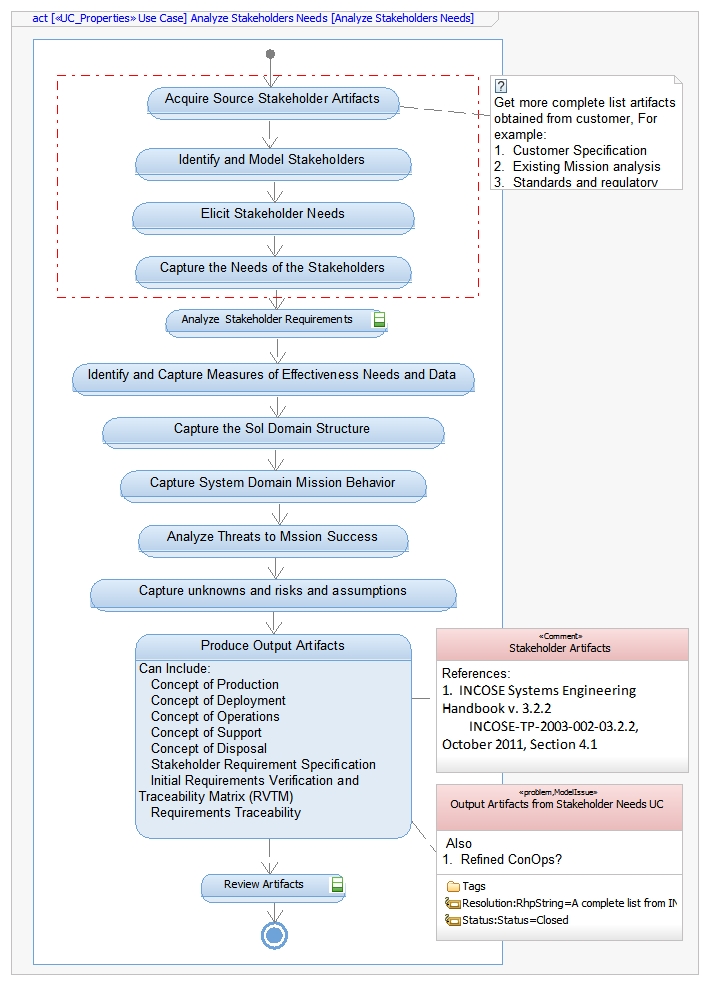
**Post Conditions** -

1. A domain level model is available including all stakeholders, appropriate mission level use cases, the required needs of the stakeholders, the system-of-interest’s boundaries and its key interfaces to other entities.

**References**

 ISO/IEC 15288-2008 - 6.4.1 Stakeholder Requirements Definition Process

 SEBok - Concept Definition



* + 1. **Derive System Requirements *(Textual refinement)***

**Goal** - The goal of this workflow use case is to derive a set of system level requirements for the system-of-interest based on the all stakeholder’s needs requirements.

**Primary Actor** - System Architect

**Secondary Actors** -

**Preconditions** - “Analyze customer needs” use case is near completion.

**Activity -**

The following is a set of tasks that are typically captured within a SysML model.

