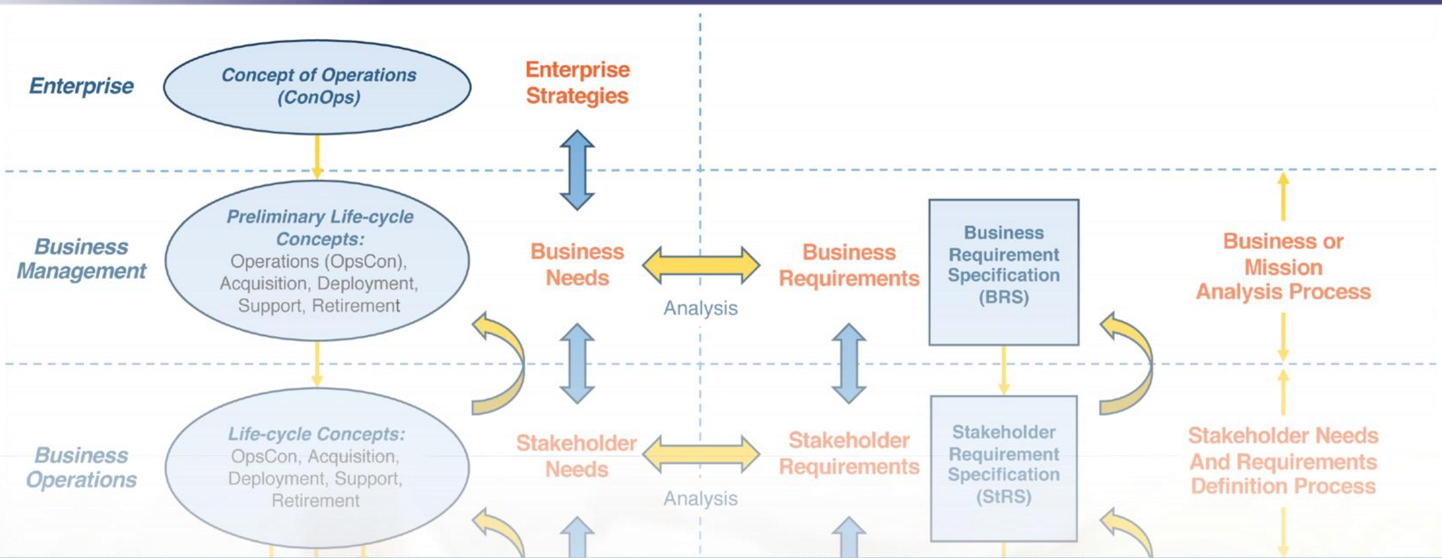



# Guide to Needs and Requirements

# May 2022

## COPYRIGHT INFORMATION

# Guide to Needs and Requirements

Document No.: INCOSE-TP-2021-003-01
Version/Revision: 1.0
Date: May 2022

Prepared by:

This document has been prepared and produced by a volunteer group of contributors within the Requirements Working Group (RWG) of the International Council on Systems Engineering (INCOSE).

## Authors

The principal authors of this Guide are:

Tami Katz, Ball Aerospace, USA
Kevin Orr, Eaton Corporation, USA
Lou Wheatcraft, Wheatland Consulting, LLC, USA

## Contributors and Reviewers

Those who made a significant contribution to the generation of this Guide are:

Ronald S. Carson, PhD, Seattle Pacific University, USA
Michael Ryan, Capability Associates Pty Ltd, Canberra, Australia
Celeste Drewien, Sandia National Labs, USA
Donald McNally, Woodward, USA
Susan Ronning, EBE Consulting, USA
Rick Zinni, Harris Corporation, USA

## Supporting Contributors

The following provided support in establishing example figures for this guide:

Saulius Pavalkis, Dassault Systemes, USA
Cary Bryczek, Jama Software, USA

# REVISION HISTORY

| Revision | Revision Date | Change Description & Rationale |
|---|---|---|
| 1.0 | May 2022 | Initial Release |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

# TABLE OF CONTENTS

# LIST OF FIGURES

# 1 INTRODUCTION

## 1.1 Purpose and Scope

Needs and requirements form the backbone of the systems engineering information model of the System of Interest (SOI) being developed. Needs and Requirements Definition and Management (NRDM) addresses how the stakeholder needs and requirements are formed and managed throughout the project lifecycle. This Guide to Needs and Requirements (GtNR) addresses the current lack of practical guidance regarding these activities, providing **application guidance** for the concepts and activities presented in the INCOSE Needs and Requirements Manual (NRM)[1].

The goals and objectives of this guide are to:

- Provide a practical, cross domain guidance with examples, to enable organizations to integrate the concepts contained in the NRM into their organization's plans, processes, and work instructions.
- Provide support for practitioners to ensure effective NRDM over the project lifecycle.
- Provide organizations guidance that can be used to successfully implement NRDM processes within their Systems Engineering (SE) activities at the proper level that best fits their culture, workforce, processes, and domain.
- Present methods to implement a data-centric approach of NRDM across the system lifecycle.
- Help practitioners understand why effective lifecycle concept and needs definition is necessary prior to successfully performing the requirements definition activities.

Note: This guide assumes that the project has been formed and authorized and does not provide insight to business management and business operations activities related to defining the problem/opportunity or defining the Mission, Goals, Objectives, (MGO) for the SOI. Refer to the INCOSE Systems Engineering Handbook [2] , Business or Mission Analysis process for insight into the definition of the MGOs for the SOI. However, given SE is iterative and recursive, the project team will need to define MGOs for the subsystems and system elements that are part of the integrated SOI physical architecture. In this context, the needs defined via the activities in this guide will result in the MGOs and measures being achieved for the SOI and each subsystem and system element that is part of its architecture. See the NRM for guidance on defining MGO.

## 1.2 Audience

This guide is intended for those who:

- Perform NRDM activities throughout the SOI definition lifecycle.

- Read, implement, verify that the integrated physical SOI meets the requirements.
- Validate the SOI meets the needs in its actual operational environment by its intended users.
- Validate that the SOI does not enable/allow unintended users to negatively impact the intended use of the system.

This guide is addressed to practitioners of all levels of experience. Someone new to this role should find specific guidance provided to be useful, and those more experienced should be able to find new insights concerning the NRDM across all lifecycle stage activities. For the more experienced practitioners of SE, this guide exposes them to NRDM in a different perspective than they may have been exposed in the past.

## 1.3 Approach

As shown in Figure 1, this guide is in alignment with, and complements, the INCOSE Needs and Requirements Manual (NRM) in support of the INCOSE Systems Engineering Handbook. It is recommended the reader has familiarity with the underlining concepts within the NRM as well as the related guides: Guide to Writing Requirements (GtWR)[3] and Guide to Verification and Validation (GtVV)[4]. The success of the NDRM activities presented are highly dependent upon the quality of the need and requirement expressions and the sets of needs and requirements as defined in the GtWR and support successful implementation of a system verification and validation effort described in the GtVV.

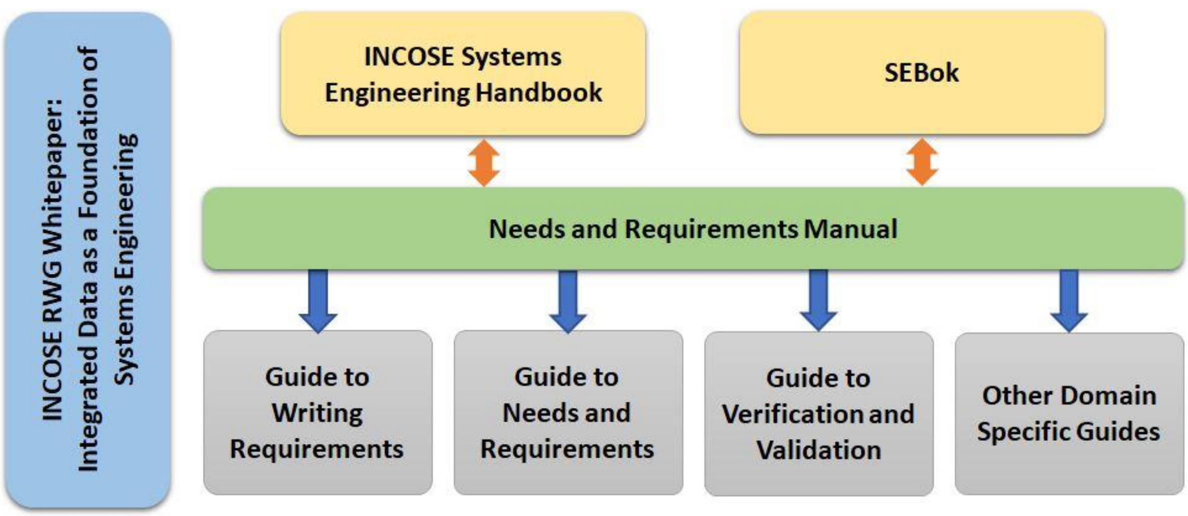



**Figure 1. Relationships between RWG and other INCOSE Products.**

Figure 2 shows the summary outline contents of the RWG guides, and how *this guide* works in conjunction with the other RWG guides for a comprehensive approach to application of the concepts within the NRM.

This guide assumes the reader/user is able to obtain access to the NRM and takes the approach to summarize and expand on the concepts elaborated within that manual. For each topic in this Guide, Input/Output (IPO) Diagrams are extracted from the NRM to provide awareness of the inputs, activities, enablers and outputs. Refer back to the NRM for details on these IPO Diagrams as needed. This guide expands on the activities associated with these activities after each IPO Diagram.

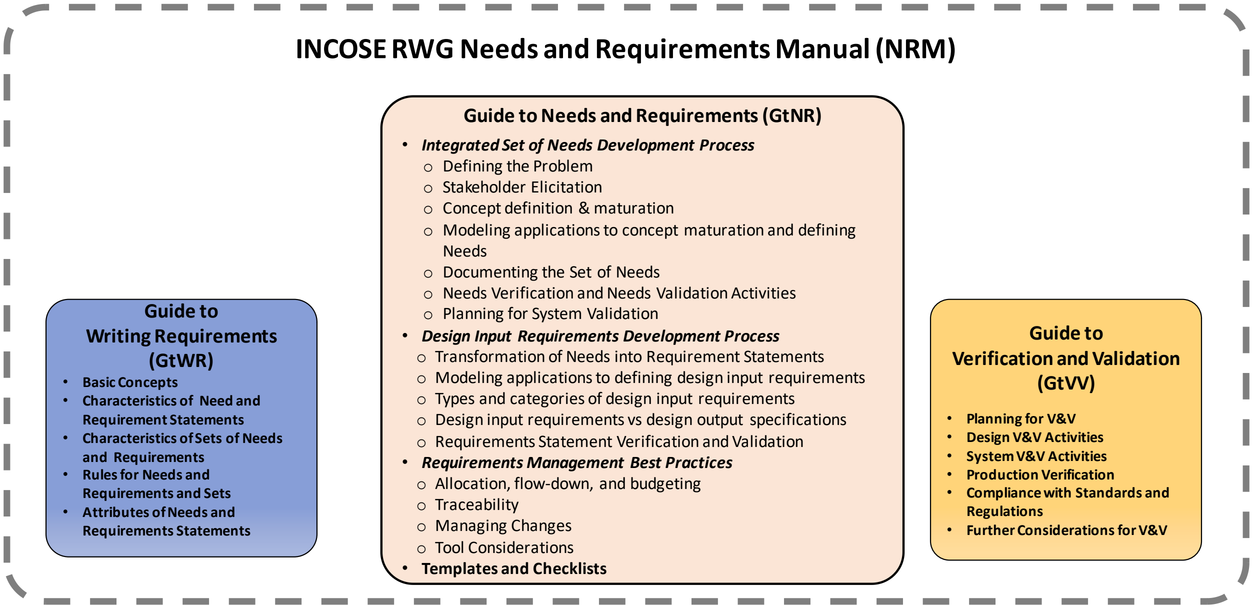



**Figure 2. Detailed Relationships between RWG Products.**

Figure 3 shows the context of this guide, where the primary focus is on definition and management of a system's integrated set of needs and its design-input requirements.

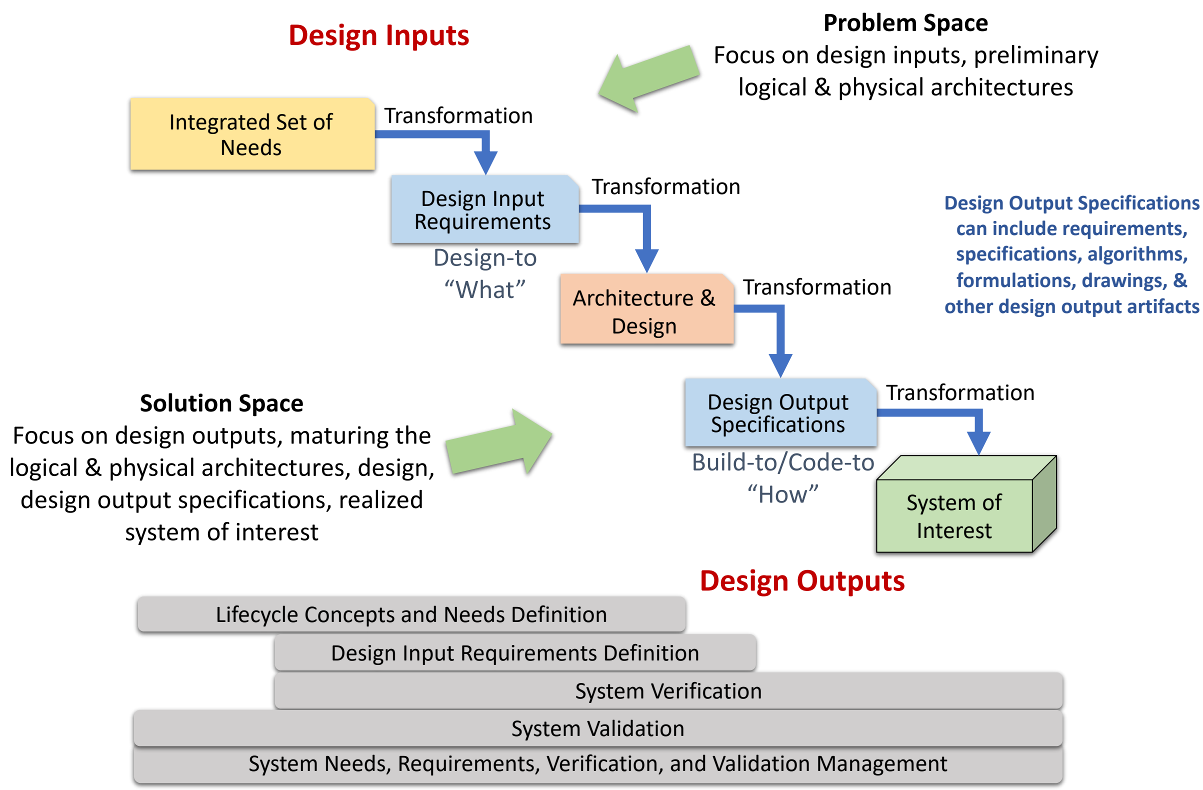



**Figure 3. Focus of Guide Content: Needs and Design input Requirements Development and Management (from NRM).**

## 1.4 Terminology and Definitions

The end item being developed can be referred to as either a system or a product (note that "product" can imply the integrated system). The integrated system is made up of subsystems and lower-level system elements (assemblies, sub-assemblies, components) that are defined within the system architecture. For this guide, System of Interest (SOI) will be used hereafter to refer to the system, subsystem, or system element that is to be developed, verified, validated, and delivered to either an internal or external customer. The NRDM activities discussed in this guide, apply to any SOI, no matter the level of the physical architecture.

Many organizations use similar terminology with differing meanings. This guide follows the terminology and definitions used in the NRM, however it is recognized that the following definitions apply for certain industries:

- "Requirements documents" and "requirements specifications" are often used interchangeably. To avoid ambiguity, this guide refers to the set of requirements that are inputs to the architecture and design processes as “design input requirements” and the outputs of the design process are referred to as “design output specifications” as shown in Figure 3. If the word “requirements” is used by itself, the meaning is “design input requirements”. If the word “specifications” is used by itself, the meaning is “design output specifications”.

- Stakeholder "needs" and stakeholder "requirements" are often viewed as interchangeable and as the single source of the requirements. In reality, stakeholder needs are separated and distinct from stakeholder requirements. Each represents a different perspective. The stakeholder needs represent an external view of the SOI with a focus on what the stakeholders need from the SOI to address their real-world expectations. Stakeholder requirements are stakeholder-owned system requirements (design input requirements) that communicate the stakeholders' perspectives of what the system must do to meet their needs. In this guide, there is an assumption there are often multiple sets of stakeholder needs and stakeholder-owned requirements that are elicited as one of several inputs into the lifecycle concept and needs definition activities. The output of these activities is an integrated set of needs that are then transformed into a set of design input requirements for the SOI.
- "Need verification" and "requirement verification", per this guide, is an assessment of need statements and requirement statements quality in order to assess correctness to conventions (such as those presented in the GtWR). In some industries, like those that use ARP 4754 (specific to the aviation industry), this may be called "requirement validation", however that term is reserved for the concept described below.
- "Need validation", per this guide, is an assessment of how well the needs represent the stakeholders' real-world expectations for the SOI. "Requirement Validation", per this guide, is an assessment of how well the requirements meet the intent of the integrated set of needs from which they were transformed as well as the allocated parent requirements from the level above, demonstrating completeness and correctness.

Short definitions are provided below for reference; refer to the INCOSE Needs and Requirements Manual (NRM) for detailed definitions/descriptions.

**Attribute**: Additional information associated with an entity which is used to aid in its definition and management.

**Change Management:** The process by which the proposed requirement changes are subjected to a defined impact assessment and are reviewed and approved, using requirements tracing and version management.

**Entity:** A single item to which a concept, need, or requirement applies: an organization, business unit, *project, supplier,* service, SOI (system, subsystem, system element), product, process, or *stakeholder class (user, operator, tester, maintainer, etc.)*.

**Lifecycle Concept:** A *textual* or graphic representation that concisely expresses how an entity can satisfy the problem, threat, or opportunity it was defined to address within specified constraints with acceptable risk that provides a business capability in terms of people, process, and products. Lifecycle concepts communicate ways in which the organization (and entities within an organization) expect to manage, acquire, define, develop, build/code, integrate, verify, validate, transition, install, operate, support, maintain, and retire the system that provides a business capability in terms of people, process, and products.

**Needs:** Formal textual statements of expectations for an entity stated in a structured, natural language from the perspective of what the stakeholders need a SOI to do, communicated at a level of abstraction appropriate to the level at which the entity exists.

**Needs Definition:** Process of eliciting stakeholder needs and requirements, identifying drivers and constraints, identifying risks, developing and maturing lifecycle concepts, deriving an integrated sets of needs, and baselining those needs.

**Need Expression**: Statement that includes a need statement and a set of associated attributes.

**Need Set:** A structured set of agreed-to need expressions for the entity (enterprise/business unit/system/subsystem/system element/process) and its external interfaces.

**Need Statement**: The result of a formal transformation of one or more lifecycle concepts into an agreed-to expectation for an entity to perform some function or possess some quality within specified constraints with acceptable risk.

**Needs and Requirements Management:** Process area focus is managing the needs and requirements for all parts of the system architecture across the lifecycle, which includes management of the sets of integrated SOI needs and requirements, managing the flow down (allocation) and budgeting of requirements from one level of the system architecture to subsystems and system elements at the next level, validating traceability, managing the approval and configuration management (CM) of the sets of needs and sets of requirements, and managing change.

**Requirements:** Formal textual statements that communicate in a structured, natural language what an entity must do to realize the intent of the needs from which they were *transformed.* For this guide "requirements" consist of the design input requirements shown on Figure 3.

**Requirement Definition:** Process area focus is on transforming the baselined integrated set of needs into a set of requirements expressed as “shall” statements that can be used as inputs for defining the architecture and design solution. The resulting set of requirements communicate what an entity must do to realize the intent of the integrated set needs and parent requirements from which they were *transformed.*

**Requirement Expression**: Statement that includes a requirement statement and a set of associated attributes.

**Requirement Set:** A structured set of agreed-to requirement expressions for the entity (enterprise/business unit/system/subsystem/system element/process) and its external interfaces.

**Requirement Statement:** The result of a formal transformation of one or more needs or parent requirements into an agreed-to obligation for an entity to perform some function or possess some quality within specified constraints with acceptable risk.

**Parent Requirements:** Requirements defined at the previous level (organization or system architecture) that have been allocated to the SOI for implementation. The parent requirements are inputs into the *Lifecycle Concept and Needs Definition* activities.

## 1.5 Roles and Responsibilities

The following types of roles in Table 1 will be involved within the activities of described in this guide. The specific project roles may vary on different projects, but the concept of responsible, accountable, consulted, and informed (RACI) are provided to address interactions between the various functions during the implementation of a project.

**Table 1. NDRM Roles and Responsibility RACI Matrix.**

| Project Role | Lifecycle Concepts & Needs Definition | Needs Management | Needs Verification | Needs Validation | Requirements Definition | Requirement Management | Requirement Verification | Requirement Validation |
|---|---|---|---|---|---|---|---|---|
| Customer | C | I | I | C | C | I | I | C |
| Project Manager / Team Lead | A | A | A | A | C | C | C | C |
| Requirements / Systems Engineer | R | R | R | R | A, R | A, R | A, R | A, R |
| Subject Matter Engineers | C | I | I | I | C | I | I | I |
| Verification Engineer or Qualification Engineer | C | | | I | C | | | I |
| Test Engineer | I | | | | C | | | |
| Quality Engineer | I | | | | C | | | |
| Project Management Professional, or Project Controller | I | | | | I | | | |
| Program Risk Manager | I | | | I | C | | | I |
| Verification and Validation Approval Authority | C | I | I | C | C | I | I | C |

The definitions used in Table 1 are as follows:

- Responsible (R): This team member does the work to complete the task.

- Accountable (A): This is the approving authority that often delegates the task and is the last to review the task or deliverable before it is deemed complete. On some tasks, the Responsible party may also serve as the Accountable one.
- Consulted (C): Consulted parties are people who provide input based on either how it will impact their future project work or their domain of expertise on the task.
- Informed (I): These team members need to be informed concerning project progress, rather than involved in the details of every task.

## 1.6 Guide Organization

This document is organized as follows:

- Section 1.0 introduces the guide, discusses key concepts, and key definitions of key terms used throughout the guide.
- Section 2.0 provides Integrated Set of Needs definition, verification and validation guidance.
- Section 3.0 provides Requirements definition, verification and validation guidance.
- Section 4.0 presents Needs and Requirements Management processes and best practices.
- Appendix A lists acronyms and abbreviations used in this document
- Appendix B provides a suggested outline for a Needs and Requirements Definition and Management Plan.
- Appendix C provides a checklist of questions for Stakeholder Needs Elicitation.
- Appendix D provides Needs and Requirements Verification and Validation checklists.
- Appendix E provides a list of references used in this guide.
- Appendix F provides a comment form for suggested inputs.

# 2 NEEDS DEFINITION, VERIFICATION, AND VALIDATION

This section expands on the NRM activities associated with defining the integrated set of needs, ensuring they are appropriately stated and correctly reflect the stakeholder expectations. This section provides practical guidance in implementing these activities; the reader is encouraged to refer to the NRM sections 4 and 5 for the underlying concepts and rationale behind these activities.

## 2.1 Lifecycle Concept and Needs Definition Activities

The lifecycle concept and needs definition activities capture the source(s) of the concern, problem, opportunity, the MGOs, and measures that drive understanding of the characteristics for acceptability or desirability of a solution and explores the conditions in which the SOI must operate. Lifecycle concepts are identified, analyzed, and matured from which an integrated set of needs is captured, baselined, and transformed into the requirements for the SOI.

Because the requirements are the result of this transformation, identifying and analyzing needs early in the SOI development lifecycle helps ensure the resulting set of requirements are complete, correct, consistent, and feasible. Managing the integrated set of needs helps ensure the realized SOI will meet stakeholder real-world expectations on schedule and budget with the expected quality. Figure 4 provides the inputs, outputs, activities, and enablers for the lifecycle concepts and needs definition activities from the NRM.

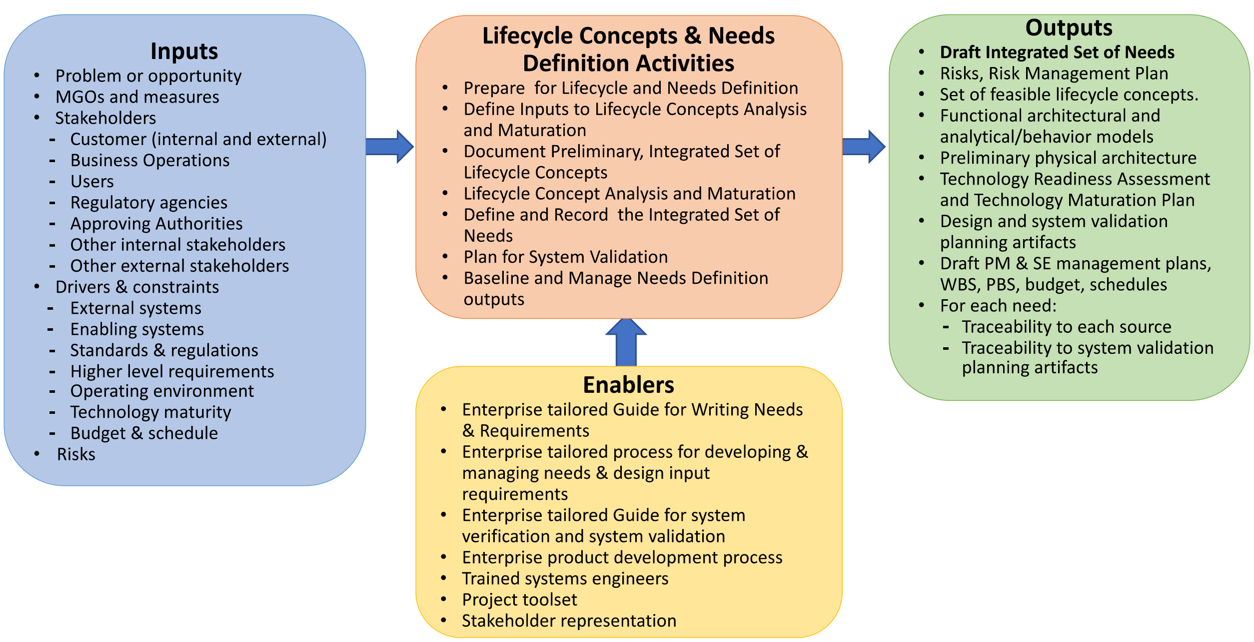



**Figure 4. Lifecycle Concept and Needs Definition IPO Diagram (from NRM).**

The steps of the lifecycle concepts and need definition activities are outlined below, showing methodologies of approach and options to consider.

### 2.1.1 Considerations for SOI Developed for Consumers Compared to a Customer

For projects being developed for a consumer market, the analysis of consumer needs and expectations are critical as they lead to the requirements for the product to be developed and delivered. Failure to understand the needs and expectations of the consumer and their operational environment can lead to a product that fails in the marketplace. In addition, most consumer products are heavily regulated for safety and security. Failure to address the applicable standards and regulations can result in a product that is not approved for its intended use by the consumer market.

For a SOI being developed in response to an acquisition contract, the customer often specifies their requirements in the contract through a statement of work (SOW) and a requirements document or in a supplier agreement, which are too often viewed as a complete set of requirements. As discussed in the NRM, however, this set of requirements may not fully reflect the intent of the set of needs and expectations for the SOI. The customer transformation of their needs (or their customer's needs) into SOI requirements may have issues leading to a defective set of requirements that are incomplete, incorrect, inconsistent, and of questionable feasibility. Examples of include:

- Failure to define the context of the SOI completely and correctly and align the requirements with the MGOs, measures, drivers and constraints, risks, use cases, operational scenarios, lifecycle concepts, operational environment, and interactions (physical and functional) with other subsystems and system elements or external systems. This often results in incorrect, incomplete, inconsistent set of needs and requirements. While the supplier may be able to produce a product that meets the customer-owned set of requirements, because of the these common defects, the product may fail system validation against the customer expectations even if not explicitly addressed in the integrated set of needs and resulting requirements.

- Failure to do to the analysis needed to identify critical technologies and assess their maturity and feasibility. As a result, performance requirements in the customer set may not be feasible.

- Failure of the customer to explicitly state their quality “-ilities" and compliance expectations in their integrated set of needs from which the contracted set of requirements were transformed. These omissions can result in an inability to produce, test, maintain, or achieve regulatory approval of the product being developed.

The steps in the subsequent sections highlight how additional analysis can enable the supplier to achieve overall validation and acceptance of the SOI they are developing in support of a customer contract.

### 2.1.2 Enabling Processes and Tools

Organizations vary in their infrastructure to support the lifecycle concepts, needs, and requirements definition activities. However, the list below can be used by an organization to assess their maturity level in supporting this effort and options for growth in their supporting infrastructure. The enabling processes and tools for lifecycle concept and needs definition, verification, validation, and management consist of the following (depending on the complexity needed):

- Needs and Requirements definition process tailored to the organizations workforce, culture, and product line.
- Checklists to guide the process activities (such as Appendix C).
- Guides/work Instructions (like this guide) and training material.
- Oversight management processes to ensure the activities are completed and the resulting artifacts are managed and configuration controlled.
- Toolsets to record and manage the needs and requirements and supporting and information data (reference Section 4.3.6).

Ideally these enabling products are defined and managed by the organization and would be another instance added to that infrastructure; these may also be found in public domain or industry standards and tailored to meet the needs of the organization.

Another key consideration as discussed in the NRM is moving from a document-centric to a data-centric practice of SE, often referred to as Model-Based Systems Engineering (MBSE). Using a data-centric practice, the needs and requirements and associated data and information are managed electronically establishing traceability across the lifecycle activities.

As an organization moves from a document-centric to a data-centric practice of SE, they may pursue toolsets as part of this planning process. Guiding questions in this step to be addressed are:

1. What processes/instructions currently exist? Are new or updated processes needed?
2. What level of configuration management is needed of the various artifacts?
3. What toolsets are in place to use?
4. What upgrades to existing tools in the toolset are needed?
5. Are the tools within the toolset able to share data? Can the data within the different tools be linked together?
6. What security is required? What export classification?
7. What licenses are needed?
8. Are project team members trained in the use of the tools in the toolset?
9. What criteria is defined to verify and validate the integrated set of needs against?

### 2.1.3 Step 1: Prepare for Lifecycle Concepts and Needs Definition

Since the primary output of the needs definition activity is the integrated set of needs, the initial step in planning for lifecycle concept and needs definition activities is to gather the inputs and enabling artifacts that will be used to support the elicitation, definition, analysis, capture, and management of the lifecycle concepts and needs as shown in Figure 4.

An additional input includes obtaining an understanding of the project and how the project will elicit and capture stakeholder needs and stakeholder-owned requirements, such as through a market driven approach or from a customer contract and stakeholder needs analysis effort.

### 2.1.4 Step 2: Define Inputs to Lifecycle Concepts Analysis and Maturation

During this step, a series of inputs are obtained that will consist of identifying the problem being solved, eliciting stakeholder input, identifying drivers and constraints, and assessing risk.

***"Define the Why" by establishing the Problem, Threat, or Opportunity and Mission, Goals, Objectives, and Measures.***

Prior to eliciting needs and requirements for the SOI, it is necessary for the project team to understand the problem, threat, or opportunity that is the focus of the project. The steps to defining the problem or opportunity include:

- Identify the business strategic and operational stakeholders that are impacted by the problem or threat or those who will benefit by pursuing the opportunity.
- Clearly define a statement of the problem, threat, or opportunity.
- Get stakeholder agreement.

Guiding questions in this step to be addressed are:

1. What measures would the stakeholders use to define success?
2. What is their intended use of the system in what operating environment?
3. What capabilities, features, functions, performance, quality, and compliance do they need from the delivered SOI?

For cases where there is no existing SOI (also known as a “green field” project), a common approach is to characterize the “as is” or “present state “of the organization in terms of the problem, threat, or opportunity and then characterize the “to be” or “future state” of the organization in terms of the resolution of the problem, neutralizing the threat, or the ability to pursue the opportunity.

For existing systems that need to be updated (also known as a “brown field” project), a common approach is to list the problems or issues with the existing “as-is” system and the reasons the SOI needs to be changed. Key information includes what they think needs to be changed and why, and what value will result from the change. What can the existing SOI no longer do, what performance needs to be improved, what changes need to be made concerning interactions with external systems, what updates are needed as a result of changes to applicable standards and regulations.

It is important to understand different perspectives. The problem/threat/opportunity, MGOs, and measures from a business perspective (developing organization or customer organization) may be different than the consumer's perspective, thus both must be addressed. The consumer does not care about the developing organizations profits, time to market, market share, etc. The consumer cares about how the resulting product meets their needs. Thus, there will be several sets of problem/threat/opportunity, MGOs, and measures that need to be defined and met by the project team from both a business perspective and a consumer perspective of the product to be developed. This may lead to conflicts, e.g., product price vs. profitability and market share.