1. Overall guidance items:
   1. These tasks are not necessarily performed in this order shown. Many are performed simultaneously depending on the needs. Also each task may not be run to full completion before advancing another parallel task.
   2. It may be determined that some of the tasks or parts of the tasks are not necessary for a particular effort. What tasks are done and to what degree should be determined during the project planning stages.
   3. As this activity progresses it may be determined that it would be beneficial to initiate another development use cases before completing this use case. This is typically done to provide more insight or to reduce project risks.
2. Identify a reference to the appropriate external documentation and reference (or import) the mission behavior, domain structure and stakeholder requirements from the domain level model.
3. Feedback to the customer concerning changes to customer needs or a request for clarification of a need
4. Define a system-of-interest context diagram with the goal of discovering the system interfaces and the system boundaries. This includes identifying external systems, organizations and human factors that may interact or influence the system directly or indirectly.
   1. Identify all system-of-interest interfaces
   2. Identify system-of-interest actors
5. Domain level Mission use cases from the domain model may be referenced or enhanced to aid in defining and refining these interactions and the goals of the system. Refine SOI behavior.
6. Define a system-of-interest states and modes state machine.
7. From the domain level requirements, analyzes the non-behavioral stakeholder’s needs and derive a set of system level requirements for the system-of-interest.
8. Capture and demonstrate traceability of these derived system level requirements to the domain level requirements.
9. Analyze system-of-interest behavioral needs -
   1. By examining the domain level mission use cases derive and capture a set of system level uses cases, their primary actor and the goal of the primary actor.
   2. For each use case analyze and capture the blackbox behavior of the system required to achieve each use case goal.
10. By examining the system level use cases derive and analyze a set of system level requirements.
    1. These requirements should include functional requirements, external interface requirements, safety requirements, security requirements, environmental requirements, states and modes, design constraints, construction constraints, etc.
    2. Capture and demonstrate traceability of the derived system level requirements to the domain level requirements and from the system level use cases to the mission level use cases.
11. Derive and analyze requirements such as non-functional requirements
    1. These requirements should include external interface requirements, safety requirements, security requirements, environmental requirements, states and modes, design constraints, construction constraints, etc.
12. System validation and verification -
    1. Derive MOPs
    2. Derive and analyze a set of test cases to validate the system meets the customer’s expectations and that all the requirements have been verified.
    3. Derive the necessary test system architecture, test components and test requirements required to verify and validate the system
    4. Demonstrate and capture traceability from the test cases to the system level requirements.
    5. Create a preliminary Verification Plan with define acceptance criteria
    6. Create and capture a Requirements Verification Matrix
13. Capture any unknowns, risks and assumptions. Derive a method to manage these entities through the product life cycle.
14. Review results with internal stakeholders including appropriate internal development, manufacturing and management teams.
15. Review results with the customer and other appropriate external stakeholders.

**Post Conditions** - Artifacts created include a system-of-interest context diagram, a set of system level use cases, system level requirements, verification and validation plan, modeling and simulation analysis plan, requirement verification matrix.

A System Requirement Specification or equivalent will produced including many of the captured artifacts.

* 1. **System Development Stage Workflow Use Cases**
     1. **SE Domain Workflow Use Cases**

C:\Users\watsonjc\Documents\Work Folder\LM_ProfilesAndScripts_8\DocGen/figures/GUID d76fa43d-2901-4b59-87b4-43e07dcec73a.emf

**Figure 3: SE Domain Workflow Use Cases**

C:\Users\watsonjc\Documents\Work Folder\LM_ProfilesAndScripts_8\DocGen/figures/GUID 31336da8-8544-453c-998b-53dbde80990e.emf

**Figure 4: Specialty Engineering**

* + - 1. ***Derive Product Architecture – (Textual refinement)***

**Goal -** The goal of this workflow use case is to evaluate the System Requirements and from them derive the most appropriate architecture to satisfy the customer needs.

**Primary Actor -** System Architect

**Secondary Actors -**

**Preconditions -** The Exploratory and Concept Stage use cases have been executed and a reasonably stable set of system level requirements are available.

**Activity -**The following is a set of tasks that are typically captured within a SysML model.

1. Overall guidance items:
   1. These tasks are not necessarily performed in this order shown. Many are performed simultaneously depending on the needs. Also each task may not be run to full completion before advancing another parallel task.
   2. It may be determined that some of the tasks or parts of the tasks are not necessary for a particular effort. What tasks are done and to what degree should be determined during the project planning stages.
   3. As this activity progresses it may be determined that it would be beneficial to initiate another development use cases before completing this use case. This is typically done to provide more insight or to reduce project risks.
2. Reference the set of artifacts produced from the Exploratory and Conceptual Stages so that it is visible, searchable and analyzed.
   1. If Model-based System Engineering was performed during these early stages this work should be a continuation and refinement of that work.
   2. This information can include use cases, requirements, structural information, MOEs, KPPs, data, etc.
3. Define logical architecture - Derive a set of candidate subsystems and or components and iteratively validate these selections by analyzing the internal system behavior and by analytical means.
   1. Show the structural relationships of the components on a Block Definition Diagram.
   2. Show the interconnectivity of these components on an Internal Block Diagram.
4. Analyze the internal system behavior - By using the blackbox behavior of the system level uses cases, create and capture equivalent white box behaviors showing subsystem/component level collaboration required for attaining the same use case goals.
5. From the allocated behavior shown on the behavioral diagrams, derive a set of subsystem/component level requirements. These derived requirements can be of any type potentially, but they are typically functional, interface, and performance type requirements.
6. Show traceability from the subsystem/component level requirements to the system level requirements.
7. As each use case progresses conduct the appropriate reviews solicit feedback. Adjudicate all issues and update the reviewed artifacts as needed.
8. Perform a configuration management baseline for the appropriate artifacts at the completion of each review.
9. Integrate other domain needs in to the solutions as needed including safety, security, RMA, SWaP, human constraints, etc.
10. Synthesizing the logical architecture to one or more physical solutions
11. Select a final solution by performing trade studies with candidate solutions
12. Derive component requirement specifications and collaborate with the implementation teams to finalize the component specifications. The form of these requirements can be a traditional document or an electronic version, such as an html document, or a limited view of the existing System level model.
13. Initiate the use case for deriving the Test and Integration plans