An example is provided below, for a coffee maker. The format of the approach is not constrained to this example, the important aspect is to understand WHY a product is being developed – what is the business case for this SOI? What are the MGOs of the SOI? What measures will be used to establish acceptable performance of the SOI?

**Problem/Threat/Opportunity:** Marketing is seeing an increase of stay-at-home workers, purchasing more coffee makers. Existing coffee makes have single functions while consumers want a multi-function hot beverage maker to get a blend of options like traditional brew or espresso. There is a huge opportunity if first to market.

**Mission Statement:** Provide a home-based, one-stop, hot beverage facility.

**Consumer Goals:**

CG1) Obtain home brew coffee quickly
CG2) Obtain consistent output of brewed coffee
CG3) Eliminate the need for multiple appliances by including additional features such a variety of hot beverages (traditional brew, espresso, and teas)
CG4) Easy to use and maintain
CG5) Highly reliable and long lasting

**Business Goals:**

BG1) Reduce time to market
BG2) Increase profitability
BG3) Increase market share

**Business Objectives:**

O1) Stock store shelves within one year
O2) Generate coffee and espresso (including steamer/frothier)
O3) Maximize reusability of existing company assets
O4) Increase market share of Company X sales regions

**Consumer-focused Measures:**

M1) 99.9% accuracy within temperature based on user temperature option selected
M2) 99.9% accuracy within duration based on type of beverage selected
M3) 10-year service life

**Business-focused Measures:**

M4) 40% per unit profit
M5) Greater than 50% market share of Company X sales regions

In many cases the identification of the problem, threats, opportunities, MGOs, and measures are done at a business enterprise or operations levels, where the initial assessment results in the authorization for a project and associated budget along with an acquisition concept concerning what will be outsourced to suppliers and what part of the development will be performed in-house. For the project team, this means seeking an understanding of this content to ensure the outcomes of the project align with the organizations overall strategy behind developing that particular project.

Note: Stakeholder needs are expressed at different levels of abstraction. The MGOs and measures are the top of the hierarchy of the integrated set of needs for the SOI.

***"Define the Who" by Identifying External and Internal Stakeholders***

Stakeholders are the primary source of needs and requirements, therefore for the project to be successful, all relevant stakeholders must be identified and included at the beginning of the project. Leaving out a relevant stakeholder often results in missing needs and requirements and a failure to pass system validation.

A key step after project formation for the project team (consisting of both project management and system engineering) is to identify the relevant stakeholders. Key stakeholders will sign-off and accept the SOI, therefore they must agree to the integrated set of needs resulting from the lifecycle concepts and needs definition activities. It is important that key stakeholders are relevant and stay as consistent as possible as they will be accepting/approving the validation of the product that stems from the stakeholder needs. Guiding questions in this step to be addressed are:

- Who pays?
- Who profits?
- Who produces?
- Who tests (is involved in design and system verification and validation)?
- Who uses or operates?
- Who maintains?
- Who regulates?
- Who accepts, approves, certifies, or qualifies the SOI?
- Who owns or controls external systems that interfaces with the SOI?
- Who owns or controls enabling systems the SOI interacts with?

- Who is involved or who may be impacted by in the disposal of the SOI at end-of-life?
- Who else cares?

Stakeholders may be, but are not limited to, paying customers, sponsors, marketing, sales, regulators, developers, testers, installers, users/operators, maintainers, and the public at large, whether or not they are acknowledged. Stakeholders can be both internal and external to the organization developing the SOI.

Once these stakeholders and their needs and expectations are identified, the next step is to determine (a) who will decide what constitutes "necessary for acceptance", and (b) what *is* "necessary for acceptance". One cannot assume that the only stakeholder that counts is the "customer" (the one paying for the development) other stakeholders can be ranked as to their say in what is necessary for the SOI to be acceptable. When there is an inconsistency or a disagreement, the rank of a given stakeholder will be taken into consideration when weighing the options.

For a developer-funded effort, determining the stakeholders is primarily the responsibility of the developer, with the caveat the acceptance criteria defined by regulators must be included. For consumer products, users in the marketplace will determine the ultimate acceptability of the solution based on sales and social media feedback.

For an acquirer-funded effort, this task becomes a negotiation among the acquirer (the one supplying the resources), supplier (developer), user, and other stakeholders as determined by the acquirer. It is the supplier's responsibility to alert the acquirer regarding risk of ignoring a particular stakeholder and any concerns or characteristics "deemed necessary for acceptability" by these stakeholders.

In all cases, a failure to address the needs of a particular stakeholder, even if ignored early, may cause programmatic failure during system validation if the system is determined to be unacceptable during operations and maintenance in the operational environment by the intended users.

A truncated example of a Stakeholder Register is shown in Table 2, which is used to provide additional information about the SOI that would be gathered during stakeholder interviews. Stakeholders can be classified into Vested (V), Influence (I), where key stakeholders are often captured, or Participate (P). This classification could also be used to identify a cross matrix between stakeholder and their impact to the MGO and measures.

**Table 2. Example Stakeholder Register.**

| Stakeholder | Internal/ External | Involvement | Desired Outcome | Classification |
|---|---|---|---|---|
| Coffee Maker Company – New Product VP | Internal | Corporate Sponsor, Gate review approval | Meet agreed to MGOs and measures | V, I |
| Coffee Maker Company – Project | Internal | Responsible for Budget, Schedule, | Meet Budget, Schedule, and Technical Goals, | I, P |

| Stakeholder | Internal/ External | Involvement | Desired Outcome | Classification |
|---|---|---|---|---|
| Manager | | and Resources | Objectives, and measures | |
| Coffee Maker Company – Sys Eng | Internal | Responsible for Development of the SOI | Pass System Verification and System Validation within Budget and Schedule constraints | I, P |
| Coffee Maker Company – Test Eng | Internal | Oversee Test Campaign | Pass all System Validation and System Verification Activities | I, P |
| Consumers | External | Buy Product and Provide Feedback | Meets our Needs and Expectations | I |
| Regulatory Agency - FDA, CPSC | External | Provide Certification, Qualification, Acceptance | Safe for Consumers and Environment | I, P |
| Retailers – online stores, etc. | External | Provide Shelf (or virtual) Space | Popular in the Marketplace, Profitable. | I |

Once stakeholders are identified, their role in context of their involvement with the SOI production are captured to provide an understanding of their individual needs and expectations.

Note that is common that there will be inconsistencies in the stakeholder needs and expectations as well as uncertainties as to their feasibility. Also, not all needs and expectations are equal in terms of priority and criticality. The project team will need to address these issues. This could highlight that not all stakeholders are treated equally, based on their relative ranking, interest, involvement, and role concerning defining what is necessary for acceptance.

***"Define What is Needed" by Eliciting Needs from the Stakeholders***

The project team engages the stakeholders to understand their needs for the SOI for all lifecycle stages. It is often the case that the stakeholder needs may first be annunciated as “requirements”, even if not substantiated. These must be treated as other “needs” and be subject to the same reconciliation and prioritization as other needs derived from use cases, operational scenarios, MGOs, measures, and lifecycle concept analysis and maturation activities. Failure to reconcile the inconsistencies, prioritize, and assess critically of the stakeholder needs incurs technical debt and ongoing risk for system development because any inherent conflict among the needs will eventually be uncovered as the SOI development occurs.

Before conducting elicitation activities, the project team must plan for each stakeholder elicitation session or activity. Key things to decide on include:

- The stakeholders who will be involved, how frequently (once or multiple times) and when?
- The elicitation techniques that will be used. Brainstorming, Workshop/Focus Group, Interviews, Feedback/Document Analysis, Interface Analysis, Observation of comparable products, Site visits, Prototyping, Survey/Questionnaires, models, diagrams, etc.

- The types of information needed from the stakeholder(s)? See Appendix C for example questions.
- Existing documentation that is available—existing requirements, specifications, interface definitions, or user documents for a similar product, regulations and standards, customer survey info, problem logs, etc.
- How information will be captured during the elicitation sessions.
- The form and media that will be used to document and manage the captured information.

#### *"Capture the Needs" by Recording the Elicitation Outcomes*

Stakeholders often state their needs and expectations in natural language and may combine several needs into one statement, or stakeholders may express needs as bullets or phrases.

Various stakeholders can have similar needs that are stated in different ways yet have common meanings; for example, user-friendly and easy to use have some common underlying objectives. An example of needs for a coffee maker is shown below:

**Key stakeholder needs:** safe, affordable, cost-effective manufacturing, user-friendly

**Consumer needs:** affordable, high quality, good coffee, options for other hot beverages, easy to use

It is critical that all the information obtained from the elicitation activities be recorded *in the project's integrated dataset*. There are various forms and media that can be used to record the information gained from the elicitation activities. In some cases, e.g., use cases and models, there is a predefined format defined within some toolsets. In other cases, the information is captured as text. Below are common forms used.

- Operational Scenarios.
- User Stories.
- Use cases.
- Diagrams/drawings.
- Models.
- Lists.
- Tables.
- Text.

Table 3 identifies some examples of scenario/use case topics for a coffee maker as provided by the Stakeholders.

**Table 3. Stakeholder Scenario/Use Case Descriptions.**

| Operational Scenario/Use Case – Nominal operations |
| --- |
| **Initial conditions: Coffee maker is in the user's facility located on a counter and plugged into an electrical receptacle. The coffee maker is connected to the users WIFI or Bluetooth for remote monitoring on the user's mobile device. The coffee maker is powered on and in standby mode.**<br><br>**Assumption: The coffee maker allows the user to add consumables and receiving container at some predetermined time prior to the desired delivery time of the hot beverage, which can be preprogrammed.**<br><br>**Use case example:**<br><br>1. The night before coffee is needed, the user fills the water reservoir using unheated tap water.<br>2. User adds the coffee to be brewed<br>3. User adds sweetener and cream (if desired)<br>4. User selects the type of receiving container (coffee cup, insulated container for consumption while driving to work, or a multi-cup container.)<br>5. User inserts the receiving container<br>6. User selects the desired brew temperature<br>7. User selects the type of beverage (coffee or espresso)<br>8. Nominal path: User starts the brew activity<br>9. Optional path: User sets the time for the beverage to have completed brewing and filling the container.<br>10. Machine brews coffee to the container.<br>11. Upon brew completion, the user removes the receiving container. |

The above operational scenario/use case is a typical sequence of events as stated by a "typical" user for nominal operations once the coffee maker is installed in the user's facility. Other stakeholders will have different perspectives, for example: packaging, price, appearance, maintenance, cleaning, safety, government regulations, quality, lifetime, disposal, and the software application that resides on the user's mobile device.

For the scenario above, each step is a source of user needs and resulting requirements. The project team will have many questions concerning each step to better understand what is stated, the rationale, assumptions, and any missing steps or information.

The individual outcomes from the elicitation activities represent the unique perspective of each stakeholder or group of stakeholders. These perspectives will need to be analyzed and integrated into an integrated set of needs and stakeholder-owned requirements. The form used for elicitation is dependent on the organization, culture, processes, toolset, and domain.

During elicitation it is important to capture nominal, alternate nominal, and off-nominal use cases and operational scenarios as well as understand the relative importance (priority) and criticality of the stakeholder needs and requirements.

Stakeholders often state needs using ambiguous terms such as user-friendly, easy-to-use, fast, high quality, good tasting, affordable, robust, etc. During elicitation, when a stakeholder uses ambiguous terminology such that their expectations are not clear, the project team should question the stakeholder: “What do you mean by xxxxx?” “In your mind, how would you describe xxxxx?” For “fast” ask “How long should it take to provide a cup of coffee?”

For some need statements, further refinement is needed to understand the measures to define metrics that will be used to measure success for the SOI.

As an example, a need for the coffee maker to be “safe” is considered vague. By querying the stakeholders, examining safety regulations and compliance needs, prototyping a coffee maker, or reading unfavorable consumer feedback, the need can be analyzed and dissected or elaborated.

This information provides more specific information the project team can use for the transformation of the need statements into to requirements for the SOI (described further in Section 3).

**Key stakeholder need:** The coffee maker needs to be safe.

**Clarification of need:** The coffee maker must be safe—that is, meet UL 1082 standard and OSHA workplace standards.

**Need Statements:**

The stakeholders needs the coffee maker to meet the UL 1082 standard.

The stakeholders needs the coffee maker to comply with OSHA workplace standards.

Needs analysis is a key task in this step. The analysis identifies needs that the project team cannot control, achieve or require further elaboration in order to understand the real need.

During elicitation, the project team will discover conflicts, inconsistencies, and issues that they will need work with the stakeholders to resolve. The need set may contain a conflict if two stakeholders specified needs that are inconsistent. For example, the size of the cup differs between two stakeholders—one wants a 12-ounce cup another wants an 8-ounce cup. When resolving conflicts, the stakeholders may need to be consulted to explain the conflict and seek resolution.

### *"Establish the Boundaries" by Identifying Drivers and Constraints*

Drivers and constraints are things outside the project’s control that constrain or drive the solution space. For example, failing to show regulatory compliance, could result in the system failing certification and acceptance for use. Drivers and constraints represent a major source of needs and requirements and drive and constrain the lifecycle concepts analysis and maturation activities for the SOI being developed by the project.

Drivers and Constraints can include:

- Design constraints (parts, materials, organizational design best practices, etc.).

- Design standards (industry, domain, business management, business operations),
- Production constraints (existing technology, facilities, equipment, cost, throughput, etc.).
- Human factors (human/machine interface - HMI).
- Regulations (laws).
- Operating environment (natural and induced).
- Operating environment (social, cultural).
- Existing systems: (interactions, interfaces, dependencies).
- Technology Maturity.
- Cost.
- Schedule.
- Mission (operational usage) drivers.
- Higher-level requirements allocated to your SOI. At the system level, these will be business operations level stakeholder needs and requirements. At lower levels of the architecture, these will be requirements allocated to your SOI from the level above.

Concurrently with the stakeholder elicitation activities, drivers and constraints need to be identified and be recorded within the project's integrated dataset in a form that will enable traceability, which could be a large and time-consuming task.

As part of identifying drivers and constraints, the project team also needs to identify any specific needs that are to be considered for the SOI with respect to manufacturing, operations, maintenance, or disposal. For example, enabling systems such as special facilities and equipment needed for system verification or system validation activities, special tooling needed for manufacturing, or any regulations concerning disposal due to hazardous materials.

There could be certain processes that need to be followed and special instructions for people (users/operators, manufacturers, testers, installers, maintainers, etc.). In general, these will come from industry standards/processes or company specific standards and processes on how the project will handle these considerations. For our coffee maker example, an understanding of the safety regulations could result in a need to include a label "Caution Hot Surfaces", at locations on the coffee maker that are hot to the human touch.

Guiding questions to ask in identifying drivers and constraints include:

- What are the relevant standards and regulations?
- What are the production processes/workmanship/facility limitations?
- Are there required design approaches (industry, mechanical, electrical, software, business management, etc.)?
- How will the SOI be verified and validated (labs, facilities, etc.)?

- What are the Human Factor considerations for the SOI?
- What are the critical technologies and what is their maturity?
- What are the existing systems that the SOI must interact with? Have those stakeholders been involved in elicitation?
- What are the higher-level requirements allocated to the SOI?
- What are the budget and schedule constraints?
- What is the operating environment of the SOI?

***"Understand Risk to Success" by Identifying and Analyzing Risks***

Risks concerning the SOI being developed should be identified and captured during the elicitation activities as initial concerns are mentioned. These are risks associated with development of the product. As part of this step those initial concerns and other risks will be captured and managed more formally. From the SOI perspective, the project team will need to decide how each risk is to be handled.

The risks that will be mitigated by design of the SOI must be clearly identified so they can be addressed during the lifecycle concept analysis and maturation activities and included in the integrated set of needs, The guiding questions to ask in addressing risks, threats, and opportunities are:

- What are the management risks? Related to budget, schedule, resources.
- What are the development risks? Related to SOI development/quality.
- What are the production risks? Related to facilities, tooling, etc.
- What are the system integration risks?
- What are the system verification and system validation Risks? Related to test resources, etc.
- What are the Compliance risks? Related to proper regulatory authority passing SOI.
- What are the Operational risks? Related to the intended use and operational environment.
- Which risks will be mitigated by the design of the SOI?

### 2.1.5 Step 3: Capture Preliminary, Integrated Set of Lifecycle Concepts

This can be one of the most frequently missed steps in the process for developing a SOI, yet this activity establishing foundation information which is used to ensure the SOI aligns with the integrated set of needs.

A **lifecycle concept** is a written or graphic representation that concisely expresses how an entity will satisfy the problem, threat, or opportunity it was defined to address within specified constraints with acceptable risk. The results of the proceeding activities are integrated into a set of preliminary lifecycle concepts and supporting data obtained from the activities used to define the inputs to the lifecycle concepts for the SOI. The set of lifecycle concepts can include concepts for acquisition, development, design, integration, verification, validation, transportation, integration, deployment, operations, support, maintenance, and retirement. These are preliminary in the sense that the detailed analysis concerning feasibility, completeness, consistency, and correctness has not been completed. The specific lifecycle concepts used depend on the organization, its product lines, processes, and culture.

Figure 5 shows an example of assessing lifecycle concepts using scenario/use cases for a coffee maker in a system model of the SOI as part of a model-based systems engineering (MBSE) approach. Note: These figures are only a few examples of creating models of the problem, threat and opportunities to highlight types of use cases that can be created.

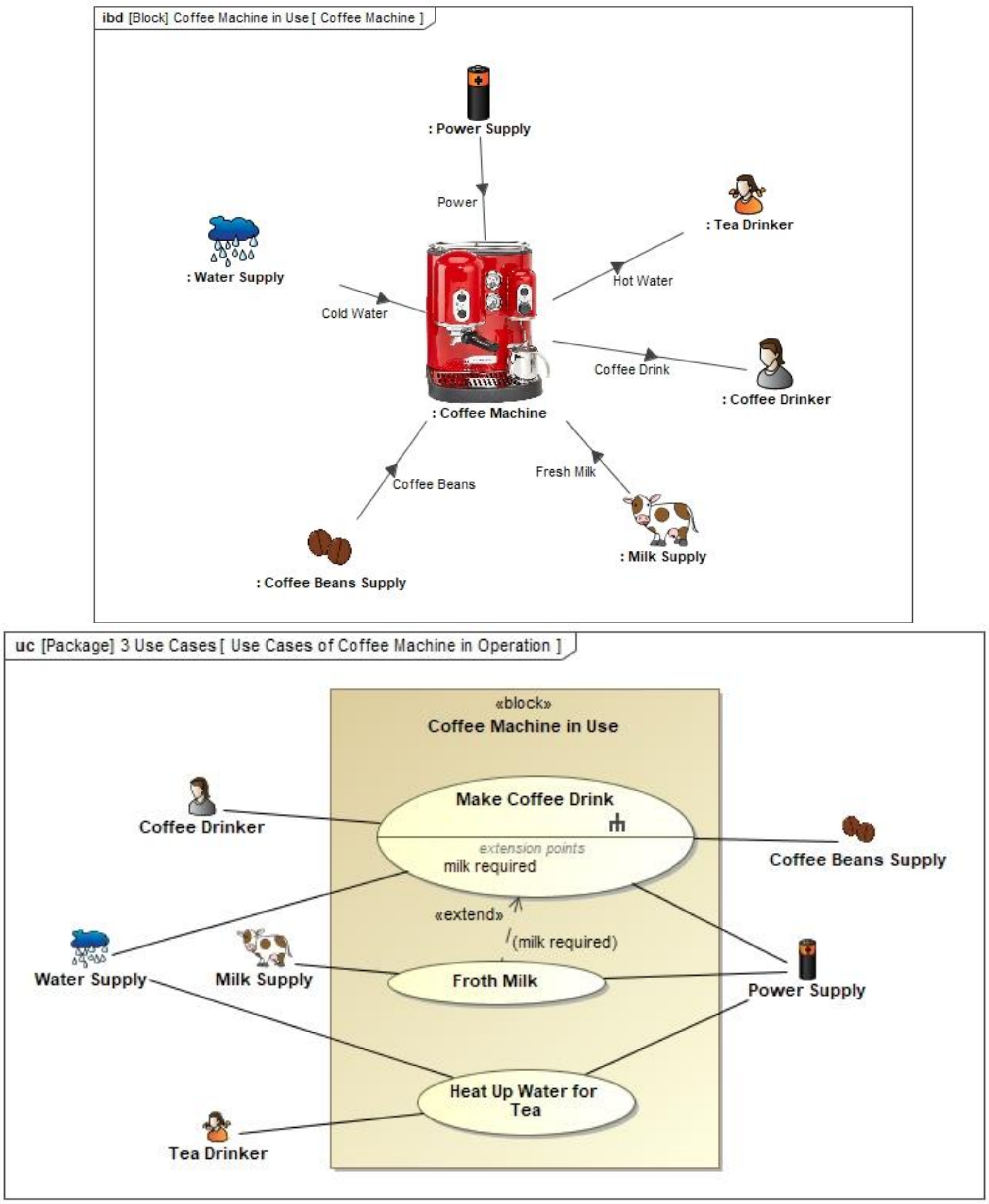



**Figure 5. Coffee Maker Use Case Model Examples.**

Each stakeholder has a unique perspective, the sum of which results in an integrated set of scenarios/use cases to help frame the needs/requirements to the lifecycle stage activities for the SOI and provide context for the integrated set of needs. The objective of this activity is to integrate the use cases and operational scenarios recorded as a result of the integrated scenario elicitation activities.

Reviewing the entire SOI lifecycle and establishing preliminary concepts and associated stakeholders supports the development of an integrated set of needs which will drive the lifecycle concepts analysis and maturation activities discussed in NRM Section 4.4. An example approach is shown in Table 4, where each column of the table represents a lifecycle stage for the SOI and each row represents an operational scenario or use case for a specific stakeholder (or group of stakeholders) for the lifecycle stages that stakeholder has a stake or involvement. Another way to accomplish this association is to include in the stakeholders and lifecycle concepts into the system model and build in a relationship between them.

**Table 4. Integrated Preliminary Set of Lifecycle Concepts (from NRM).**

| Stakeholder (S)/ Lifecycle (L) | L1 | L2 | L3 | L4 | L5 |
|---|---|---|---|---|---|
| **S1** | X | | X | | X |
| **S2** | | X | X | | |
| **S3** | X | | X | | X |
| **S4** | | X | | X | X |
| **S5** | | | X | X | X |
| **Combined** | XXX | XXX | XXX | XXX | XXX |

The task of the project team is to combine the various operational scenarios or use cases into an integrated set of operational scenarios or use cases for each lifecycle stage. This task can be challenging in that each stakeholder or stakeholder group may have a different perspective based on their role, unique needs, experience, and political environments both internal and external to the organization.

While integrating the lifecycle concepts and other information from the elicitation activities in the previous step, a valid need may arise but later be revealed as inaccurate, ambiguous, inconsistent, not needed, or not feasible. For example, an assumption was made that the coffee maker needs to add heated cream or milk to the beverage.

**Need**: The stakeholders need the coffee maker to heat milk.

**Clarification of stakeholder need:**

Is there any need to separately heat another fluid other than water? If so, what? **Response:** Yes, users would like to have the coffee maker add heated creamer, cow's milk, soy milk, almond milk, or coconut milk once the beverage has been brewed.

**Need Statement**: "The users need the coffee maker to separately heat creamer, cow's milk, soy milk, almond milk, and coconut milk to then be added to the brewed coffee."

The project team can resolve these issues and restate the need in a form that is not design-specific, yet the design can be validated for heating cream, cow's milk, soy milk, almond milk, and coconut milk as an example of meeting this need. Through discussion, it may be determined that the priority of the capability to heat fluids other than water is a lower priority than the safety of the coffee maker. Furthermore, the customer survey did not support the real need for such a capability but instead supported the desire for the coffee maker to process pre-packaged products that result in soy, almond, or coconut milk coffee and lattes.

During this analysis, a new need was discovered—some stakeholders want a coffee maker that is capable of processing a variety of pre-packaged coffee, tea, and milk products—not to heat the milk but to provide heated water for the prepackaged product. This need, in turn, will result in the project team to investigate specific aspects (feasibility) associated with meeting this need and deciding whether to including this new need in the integrated set of needs that are agreed to by the stakeholder(s). While some stakeholders would like this feature, a marketing analysis may conclude that the targeted consumer base would prefer to not use pre-packaged products and the added complexity to the coffee machine to include both types of product and resulting increase in cost, could result in lower sales and lower market share.

During the elicitation, there is often a need to make assumptions regarding functional and performance criteria to allow the overall work to continue. It could be the case where further analysis or research is required that would interrupt the overall work in progress to develop an integrated set needs and resulting requirements.

Following the elicitation activities, the project team should ensure:

- All relevant stakeholders have been involved in the elicitation activities.
- All elicitation outcomes have been accounted for and captured.
- The elicitation outcomes are communicated at the right level of abstraction. Are they an implementation-free understanding of the needs and requirements for the SOI by defining what is expected (design inputs) without addressing how (design outputs) to satisfy the MGOs and achieve the defined measures?
- Issues and risks have been recorded and a plan for mitigation has been established.
- Rationale has been captured for each need.
- The priorities and critically of the needs have been established and recorded.
- Any conflicts and inconsistencies have been captured and a plan for resolution developed.
- Ambiguous terms such as user-friendly, easy-to-use, fast, high quality, good tasting, affordable, robust, etc. have been resolved.
- The feasibility of the recorded outcomes have been assessed and those that are questionable have been identified for further analysis.
- All interactions with external systems identified during elicitation has been recorded.

Stakeholders have the knowledge to validate whether the preliminary set of lifecycle concepts reflect the intent of their stated needs and requirements. Definition of a set of integrated lifecycle concepts requires the integration of a number of disparate views, which may not necessarily be harmonious. Before proceeding, it is critical that the project team has confirmation from the stakeholders that they understand and agree with the problem statement, MGOs, measures, strategic and business operations level lifecycle concepts, stakeholder needs and requirements, drivers and constraints, and risks as currently recorded within the project's integrated dataset.

This is especially true concerning how the project team plans to resolve inconsistencies, disagreements, ambiguity, and feasibility issues identified previously. There are a variety of methods of capturing this information for presentation; it could be in a document, within a model, presentation charts, matrix, diagrams, etc. depending on the stakeholder audience and toolsets being used by the project team.

***Considerations for SOIs being developed to Customer Requirements***

Even in cases where the customer clearly provides a set of requirements, understanding the intended use of the SOI in its intended operational environment by its intended users could result in identification of gaps that the customer may not have realized that needed to be addressed when defining, analyzing, and maturing the lifecycle concepts, defining the integrated set of needs, and transforming them into the set of requirements. By only providing a product that addresses the customer supplied requirements, the SOI may pass system verification but fail system validation.

One approach the project team can take is to use the checklist in Appendix C to ensure all lifecycle concepts and considerations are included within the customer requirements and identity any gaps. Resolution of those gaps can be addressed by development, analysis, and maturation of an integrated set of lifecycle concepts associated with the information gathered during elicitation activities (keeping in mind there will be different perspectives (management, operational, system, etc.).

### 2.1.6 Step 4: Analyze and Mature the Lifecycle Concepts

In the proceeding step, a preliminary set of integrated lifecycle concepts has been defined. In this step the project team does a more detailed analysis of these concepts to mature these concepts to establish completeness, correctness, consistency, and feasibility. Refer to the NRM for a more detailed elaboration of this step. This analysis is often referred to as "stakeholder needs analysis "as the major outcome of these activities will be a mature set of lifecycle concepts from which the integrated set of needs will be derived,

The use cases and operational scenarios for each lifecycle stage that were captured during elicitation are used to develop or build upon a system model of the SOI. The focus of this model is for the SOI to be developed in the context of all lifecycle stages. This model will be matured, a preliminary architecture defined, and the integrated set of needs will be derived during these activities; an example model for the coffee maker operational scenario is shown in Figure 6. Note: This figure is one illustration of creating a model depicting how the architecture addresses the problem, threat and opportunities;

in actuality there would be several of these types of activity diagrams created for the SOI.

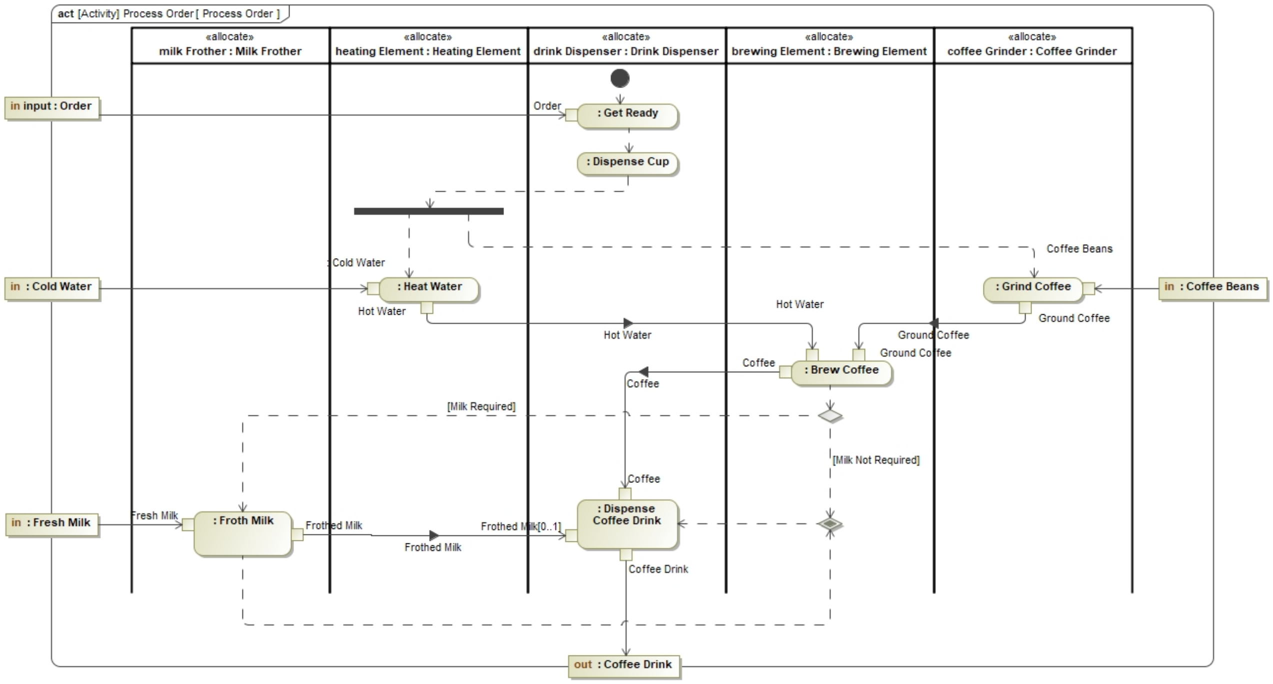



**Figure 6. Operational Scenario Model Example.**

From the uses cases and operational scenarios, the project team can identify the expected functionality, performance, and interactions with external systems. This information is used to build a variety of models and diagrams to help better understand the SOI to be developed. Nominal, alternate nominal, and off-nominal cases are included in the models for each lifecycle stage.

As part of the modeling effort missing functions, inputs, outputs, interactions with external systems are identified as well as inconsistencies and needs that may have been incorrectly or ambiguously stated. Concepts concerning operational risks that are to be mitigated by the design of the SOI, interactions with external systems, quality expectations, regulations and standards are defined, matured, and included in the model.

There may be several cycles of analysis needed. In order to establish feasibility, the project team will need to develop a physical architectural model in order to identify and assess the maturity of critical technologies and assess the feasibility of meeting the expectations of the stakeholders. It is from this underlying analysis that the integrated set of needs is based.

In the proceeding steps the project team should have captured any conflicts and inconsistencies, ambiguities, issues concerning feasibility, operational risks, etc. which will need to be reconciled by the project team as part of this step to analyze and mature the lifecycle concepts.

The outcome of the lifecycle analysis and maturation step is at least one feasible set of lifecycle concepts that will result in the MGOs, measures, stakeholder expectations being met within the defined drivers and constraints, with acceptable risk for this lifecycle stage from which the integrated set of needs will be derived.

### 2.1.7 Step 5: Define and Record the Integrated Set of Needs

This step involves the derivation of an integrated set of needs from the mature, feasible set of lifecycle concepts defined in the previous step. A **need statement** is the result of a formal transformation of one or more lifecycle concepts into an agreed-to expectation for an entity to perform some function or possess some quality (within specified constraints with acceptable risk).

Developing and agreeing on a set of feasible lifecycle concepts for the SOI results in an integrated set of needs based on those concepts. Per the definition of a "need", the project team derives an integrated set of needs that reflect the set of feasible system lifecycle concepts selected, the MGOs, measures, business operations level and system level stakeholder needs and requirements, drivers and constraints, and risk mitigation. It is this integrated set of needs that will be transformed into the set of requirements for the SOI and that is the subject of requirement, design, and system validation as discussed later in this guide. The sources of the integrated set of needs are shown in Figure 7.