**Post Conditions -**

**Output -** Typical deliverables for this effort include Subsystem Requirement Specifications, Software Architecture Document, H/W and/or SW Component Specifications, Interface requirement specifications, Interface design specifications, etc.

* + - 1. ***Evaluate System Safety (Not refined)***

**Goal** - The goal of this workflow use case is to evaluate the system for safety related hazards and derive a plan to mitigate these risks.

**Primary Actor** -

**Secondary Actors** -

**Preconditions** -

**Activity** -

1. Identify Safety Hazards undesired events and their causes
2. Identify applicable safety standards
3. Conduct analysis to determine the severity level, the probability of occurrence and assess the level of risk.
4. Determine if the analysis results are acceptable for use.
   1. If it is acceptable, capture these results and show tractability to identified Hazard.
   2. If not, determine best corrective solution to eliminate or minimize the level of risk. This could be by design and/or by procedure/process
   3. Update Model and other information base
      1. Derive Safety related requirements that address Hazards.
      2. Show how these requirements are satisfied
      3. Show traceability from Hazards to Risk mitigation requirements to system elements satisfying those requirements.
5. Verify solution to determine if Hazard has been appropriately addressed.

**Post Conditions -**

* + - 1. ***Collaborate with Implementation Domain Team (Not Refined)***

**Goal** - Goal of this workflow use case is provide an automated capability to effectively share information between Systems Engineering and the component implementation engineering domains, such as the software, electrical, and mechanical domains.

**Primary Actor** - Systems Engineer

**Secondary Actors** - Component Engineers

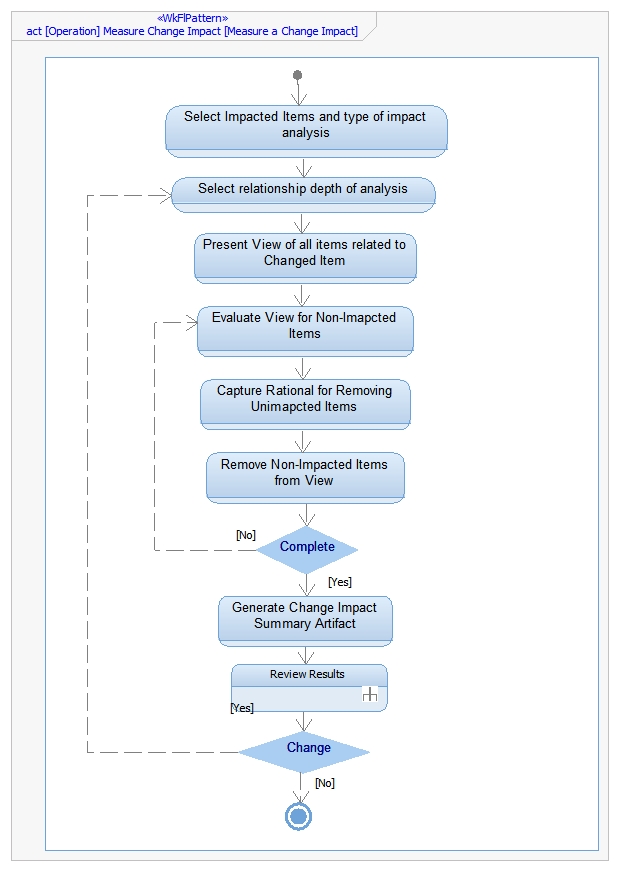
**Preconditions** -

**Activity -**

1. Interact with implementation domain to derive component specification
2. Baseline component specification with consideration to other peer and interactive components
3. Attend design reviews as needed
4. Address issues, concerns, ideas from component engineering

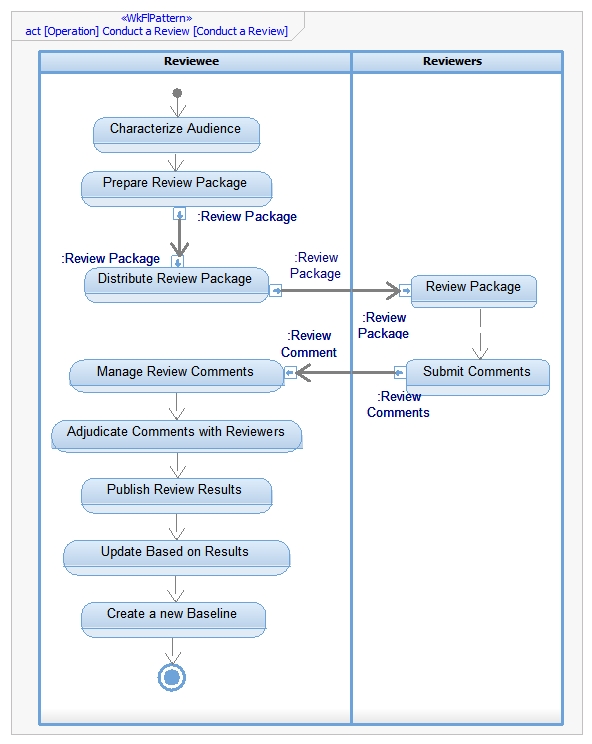
**Post Conditions** -

* 1. **Supporting Activity Behaviors**
     1. **Measure a Change Impact (Not Reviewed)**



**Figure 5: Measure a Change Impact**

* + 1. **Conduct a Review – (Reviewed and Updated)**
       1. **Conduct a Review (Before Review)**



**Figure 6: Conduct a Review**

* + - 1. **Conduct a Review (After Review)**

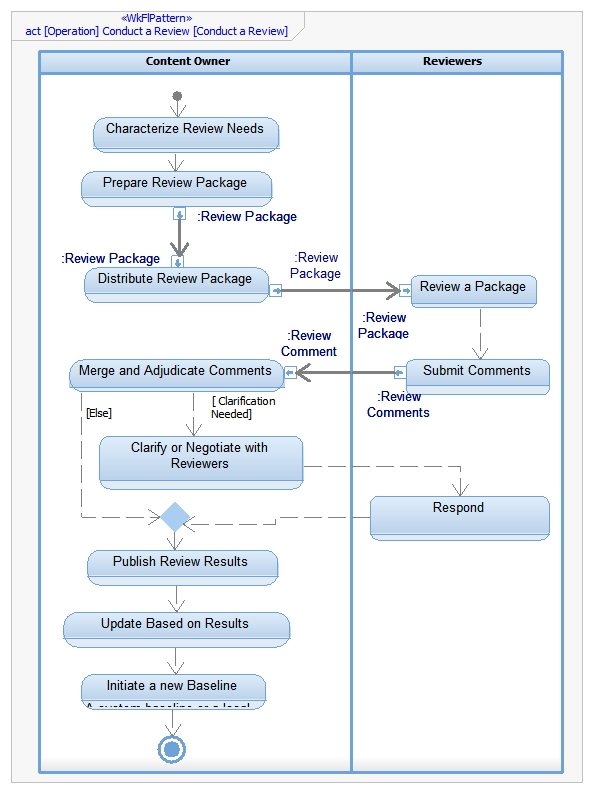
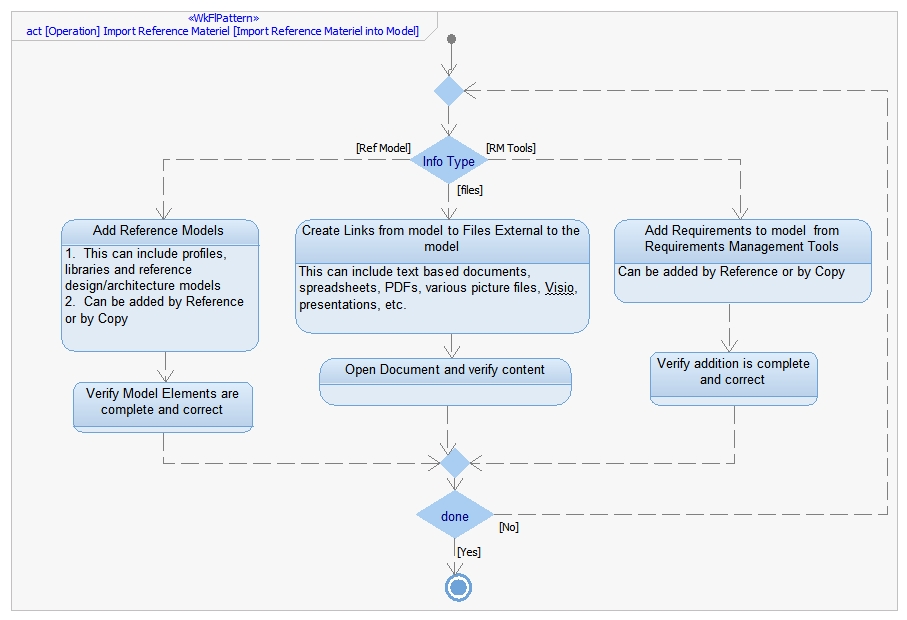