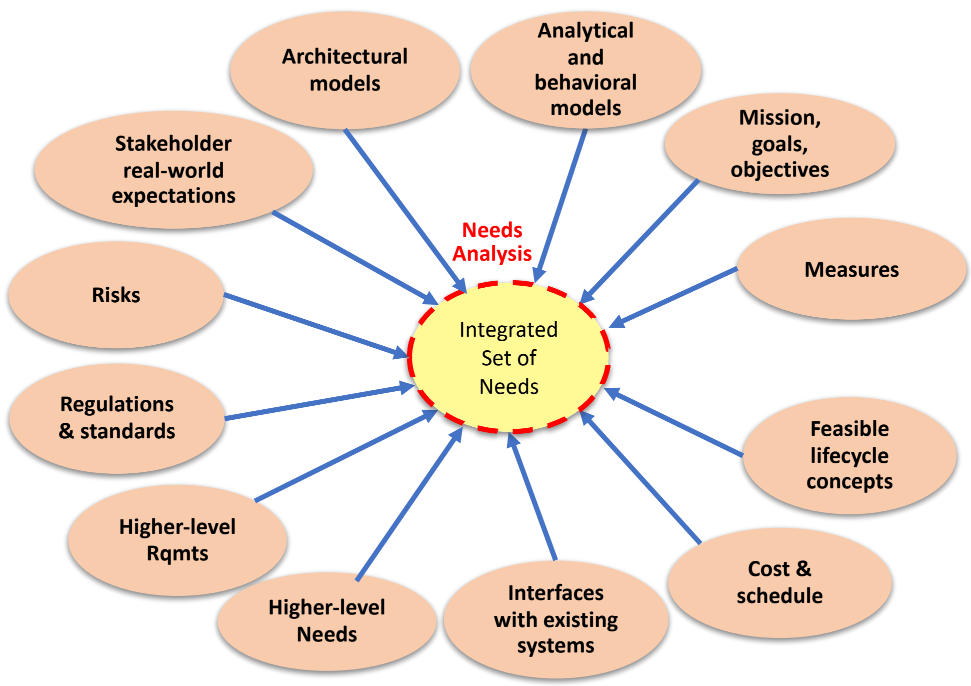



**Figure 7. Sources of the Integrated Set of Needs (from NRM).**

Need statements use structured, natural language statement from the perspective of what the stakeholders need the SOI to do, as contrasted with requirements that are stated from the engineering perspective of what the SOI needs to do to meet the needs. A structured set of agreed-to need expressions for the SOI represents a need set that accurately reflects the intent of the stakeholders' needs and expectations. To help distinguish needs from the requirements, the needs statements do not include the word "shall".

- Using "shall" in system need statements results in confusion. One approach to avoiding this confusion use the format: "The stakeholders need the system to ………" or for a goal "The stakeholders would like the system to ……."
- These statements are in contrast to the form of the SOI's technical requirements, "The System "shall" …" or for a goal, "The system "should" ……"

Using these distinct formats helps make clear what is a need statement and what is a technical requirement. The GtWR defines the structure and characteristics of well-formed need statements and sets of needs as well as a set of rules that help to write need statements that have those characteristics. Sample need statements are shown in the coffee maker example below:

**Key stakeholder and consumer need statements:**

- The stakeholders and consumers need the coffee maker to be safe.
- The stakeholders and consumers need the coffee maker to operate using standard home electrical power at the region where this product will be sold.
- The consumers need to be able to remotely program the coffee maker.
- The consumers need the coffee maker to make other types of beverages.

Need statements are expressed at a higher level of abstraction than the requirements that are transformed from them. The focus is on the SOI as externally observed. The needs statements should also be stated such that the SOI can be validated that it meets the need.

As an example, the need for the coffee maker to "be safe" is vague. By querying the stakeholders, examining safety regulations and compliance needs, prototyping a coffee maker, or reading unfavorable customer feedback, the need can be analyzed and dissected or elaborated. The resulting needs statement provides a specific measure to use for transformation to requirements for the SOI (described further in Section 3).

**Key stakeholder need:** The coffee maker needs to be safe.

**Clarification of need:** The coffee maker must be safe—that is, meet UL 1082 standard and applicable OSHA workplace standards.

**Need Statements:**

The stakeholder needs the coffee maker to comply with the UL 1082 standard.

The stakeholder needs the coffee maker to comply with applicable OSHA workplace standards.

For this example, it is acceptable for a need statement to state the specific standard or regulation concerning safety. It is also acceptable to use the word "applicable" in that from the stakeholder perspective that is their expectation. However, this level of abstraction is too high for what would be communicated within a requirement statement. The system level technical requirements would be more specific as to exactly which specific requirements in those standards or regulations are applicable and are being invoked on the SOI.

When managing the needs, it is best to define **attributes,** or additional information, that aid in the understanding or management of the need; an example of this is with an attribute called "Rationale". In the above example, the need statement about affordability might have an attribute designating it as “price-related” to collect and manage price-related needs. If this need is important, the need statement may have an attribute designating “high” for priority.

A **need expression** includes a need statement plus a set of associated attributes, including validation attributes. An example of two need expressions captured in a requirements management tool (RMT) is shown in Table 5.

**Table 5. Example of Need Expressions**

| ID | Name | Need Statement | Rationale | Safety Related | Priority | Price-Related | | |
|---|---|---|---|---|---|---|---|---|
| COF-MN-1 | Safe | The stakeholders and consumers need the coffee system to be safe. | Key stakeholder: ensure reputation of product; meet regulations Consumer: not get hurt making coffee | Yes | High | No | 2 | 1 |
| COF-MN-2 | Affordable | The stakeholders and consumers want the coffee system to be affordable. | Key stakeholder: maximize profits Consumer: minimize cost | No | High | Yes | 1 | 0 |

To ensure alignment of project data, the integrated set of needs must be recorded within the projects integrated dataset in a form and media suitable for review and feedback from the stakeholders. It is critical that the project team has confirmation from the stakeholders that the project team understands their expectations, resulting needs and requirements, drivers and constraints, and risks as communicated in the SOI’s integrated set of needs. It is also critical that the stakeholders agree with the MGOs as well as the measures. This agreement is critical because the integrated set of needs represents what is “necessary for acceptance” of which the SOI will be validated against. The chosen method for recording the integrated set of needs should allow:

- Traceability to the sources of the needs.
- Traceability between each need and the requirements transformed from the need.
- Including attributes for the need statements resulting in need expressions
- Traceability to system validation artifacts.

***Considerations for SOIs being developed to Customer Requirements***

When a customer has supplied a set of requirements for the SOI to a supplier, the supplier should then define a set of lifecycle concepts, analyze and mature those concepts, and derive an integrated set of needs that will result in the achievement of the customer requirements and that can be used as the rationale for the supplier requirements transformed from those needs. Getting agreement from the customer on this integrated set of needs ensures the customer and supplier are on the same page and they represent what is necessary for acceptance and to which the SOI will be validated against.

For suppliers developing a product to a customer provided set of requirements, it is tempting to look at that set of requirements as the *only* set of requirements. In reality, there are other stakeholder needs and requirements that must also be considered to ensure the product is able to be developed, integrated into the overall system, and enable system validation. This is highlighted in Figure 7, with specific examples for an acquired SOI shown below:

- The production team needs the product to be manufacturable.
- The test team needs the product to be testable (able to be verified to meet the requirements and validated to meet the integrated set of needs).
- The customer and public needs the product to be safe and also secure from a cyber perspective.
- The users need the product to be easy and safe to interface with from a human factors perspective.
- The supplier organization may need this product to conform to a strategic development effort aligning with technology maturation.
- The product may need to address regulatory compliance.
- There is the potential that the customer may have missed a use case related to some compatibility with existing infrastructure within the operating environment in their development of their requirements.
- There is the potential that the customer may have missed an external system or enabling system with which the SOI may interact.
- There is the potential that the customer may not have completely communicated their expectations for quality.

Needs generated from the above examples are not always specified in the contract and customer set of requirements. Any of these example considerations could lead to derived needs that must be captured in some fashion to develop requirements on the product in addition to the customer requirements.

If suppliers only follow their contractual requirements, they are likely to generate a SOI that may not work as needed when integrated into the macro system of which it is a part, or reduce ability for validation of the integrated system in its operational environment, leading to reduction in stakeholder demand for that product. In some ways these may compete with the customer requirements, which is when a reconciliation effort will occur to assess if a customer need takes priority, or the supplier's (in which case the negotiation of waivers and contract updates occur).

This type of analysis is greatly enabled by the use of language-based models, enabling a system model of the product in its operating environment showing the concept of operations, interfaces, mission profile, assumptions on how the SOI is used and interacted with, enabling an analysis of "missing" customer requirements against the real needs and uses by the intended users in the intended operational environment.

### *Organizing the Integrated Set of Needs*

The integrated set of needs represent the agreed-to outcomes of the lifecycle concept analysis and maturation activities. These outcomes include functionality (what the stakeholders need the system to do), expected performance and quality (the “how well” characteristics), the conditions of action, including triggering events, system states, and operating environments (the “under what operating conditions”), as well as compliance (with standards and regulations).

To help with the development and organization of the requirements that will be transformed from the integrated set of needs, it is useful to organize the needs into the following groupings: function, fit, form, quality, and compliance.

- **Function**: The action or actions that the SOI needs to perform. Function includes functionality and associated performance. The functions address the features the stakeholder expect the SOI to have, performance addresses how well, how many, how fast, etc. attributes of a function.
- **Fit**: The ability of the system to interface with, interact, connect to, operate, and become an integral part of the macro system it is a part of. “Fit” includes human system interactions and interfaces as well as both the induced and natural environments (conditions of operations, transportation, storage, maintenance).
- **Form**: The shape, size, dimensions, mass, weight and other visual parameters that uniquely distinguish a system. “Form” addresses the physical nature of the SOI. For software “form” could address programming language, lines of code, memory requirements, etc.
- **Quality**: Fitness for use (safety, security, and various quality “ilities”, e.g., reliability, testability, operability, availability, maintainability, operability, supportability, manufacturability, interoperability, safety, security, to name a few.
- **Compliance**: Satisfying design and construction standards and regulations

A key advantage of using this form of organization is to help ensure the integrated set of needs has the characteristic *C10, Complete* as defined in the GtWR. Each of the above groupings represents a distinct perspective and source of the needs. Failing to consider each perspective will result in missing needs and corresponding requirements.

The function and fit needs come from both stakeholder elicitation as well as the diagrams and models developed during lifecycle concept analysis and maturation activities. However, relying strictly on diagrams and models as a source of needs often results in an incomplete set of Needs. Form and quality needs are system attributes more closely associated with the physical system rather than a functional architecture or analytical/behavioral models of the physical system. Compliance needs and most of the drivers and constraints come directly from the stakeholder elicitation activities.

### 2.1.8 Step 5: Plan for System Validation

The outcome of a project is a validated system. System validation is obtaining data that can be used as evidence to show that the verified SOI satisfies the integrated set of needs, MGOs, and measures that, together, define what is *necessary for acceptance* as discussed earlier. The GtWR includes the characteristic, *C14 - Able to be Validated*, for a well-formed integrated set of needs. "Able to be Validated" means that the project team and the customer will be able to validate that evidence will be able to be obtained needed to determine that the resulting SOI will be able to meet the integrated set of needs.

A best practice to ensure the needs are "able to be validated", is to plan for how the project will validate that the system will meet the integrated set of the needs during system validation activities that occur during system integration, system verification, and system validation as discussed in NRM Sections 10 and 11.

Information that should be defined concerning system validation for each system need statement includes the validation method, strategy, success criteria, and the organization responsible for planning and executing the system validation activities. This information can be defined within the validation attributes that should be included (along with the other attributes discussed in NRM Section 15 within each need expression).

### 2.1.9 Step 6: Baseline and Manage the Needs Definition Outputs

Needs can be recorded and managed via a document, database, or model-based approach. The chosen method, in addition to representing the need statement, should allow the assignment of:

- Stakeholders who have the need.
- Rationale for the needs statements.
- Attributes, such as need number, source (if a document), priority, etc.
- Agreed-to artifacts and date of agreement.

Changes to needs must also be managed. Pending changes can be communicated through the chosen approach or via the communication channels earlier agreed to. Using a data-centric approach to SE, traceability can be established within the project toolset and the traceability matrices can be represented as reports generated by the toolset. Overall approaches to managing an integrated set of needs is described in Section 4.

## 2.2 Needs Verification and Needs Validation

Given the possibility of conflicting or poorly defined stakeholder needs, the activities of needs verification and needs validation is important to ensure the project team is designing and delivering the right product. The focus of the needs verification and needs validation is on the activities used to elicit and define the integrated set of needs.

- Has an acceptable elicitation process been followed?
- Have all the stakeholders been consulted?
- Has a feasible, correct, consistent, and complete set of lifecycle concepts been defined that reflects the MGOs, measures, stakeholder needs, stakeholder-owned requirements, and drivers and constraints with acceptable risk?
- Are the needs the right needs—do they accurately represent the agree-to lifecycle concepts and other sources as shown in Figure 7 from which they were transformed?
- Do the needs correctly and completely capture what the stakeholders need the SOI to do in the operational environment in terms of form, fit, function, compliance, and quality when operated by the intended users in the intended operational environment?

Verification and validation of SOI needs should result in a stakeholder agreed-to integrated set of needs. Signatures by the relevant stakeholders and the approving authority may be needed to confirm agreement that the integrated set of needs represents the real-world expectations of the stakeholders and represents what is necessary for acceptance. The overall system verification and validation activities is shown in Figure 8, with the detail view of the needs verification and validation activities shown in Figure 9.

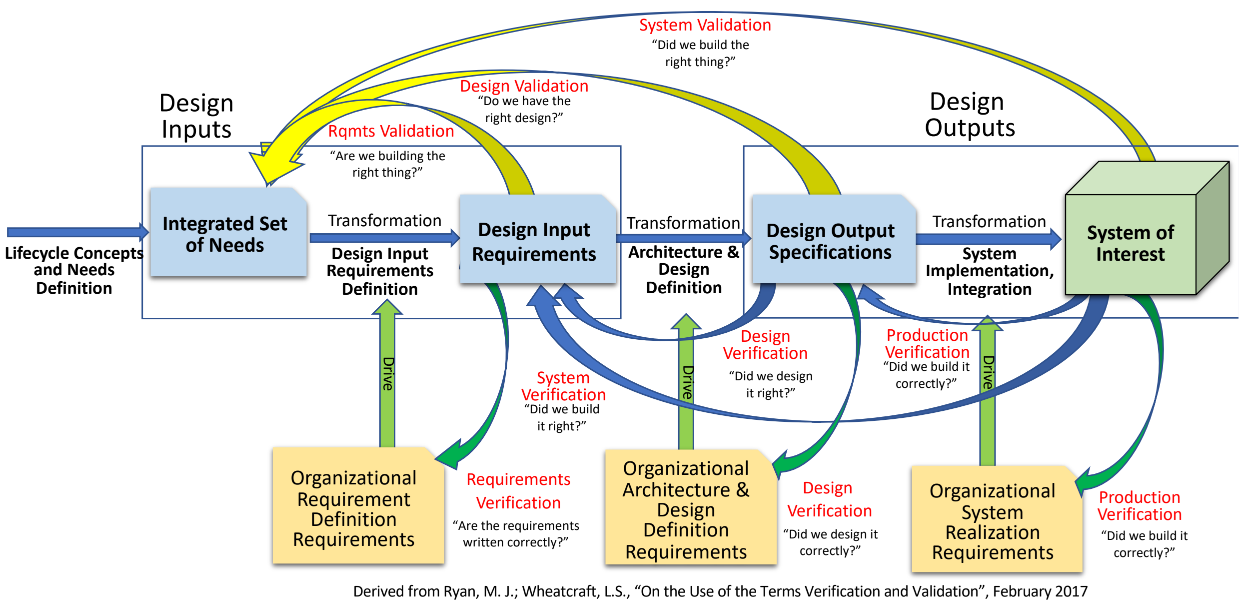



**Figure 8. Verification and Validation Concept Overview (from NRM).**

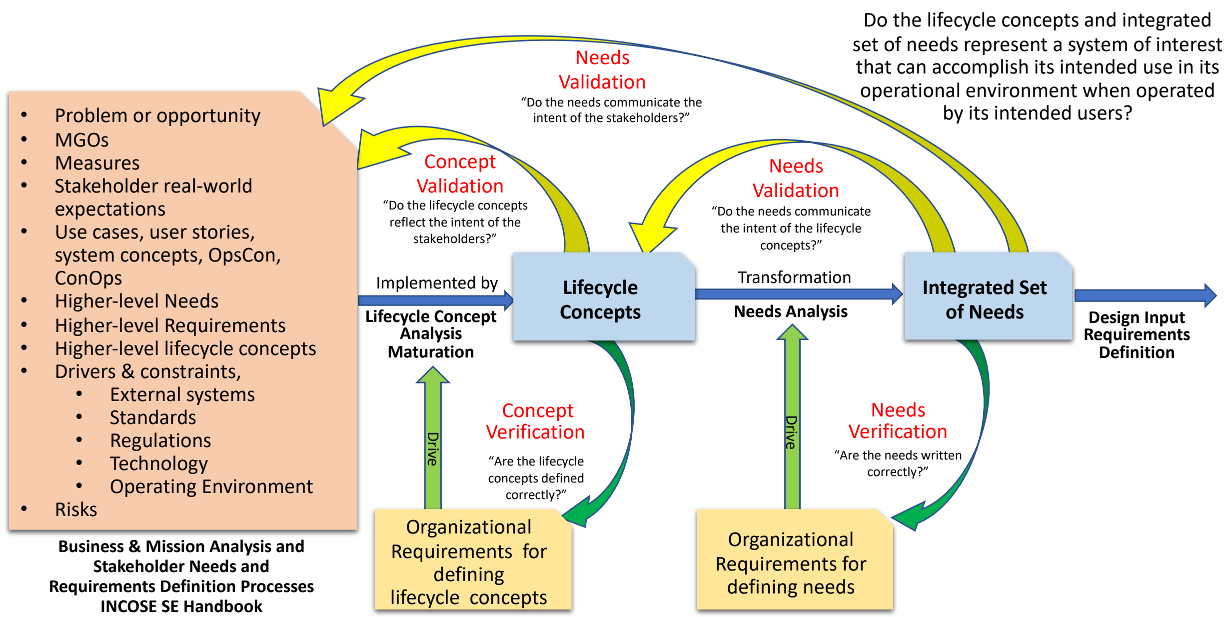



**Figure 9. Needs Verification and Validation Overview (from NRM).**

### 2.2.1 Enabling Processes and Tools

Organizations vary in their infrastructure to support the lifecycle concepts, needs, and requirements definition activities. However, the list below can be used for an organization to assess their maturity level in supporting these activities and options for growth in their supporting infrastructure.