Figure Conduct a Review AFTER REVIEW

* + 1. **Import Reference Materiel into Model (Discussed with Ron Williamson after the session)**
       1. **Before Changes**



**Figure 8: Import Reference Materiel into Model**

* + - 1. **After Changes**

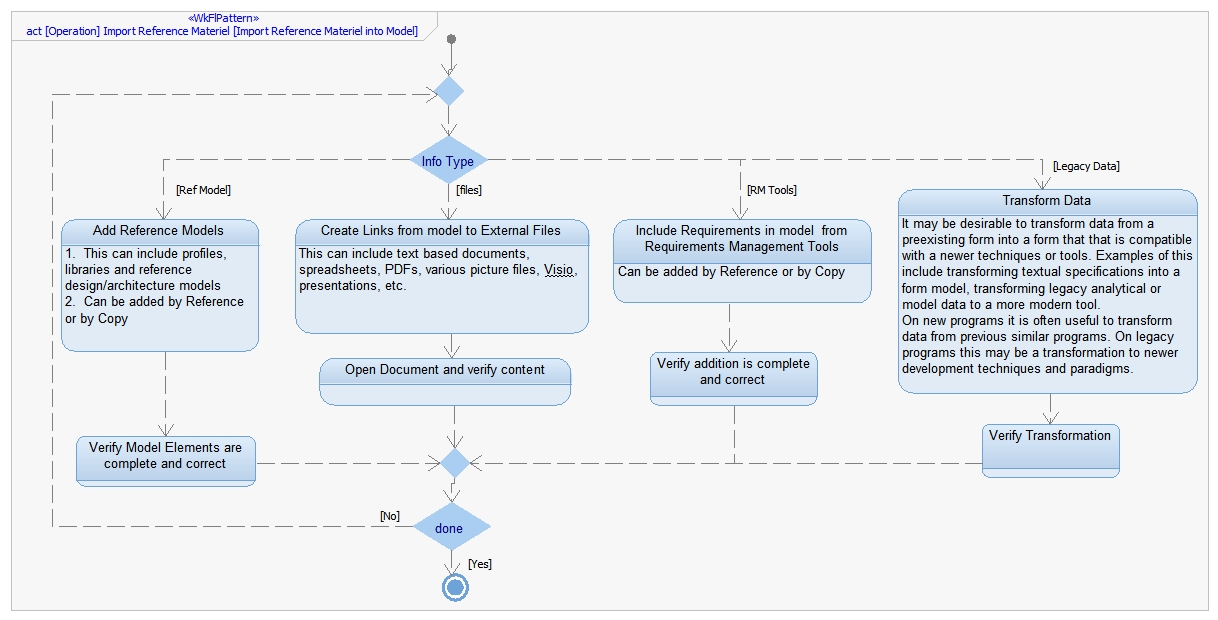


Figure - Import Reference Material into Model AFTER REVIEW

* + 1. **Analyze Requirements**
       1. **Before Changes**

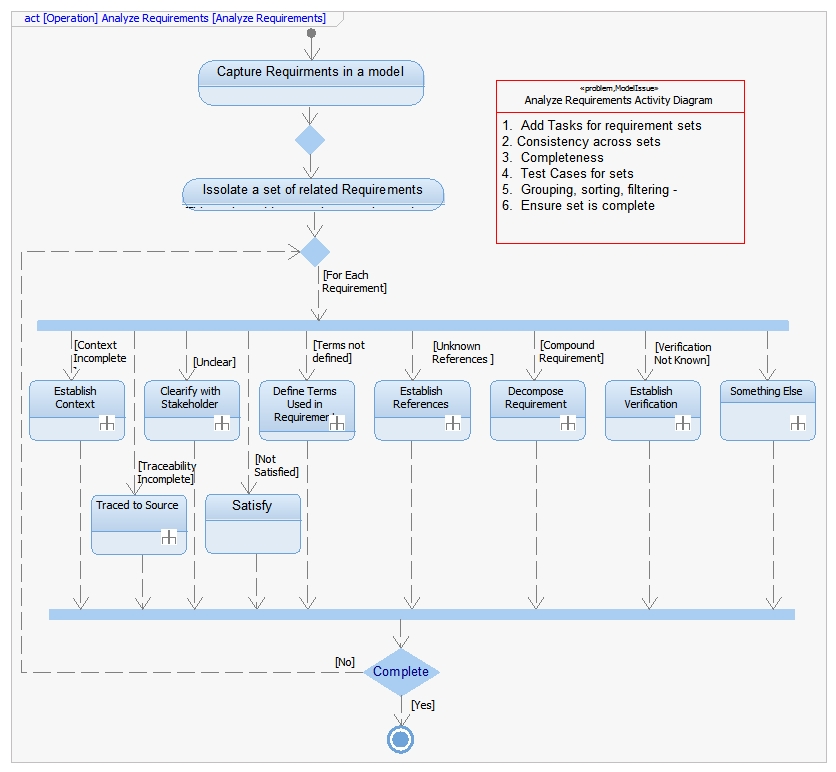


Figure Analyze Requirements

* + - 1. **After Changes**

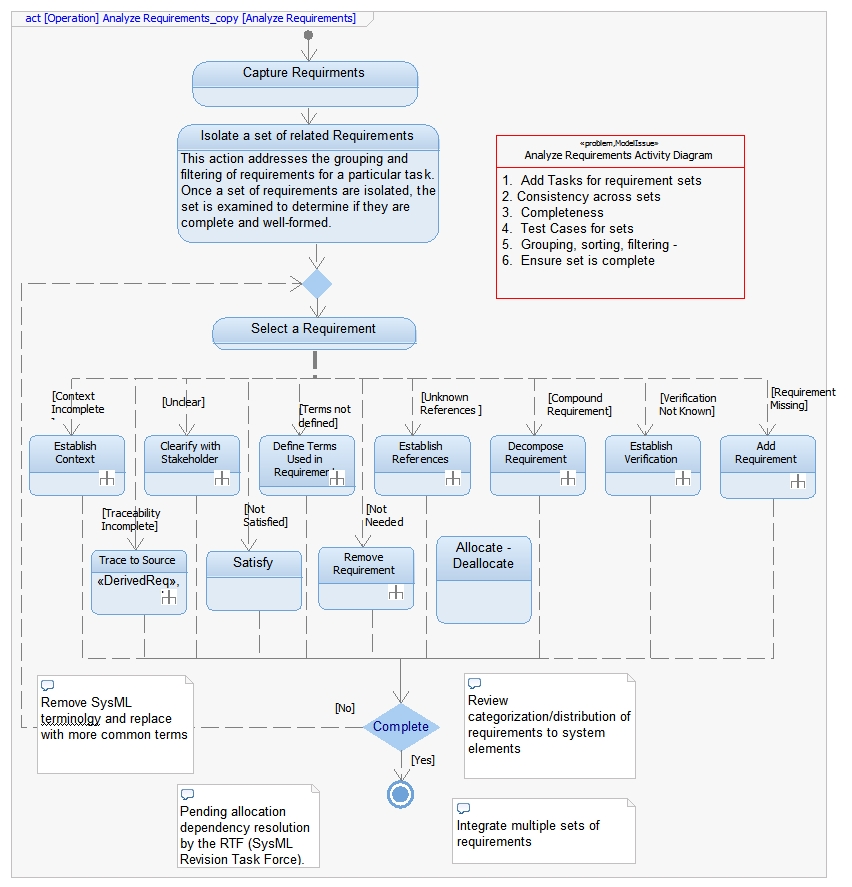


Figure Analyze Requirements AFTER REVIEW