The enabling processes and tools for lifecycle concept and needs definition, needs verification, needs validation, and management consist of the following (depending on the complexity needed):

- Enterprise tailored Guide to Writing Requirements (based on the INCOSE GtWR).
- Enterprise tailored process for developing and managing needs and requirements.
- Tailored checklist for needs verification and needs validation criteria and tasks (see examples in Appendix D).
- Enterprise process for product development.
- Trained systems engineers.
- Needs and requirements management tool(s).
- Diagramming and modeling tool(s).
- Stakeholder representation.

Of particular importance are the needs and requirements management and modeling applications within the project's toolset used for the underlying analysis used to produce the input artifacts from which the needs were transformed and recorded.

## 2.2.2 Needs Verification

Need expressions should have the characteristics of a well-formed need expressions and the integrated set of needs has the characteristics of a well-formed sets of needs as defined in the GtWR.

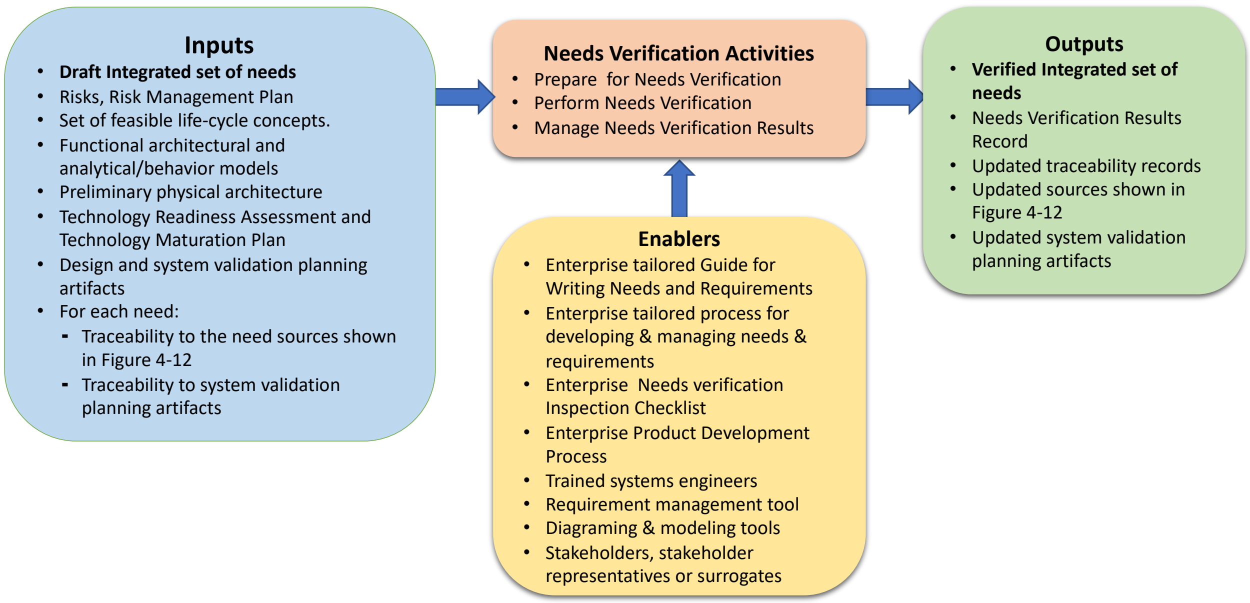



**Figure 10. Needs Verification IPO Diagram (from NRM).**

Figure 10 provides the inputs, outputs, activities and enablers for the Needs Verification activities from the NRM. The basic steps are:

- Plan for Needs Verification.
- Perform Needs Verification.
- Manage Needs Verification Results.

### 2.2.2.1 Step 1: Plan for Needs Verification

This activity provides the planning to ensure specific processes, tools and resources are addressed to perform an assessment of need expressions against standards. Based upon approach, this could be standards for well-formed need expressions defined within the GtWR, or standards for modeling efforts in MBSE standards. Planning for a verification effort includes determination of an approach towards the capture of needs, and standards of convention for formulating needs expressions.

While some of the of the needs verification activities must be done manually, others may be able to be automated depending on the capabilities of the applications in the project toolset. To what extent the project can rely on automated needs verification depends on a tradeoff between effort and risk. The need for and importance of using automation for needs verification will increase as systems become more complex and the number of needs increases.

A key preparation activity for needs verification is the creation of a "Needs Verification Inspection Checklist". If the organization has a generic checklist, they can tailor the checklist to the system being developed and the project's processes. This checklist serves as a standard to measure the needs against and will help guide all the needs definition and verification activities. Having addressed each of the areas and questions within the checklist will help lead to successful completion of the lifecycle concepts and needs definition activities as well as the needs verification activities. An example Needs Verification Inspection Checklist is included in Appendix D.

#### 2.2.2.2 Step 2: Perform Needs Verification

There are several activities to ensure the individual needs expressions and integrated set of needs are well-formed and provide objective evidence they have been verified. Guided by the Needs Verification Checklist (Appendix D), the project team will verify the need expressions are well formed, contain attributes supporting their management, and that the integrated set of needs are complete. Needs verification in terms of traceability and attributes can to some extent be automated using management tools, enabling advanced analysis involving traceability of needs to the input artifacts. As a minimum, these tools can identify missing traceability or attributes via reports (example is a report generated to list all needs that do not trace to a source or a source that does not have an implementing lifecycle concept and resulting need).

Note: While applications within the project toolset can produce trace reports, these applications cannot yet determine the accuracy and correctness of the traces between entities. This assessment must be done manually.

#### 2.2.2.3 Step 3: Manage Needs Verification Results

A needs verification record is created recording the results and outputs of the needs verification activities. Defects found in the integrated set of needs as well as any defects found in any of the input SE artifacts or PM work products, must be addressed and corrected before they are baselined.

If it is identified that there is insufficient data to complete needs verification or a lack traceability records within the project toolset, the issue should be considered to be a risk that could cause subsequent requirement, design, and system validation activities to be unsuccessful.

Applications within the project toolset should allow tracking of the status of the needs verification activities using a dashboard that communicates key metrics related to the needs verification results. These metrics can be obtained using the need attributes contained within each need expression. Making this information accessible will enable all project team members and key stakeholders to have the same view of the maturity and status of the needs verification activities. Overall approaches to managing an integrated set of needs is described in Section 4.

### 2.2.3 Need Validation

Even though individual needs statements and the integrated set of needs may be well-formed, and traceability has been verified, the message they communicate may not be as intended. It is important that organizations do both needs verification and needs validation; failing to do so adds risk that the SOI being developed will fail to do what it was intended in its operational environment when operated by its intended users and not meet the validation success criteria that defines what is necessary for acceptance.

The needs validation activities examines the needs expressions and the integrated set of needs to validate each need statement and the integrated set of needs clearly communicate the intent of the sources from which they were transformed, in a language understandable by all project team members, customers, and other key stakeholders. The needs validation activities will determine if the integrated set of needs will result in an SOI that does what it was intended in its operational environment when operated by its intended users and will be acceptable to the customer, users, and other stakeholders.

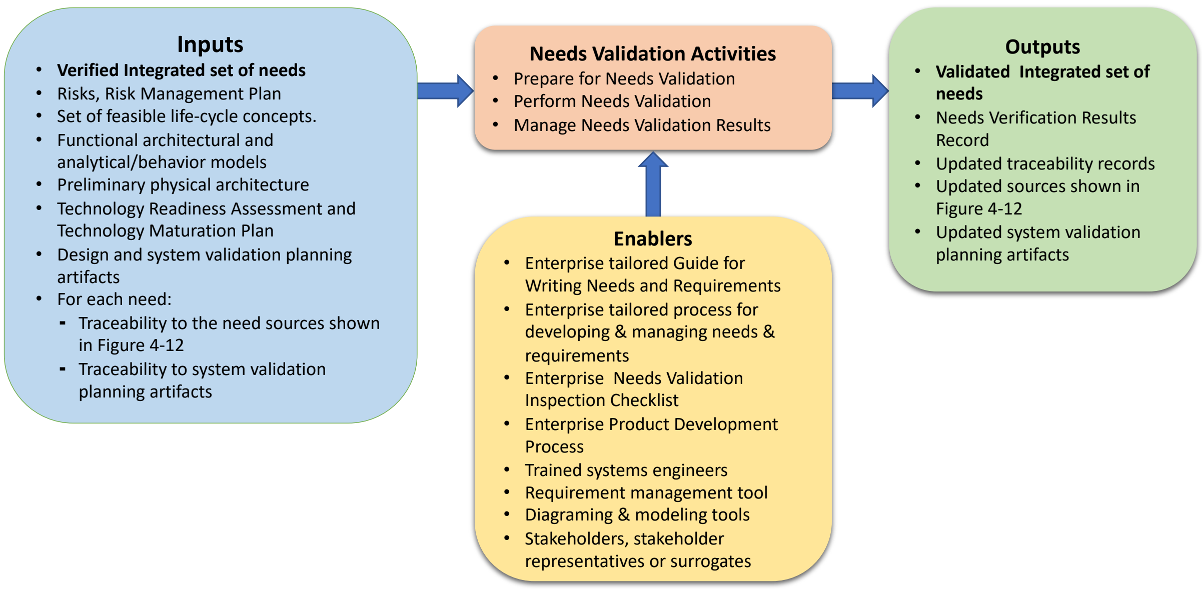



**Figure 11. Needs Validation IPO Diagram (from NRM).**

Figure 11 provides the inputs, outputs, activities and enablers for the needs validation activities from the NRM. The basic steps are:

- Plan for Needs Validation
- Perform Needs Validation
- Manage Needs Validation Results

### 2.2.3.1 Step 1: Plan for Needs Validation

The first step in planning for needs validation is to ensure the enablers from Section 2.2.1 are in place, and the project team members doing the needs validation activities have access to the inputs. Each need expression should include a trace to its source and rationale to help understand the source of the need and intent of what the need statement is intended to communicate.

Because the focus of needs validation is on content concerning what is being communicated, needs validation is difficult to automate using tools and is best approached as an analysis activity to confirm that the need expressions align with the overall set of lifecycle concepts and needs established during the System *Lifecycle Concepts and Needs Definition* activities and the associated analysis records, diagrams, and models.

A key preparation activity for needs validation is the creation of a Needs Validation Inspection Checklist. If the organization has a generic checklist, then tailor the checklist to the system being developed and the project's processes. This checklist serves as a standard to measure the needs against and will help guide the needs definition and validation activities. Having addressed each of the areas and questions within the checklist will help lead to successful completion of the lifecycle concepts and needs definition activities as well as the needs validation activities. An example Needs Validation Inspection Checklist is included in Appendix D.

#### 2.2.3.2 Step 2: Perform Needs Validation

Each need expression is validated with the stakeholders by discussing and gaining agreement on: Is this the right need? Does it communicate the stakeholders intent? Needs validation confirms the project team's understanding of the SOI to be developed is what was intended by the stakeholders.

For example, the coffee maker is a coffee maker which must have a minimum capability to brew one 12-ounce cup per minute and a maximum capability to brew four 12-ounce cups within 4 minutes—without need for the user refilling a reservoir with water—not including any user operations time. The coffee maker will have three delivery temperatures that the user may select—Warm, Hot, Very Hot—corresponding to water temperatures of 150 ±2°C, 160 ±2°C, and 170 ±2°C. (Obviously, this is only a subset of all of the needs but serves to demonstrate the activity.)

Guided by a Needs Validation Inspection Checklist (Appendix D), the project team will perform an analysis to confirm the needs are correct, the integrated set of needs are complete and correct, each need is feasible, the integrated set of needs are feasible, criteria for acceptance is defined and agreed to, and the needs associated with interfaces are established in a manner that assures the system will be able to pass system validation.

Needs validation will occur over the SOI lifecycle, first gaining agreement that all the needs have been captured and represented in the integrated set of needs. The integrated set of needs will then be used to define the set of requirements which in turn is used to design and produce the SOI.

The integrated set of needs plays an important role in the overall decision-making process. The following questions should be used in gaining stakeholder validation of the integrated set of needs:

- If the SOI is validated to meet the agreed-to integrated set of needs, will the SE team be building the right thing?

- If the project team designs the SOI to this integrated set of needs, do they have the right design to meet the needs?
- When the SE team verifies the SOI meets the requirements, will the SOI be able to be validated that it meets the stakeholder needs?

#### 2.2.3.3 Step 3: Manage Needs Validation Results

A needs validation record is created documenting the results and outputs of the needs validation activities. Defects found in the integrated set of needs as well as any defects found in any of the input artifacts or work products, must be addressed and corrected before they are baselined. Changes to any of the system validation information contained in the system validation attributes included in each system need expression should be updated in other system validation planning artifacts.

If it is identified that there is insufficient data to complete needs validation or a lack of a set of feasible lifecycle concepts from which the needs were transformed, the issue should be addressed as a risk that could cause subsequent requirement, design, and system validation to be deemed unsuccessful.

Applications within the project toolset should allow tracking of the status of the needs validation activities using a dashboard that communicates key metrics related to the needs validation results. These metrics can be obtained using the needs attributes within the requirements management tool (RMT). Making this information accessible will enable all project team members and key stakeholders to have the same view of the maturity and status of the needs validation activities. Overall approaches to managing an integrated set of needs is described in Section 4.

# 3 Requirements Definition, Verification, and Validation

This section focus is on transforming the needs into a set of design input requirements, hereafter referred to simply as "requirements". The guidance in this section is to ensure the resultant requirements are appropriately stated and correctly reflect the intent of the integrated set of needs from which they were transformed.

This section will provide practical guidance in implementing these activities, the reader is encouraged to refer to the NRM sections 6 and 7 for the specific concepts and rationale behind these activities.

## 3.1 Requirements Definition Activities

The Requirements Definition activities focus is on transforming the baselined integrated set of needs for a SOI into unique, quantitative, and measurable requirements expressed as “shall” statements. These requirements are inputs for defining the system architecture and implementing design solution that results in a set of design output specifications to which the SOI will be built or coded. Figure 12 provides the inputs, outputs, activities and enablers for the Design Input Requirements definition activities from the NRM.

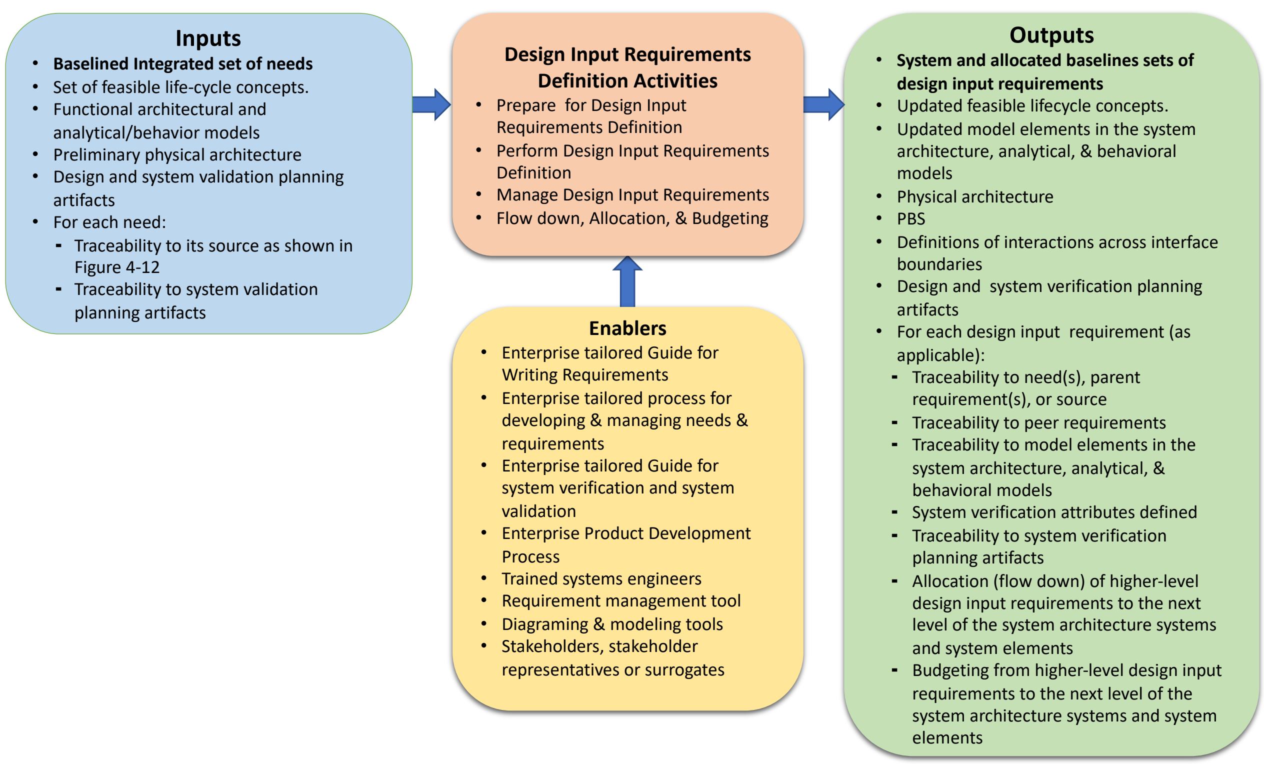



**Figure 12. Requirements Definition IPO Diagram (from NRM).**

The basic steps of the Requirement Definition activities include:

- Plan for Requirements Definition
- Perform Requirements Definition
- Manage the Requirements
- Flow down, Allocation and Budgeting of the Requirements

Note: the lifecycle technical processes are iterative and recursive. As such, the activities described in this section apply to and are repeated for each subsystem and system element within the system physical architecture.

### 3.1.1 Enabling Processes and Tools

Organizations vary in their infrastructure to support the lifecycle concept, needs, and requirements definition activities. However, the list below can be used for an organization to assess their maturity level in supporting this effort and options for growth in their supporting infrastructure. The enabling processes and tools for requirements definition, requirements verification, requirements validation, and management consist of the following (depending on the complexity needed):

- Requirements definition process tailored to the organizations workforce, culture and product line.
- Guides/work Instructions (such as the GtWR) and training material.
- Oversight management processes to ensure the activities are completed and resulting artifacts are managed and configuration controlled.
- Toolsets to develop, define, and record the requirements (such as requirements management tools, system modeling tools, performance analysis tools, etc.).
- Toolsets to manage the data.

Ideally these enabling products would be developed and managed by the organization, and this would just be another instance added to that infrastructure; these may also be found in public domain or industry standards and tailored to the needs of the organization.

### 3.1.2 Step 1: Plan for Requirements Definition

Preparing for requirements definition consists of gathering or obtaining access to the required input artifacts shown in Figure 12. It is advised to start with the integrated set of needs (discussed in Section 2), to ensure that the resultant requirements align to the set of needs for the SOI.

- It is risky to simply copy or adopt requirements from another system or project with understanding the basis for those requirements in the other project. Over time, an organization may accumulate literally hundreds of requirements as a result of such copying without validation.
- It is also risky to merely work to a project's customer requirements, as the SOI may miss fundamental capabilities to work as intended or be able to be integrated and tested as discussed earlier in Section 2. As an example, the system supplier should consider that the customer may not have specified

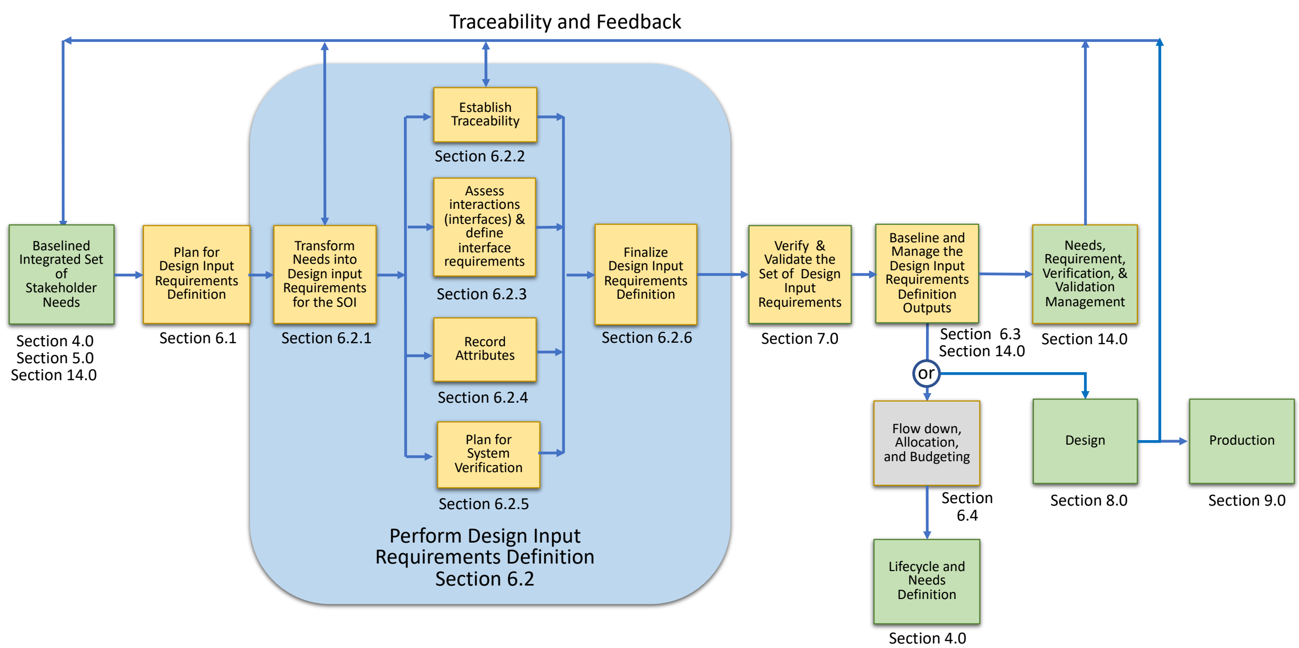
requirements for aspects such as affordability, producibility, compliance, quality, testability of their product, and compliance resulting in an incomplete set of requirements.

Another part of the planning process is to determine how the requirements will be managed, baselined, communicated, and the process for configuration management. Section 4 provides guidance on approaches for these areas, which should be in place at the start of *Requirement Definition*.

### 3.1.3 Step 2: Perform Requirements Definition

Requirements are transformed from needs and ultimately generated at the correct level of abstraction consistent to the level of a system element within the SOI's physical architecture. This section addresses the requirements definition process first, and then expands upon how the resultant requirements within the set are associated with other requirements and the needs from which they were transformed.

Requirements definition involves numerous activities as shown in Figure 13. Each activity results in data and information needed to define the requirement expressions. Note: The section references in Figure 13 address the key sections in the NRM which describe these activities in more depth.




Note: Section references are for the NRM.

**Figure 13. Requirements Definition Activities (from NRM).**

#### 3.1.3.1 Transforming Needs into Design Input Requirements

Requirement definition begins with the transformation of the set of needs into a set of requirements appropriate to the SOI that communicate "what" and not including requirements associated with the "how" design realization of the physical SOI.

While the focus of needs is from the stakeholders' perspective (e.g., the stakeholder needs the SOI to do something), the requirement statements are defined from the technical, supplier perspective of the SOI (e.g., the SOI shall do something to meet the need from which it is transformed). The project team transforms the set of needs using a requirements analysis process, asking: "What must the SOI do to fulfill each of the needs?" The answer will be one or more requirements. If the need references an allocated higher-level requirement, the answer will be one or more child requirements, that will result in the allocated requirement being met.

Areas addressed within the resulting set of requirements include functionality (what the system must do), expected performance and quality (the "how well" characteristics), the conditions of action, including triggering events, system states, interactions with other systems (interfaces), and operating environments (the "under what operating conditions").

Analyses can also address so-called "non-functional" requirements like design constraints ("make it like this"), natural and induced environments (conditions of operations, transportation, storage, maintenance), fitness for use (safety and various "ilities", e.g., reliability, testability, etc.), compliance (with standards and regulations), and physical characteristics.

Requirements analysis typically requires a domain-specific analysis method, e.g., discrete event simulation, which clearly identifies the desirable outputs in response to inputs. To be complete, the analysis must address not only the nominal inputs and outputs but also alternate and off-nominal inputs and outputs.

Additional performance-related models may be required to determine the "how well" characteristics of the functions, e.g., radio-frequency link analysis for wireless communications, reliability and maintainability calculations to ensure adequate system availability. It is high risk to attempt to address all characteristics in a single model because the model will either not address specific elements with enough definition and accuracy, and/or a "complete" model will become too complicated to complete and execute. Models used during the lifecycle and needs definition as part of needs analysis can be further elaborated as part of the requirements analysis activities.

The GtWR defines the structure and characteristics of well-formed requirement statements and sets of requirements along with a set of rules to help achieve those characteristics.

Recommended steps for transformation of needs to requirements include:

1. Review each need and evaluate how the SOI would respond to that need.
2. Draft one or more requirements that corresponds to the need.
3. Assess the requirements against the characteristics of the GtWR (is it singular, verifiable, feasible, etc. and the rules defined in the GtWR that result in those characteristics?).

4. Group the requirements into logical categories and assess for any overlaps and gaps against the needs.

5. Review the requirements with stakeholders to ensure needs are addressed and the requirement set is complete.

An approach that is helpful is to use the needs-to-requirements transformation matrix shown in Table 6. This type of matrix could be established in a document, a database, a requirement management tool, a system model, as appropriate for the approach being taken by the developing organization.

**Table 6. Example Needs to Requirements Transformation Matrix.**

| Column A<br>Needs | Column B<br>Design input requirements | Column C<br>External Interface |
|---|---|---|
| **Functional/Performance (Function)** | | |
| Need 1 | Rqmt 1 (could be more than one) | Interface (if applicable) |
| Need 2 | Rqmt 2 (could be more than one) | Interface (if applicable) |
| **Operational (Fit)** | | |
| Need 3 | Rqmt 3 (could be more than one) | Interface (if applicable) |
| Need 4 | Rqmt 4 (could be more than one) | Interface (if applicable) |
| **Physical Characteristics (Form)** | | |
| Need 5 | Rqmt 5 (could be more than one) | Interface (if applicable) |
| Need 6 | Rqmt 6 (could be more than one) | Interface (if applicable) |
| **Quality (-ilities)** | | |
| Need 7 | Rqmt 7 (could be more than one) | ………. |
| Need 8 | Rqmt 8 (could be more than one) | ………. |
| **Standards/Regulations (Compliance)** | | |
| Need 9 | Rqmt 9 (could be more than one) | ……… |
| Need 10 | Rqmt 10 (could be more than one) | ……… |

Using the needs-to-requirements transformation matrix, Column A consists of need statements written from the stakeholder's perspective and Column B reflects requirements communicating what the system must do such that the needs in Column A will be met. Each need statement in Column A, will be transformed into one or more well-formed requirement statement(s) that are written at the level of abstraction appropriate for the level of architecture the SOI exists and to which the design input requirement is written.

If a need in Column A involves a standard or regulation, multiple requirements will need to be addressed as discussed below. The resulting requirements are the project team's response to the requirements in the standard or regulation whose implementation will result in compliance with the standard or regulation.

Column C is used to record the fact that the SOI needs to interact with another system to implement the requirement(s) stated in Column B. Entries in Column C will be used to validate the context diagrams and models developed during the lifecycle concept analysis and maturation activities discussed in Section 2. If Column C indicates the requirement in Column B is an interface requirement, then the Column B interface requirement will need to address the specific interaction between the SOI and the external system and include a reference (pointer) to where that interaction is defined. The interaction will need to be defined (ICD, data dictionary, etc.) as discussed in NRM Section 6.2.3.

An example of transformation of a need to resulting requirements is shown Table 7.

**Table 7. Example Coffee Maker Need to Requirements Transformation.**

| Column A | Column B | Column C |
|---|---|---|
| Needs | Design input requirements | External Interface |
| **Functional/Performance (Function)** | | |
| The stakeholders need the coffee maker to have three temperature settings (warm, hot, extra hot) for the water temperature and control the temperature to within 2 C. | The coffee maker shall have three selectable settings for coffee brewing: Warm, Hot, Extra Hot.<br><br>The coffee maker shall heat water to within +/- 2degrees C, of the selected temperature setting listed below, after the operator selects one of the temperature settings: Warm: 96°C, Hot: 100°C, Extra Hot: 105°C. Rationale: Values are resulting from analysis associated with consumer research surveys and safety regulations.<br><br>The coffee maker shall heat water from an initial temperature of 13°C +15°C/- 10°C to the selected brewing temperature within 1 minute. Rationale: Values are consistent with tap water temperature ranges.<br><br>NOTE: This represents only a subset of requirements, there would be additional requirements related to final coffee temperature, quality of water, definition of local environments, etc. | Water Source<br>Operator |

Requirements can often be categorized into different types:

- **Functional / Performance** - Functional requirements and their associated performance characteristics are unique and are the heart of what actions and capabilities the stakeholders need the SOI to provide. Each function will have one or more functional/performance requirements that address performance characteristics such as how well, how many, how fast, etc. a function needs to be performed.

  When documenting performance requirements make sure they trace to the function they apply to. Note: from a system verification perspective, each function must be expressed as a family of requirements, each stating the function and a performance characteristic. If a function has multiple performance characteristics that need to be stated, then there will be a function/performance requirement for each characteristic. A rule of thumb for functional/performance requirements:

    - If the function deals with a primary function, purpose, or capability of the system, put it under the Functional/Performance category. This would include functional/performance requirements dealing with a core function or capability of the SOI. These requirements would include those that were transformed from a need for functions that have the attribute “critical” defined.
    - If the function deals with a secondary concern during operations to accomplish the primary functions, document the requirement in the operational section. This would include functional/performance requirements dealing with enabling functions – functions that enable the primary functions to be performed. For example, a function that involves installation or maintenance but does not have anything to do with a primary purpose of the system, would be documented as a Fit/Operational requirement.

**Example:** The coffee maker shall heat water from ambient to 100+/1 °C in less than or equal to one minute.

- **Fit / Operational** - Fit/operational design input requirements deal with the ability of the system to operate within its operational environment by its intended users. “Fit” includes human system interactions and interfaces as well as both the induced and natural environments enabling the SOI to become an integral part of the macro system it is a part of. Some organizations classify “fit” requirements as “Operational Requirements” given they relate to the external operational environment, however this can be misleading as this may omit aspects of a lifecycle beyond operation (such as transportation, storage, facilities, personnel interface, maintenance, and logistics).

**Example:** The coffee maker shall operate within external environmental temperatures between 0°C to 38°C.

- **Form** – Form design input requirements address the physical characteristic of the SOI; they address the shape, size, dimensions, mass, weight, and other physical attributes that uniquely distinguish the SOI. For software, “form” requirements may address number of lines of code, memory requirements, programming language, etc. “Form” requirements address SOI characteristics associated with the physical system that are not always included within functional, architectural, or analytical/behavioral models.

**Example:** When in the operational configuration, the coffee maker shall fit within an envelope 31 cm wide, 31 cm deep, and 146 cm high.

- **Quality** - Quality design input requirements address “fitness for use” (safety, security, and various quality “ilities”, e.g., reliability, testability, operability, availability, maintainability, operability, supportability, manufacturability, interoperability, safety, security, to name a few.) These are domain specific, and missing or incorrect quality requirements are often the cause of failed system validation, product recalls, warranty work, and negative reviews on social media. Each organization needs to define the types of quality requirements they will include within the set of design input requirements based on the domain and expectations of their customers and needs of other stakeholders and included in the checklist and template for design input requirements to ensure that all are addressed. Each will result in unique requirements for the SOI.

> **Example:** The coffee maker shall have an operational life of greater than or equal to 10,000 hours.

- **Compliance** - Compliance requirements address the design and construction standards and regulations the project team must show compliance. For many systems, applicable standards and regulations can represent a substantial portion of the design input requirements. Compliance with standards and regulations are drivers and constraints that come directly from the stakeholder elicitation activities. Many standards and regulations contain at least three types of requirements:
    - Process requirements on the developer/designer.
    - Technical requirements (design input requirements as well as design output specifications) on the SOI itself.
    - Requirements dealing with system verification and system validation activities.

  It is important that the process requirements be addressed in project plans, SOWs, or agreements directing the project team or supplier; 1) the SOI technical requirements that are invoked in set of requirements, 2) the applicable verification/validation/test planning artifacts, and 3) system verification and system validation procedure requirements. Important: Do not mix these three types of requirements in the SOI set of design input requirements.

  Care needs to be taken as many standards do not have well-written requirements that are verifiable, while others are industry "standards" which are well understood with respect to implementation (and proof). Because of this, the project team should not blindly copy and paste individual requirements from applicable standards and regulations into their requirement set; nor should they invoke requirements within a standard or regulation without first assessing applicability and ability for the standard requirement(s) to be understood, implemented, and the SOI verified to meet them.

> **Example:** The coffee maker shall comply with the requirements in Section xyz of the UL 1082 standard

Additional considerations in requirement definition:

**Design Input Requirements versus Design Output Specifications:** Design input requirements address "what/design to", where the design output specifications address "how/build-to/code-to". The design output specifications are typically provided in the format of engineering drawings, models, instructions and procedures. A good visualization is provided in Figure 14, which shows a "line" between the needs and design input requirement activities and the design activities.

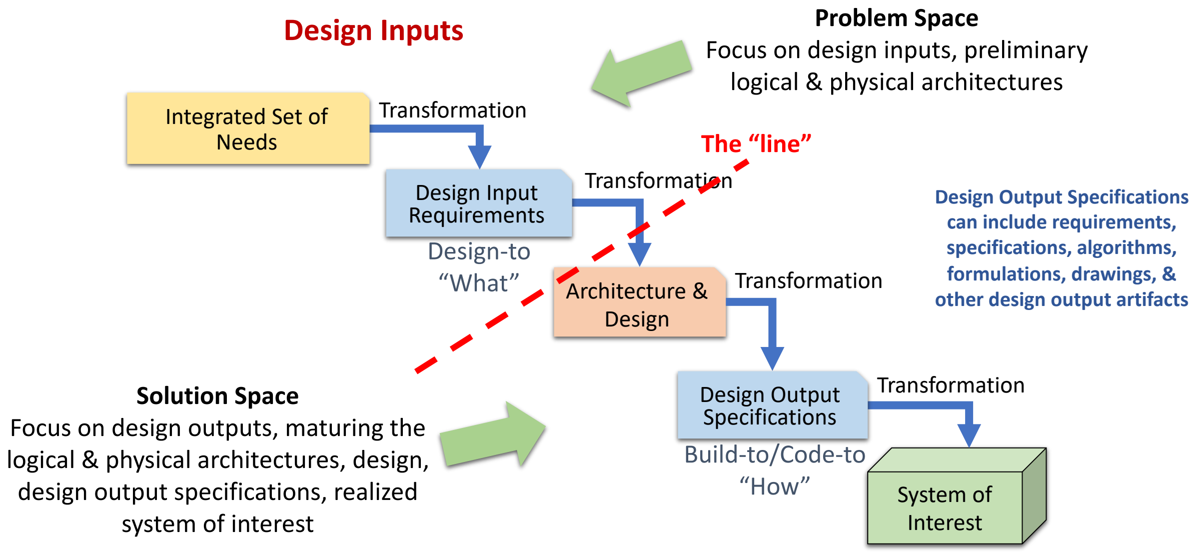



**Figure 14. The "Line" Between Design Inputs and Design Outputs (from NRM).**

Design input requirements for the system flow down to subsystems and system elements at the next level of the system architecture for further decomposition and elaboration. For system elements that need no further decomposition to be realized by the Acquisition Process or Design Definition Process (buy, build, code, or reuse), the line shows the boundary between design inputs and design outputs.

**Tolerances and Ranges:** There are several considerations when defining tolerances and ranges within a requirement statement. First, is if the tolerance or range is too "tight" or the range too "narrow" the cost to design and manufacture a system that meets that tolerance or range will be more expensive, as well as to successfully verify the system meets the tolerance or range. It is important to plan ahead to design, manufacturing, and system verification when stating tolerances and ranges in a requirement statement.

Second, if the tolerance is too "loose" or range is too "wide", the possible variability during manufacturing may result in an issue with "tolerance stack up". This is when the combination of +/- tolerances or allowable ranges may result in a system that does not work. For example, there may be a performance requirement for the overall system that has been allocated to several parts at the next level of the architecture. At the system level the performance requirement has a range of acceptable values, each of the child requirements at the next level will also have to have a range specified; each child requirement may be further allocated to parts and components, each deriving their own requirement with a range. If these parts and components are designed and manufactured independently, there is a high likelihood that when integrated together the combination of +/- ranges will result in the parent requirement failing to perform within its required range. For mechanical parts with tolerances, the end result may be where two parts that need to be mechanically attached will have mechanical interfaces that do not align properly.

**Accuracy and Precision**. For requirements that deal with taking a measure, the concepts of accuracy and precision (or imprecision) must be considered as a way to express expected performance. While similar to tolerance, accuracy and precision are different concepts.

Accuracy is a measure of how close an average of measures is to a baseline value expressed as a percent. Precision is a function of the distribution of measurements taken, i.e., closeness of agreement between independent, repeated measures obtained from the same sample under specific conditions. The closer each measurement is to other measures, the more precise the measuring system.

Precision is represented by the standard deviation (in units of the test) or coefficient of variation (in units of percent). Imprecision (random error or random variation) represents a lack of repeatability or reproducibility of the same result; represented by the standard deviation (in units of the test) or coefficient of variation (in units of percent).

As for tolerances, there are several important considerations. First, if the accuracy or precision values are too “tight” or “narrow” the cost to design and manufacture a system that meets those performance values will be more expensive, as will the cost to verify that the system meets those values. It is important to plan ahead to design, manufacturing, and system verification when stating requirements that deal with accuracy and precision. Second, is the expectations for accuracy and precision over the life of the system as discussed below.

**System Lifetime and Expected Performance**: It is common there will be requirements for performance stated separately from the quality requirements (-ilities) – including a lifetime requirement. An issue that should be addressed when defining the requirement is whether the performance or quality value in the requirement statement applies to beginning of life (BOL) performance of the system or end of life (EOL) expected performance. For example, the customers would expect the accuracy and precision requirements for an instrument to apply to both BOL and EOL. Degradation is almost a certainty in most systems. In some cases, some degradation may be allowed in the system performance. If so, there should be a requirement stating the amount of allowed degradation over the required life of the system. This could result in a system whose performance is better than stated in the requirement for BOL in order to meet the EOL requirement. In some cases, it may be best to state requirements for both BOL and EOL. If there is no degradation requirement and the expectation for performance is EOL, this must be stated clearly so the project team can make allowances during design for natural degradation within the system components so the system verification and validation success criteria can be met. For consideration, when there are requirements for EOL performance, do these requirements assume preventative maintenance?

There is also what is referred to as design lifetime versus operational lifetime. Consumer product warranties are based on the design life. There are many cases when the actual operational lifetime is greater than the design life (e.g., rovers on the surface of Mars, cars that are driven 200,000 miles); there are also examples for some consumer products where the operational life is very close to the design life. Warranties for consumer products are often based on some lifetime metric.

### 3.1.4 Step 3: Manage the Requirements

Organizations need to define a specific approach to organize and manage their requirements best suited to their processes, toolset, culture, and product line – especially concerning how they will address requirements within design and construction standards. This approach should be documented in their organizational needs and requirements definition and management process documentation, work instructions, and the project’s Systems Engineering Management Plan (SEMP). It is useful to include a template or outline showing the specific method of organization expected. This template/outline also serves as a checklist to help ensure all the various categories of requirements have been addressed.

The SOI requirements can be recorded and managed via a document, database, or model-based approach. The chosen method, in addition to representing the requirement statement, should allow the assignment of:

- Requirement identifier.
- Attributes, such as rationale, priority, type, etc.
- Traces as discussed earlier.
- Approval status information and date of approval.

Changes to requirements must also be managed. Pending changes can be communicated through the chosen approach or via the communication channels earlier agreed to. Using a data-centric approach traceability can be established within the project toolset and the traceability matrices can be represented as reports generated by the toolset. Traceability is key to managing change within the sets of requirements and between sets of requirements when a change to a requirement could impact requirements that requirement is traced to. Approaches to managing a set of requirements is described in Section 4.

### 3.1.5 Step 4: Flow Down, Allocation and Budgeting of the Requirements

#### 3.1.5.1 Allocation, Budgets and Margin

System, subsystem, and system element lifecycle concepts, needs, and requirements sets are highly interrelated and dependent. These relationships and dependencies must be considered to ensure the integrated and verified system can be validated to meet its integrated set of needs.

A single need or requirement on its own nor a single set of lifecycle concepts, needs, and requirements for a subsystem or system element that is part of the SOI on its own would not achieve this (nor would standalone models of the subsystems or system elements within the SOI architecture). This family is often shown notionally in a tree diagram, such as the one shown in Figure 15. The result is an integrated family of lifecycle concepts, integrated sets of needs, and requirements sets that exist at multiple levels of the system architecture, linked together to form a data and information model of the integrated system.

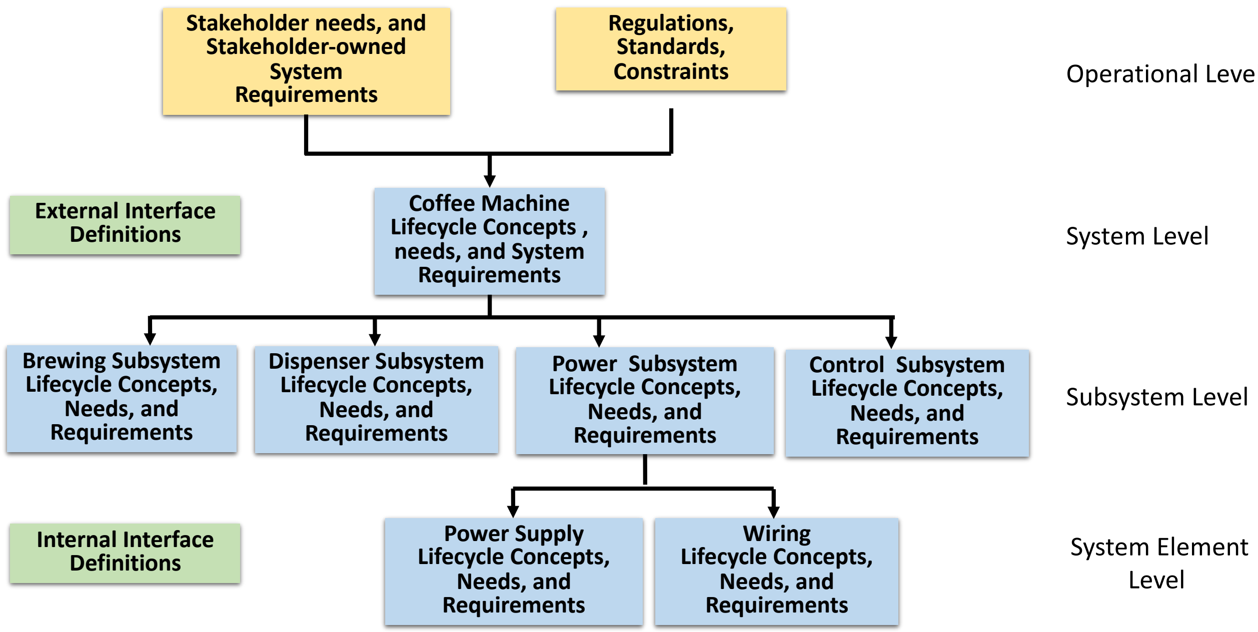



**Figure 15. Requirement Tree Diagram Example for a Coffee Machine (from NRM).**

**Allocation** is the process by which the design input requirements defined for an entity at one level of the physical architecture are assigned (flow down) to the entities at the next lower level of the architecture that have a role in the implementation of the allocated requirement as shown in Figure 16.

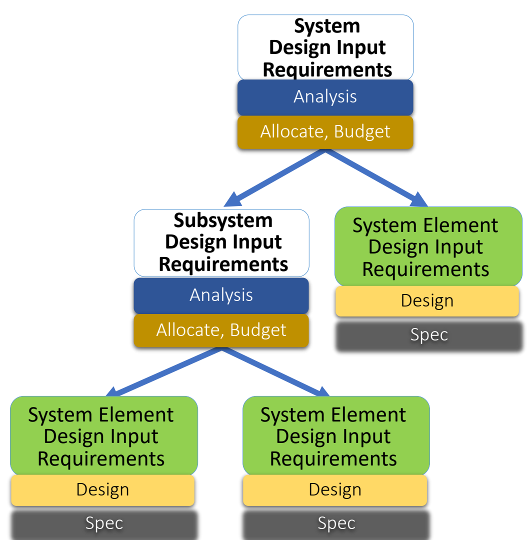



**Figure 16. Flow Down of Requirements via Allocation and Budgeting (from NRM).**

As part of this analysis, the project team determines what "role", if any, each subsystem and system element at the next level of the architecture has in the implementation of each design input requirement being allocated. For each subsystem or system element that has a role, the requirement will be allocated to those subsystems or system elements. The resulting integrated set of needs are transformed into a set of design input requirements for the subsystem or system element the higher-level parent requirements were allocated to.

A different approach to system engineering architectures may be needed to accommodate software-intensive systems. Rather than developing subsystems that contain software, the view should change that we are developing software-intensive systems where hardware and mechanical systems are enabling systems for the software. In this context, the software is “constrained” by the hardware. The mechanical systems hold the hardware together providing structure and the “skin” of the system. An alternate approach for allocation in this scenario is to first divide the system into hardware, mechanical, and software, and then decompose the hardware, mechanical, and software systems into lower-level subsystems and system elements.

There are two types of allocation:

1. Allocating responsibility, where the receiving entity has some role in meeting the intent of the allocated parent requirements.
2. Allocating some quantity, such as resource production or utilization, performance, quality, or some physical attribute.

Physical attributes include mass, volume, etc. Performance is associated with functional requirements in terms of how well, how fast, how many, etc. For example, accuracy, precision, time, bandwidth, consumption of a consumable, or power use.

**Budgets** are created to address allocation of quantity that need to be managed and controlled at the system level. In a data-centric practice of SE, budgets are managed and controlled within a model of the system. In a document-centric practice of SE, budgets are commonly managed and controlled either within the RMT or a spreadsheet referenced by the requirements.

A critical concept associated with budgeting is that the budgeted quantities result in requirements that have a dependency - a change in one will result in the need to change another. Because of these dependencies, establishing traceability between the child requirements and their allocated parent as well as between peer requirements is critical. The RMT being used should allow management of the allocations to the entities separately from the traceability. These relationships are shown in Figure 17.

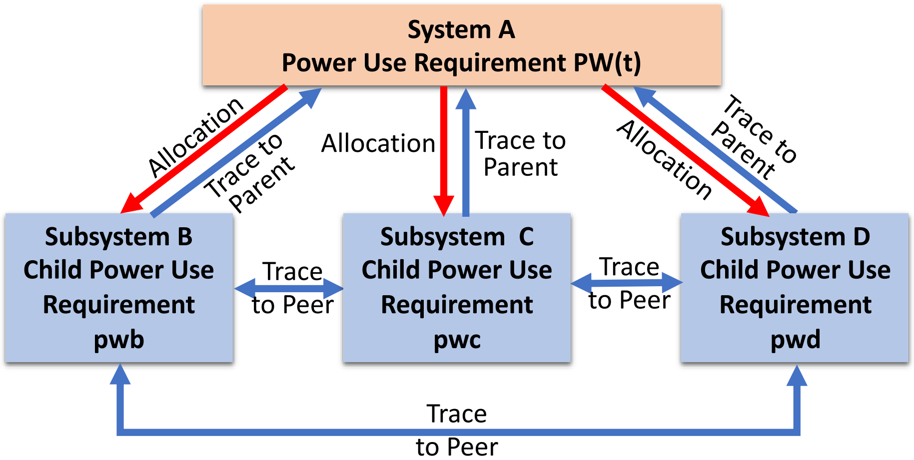



**Figure 17. Allocation and Budgeting Example (from NRM).**

Because of the dependencies, it is useful to view allocation of resources as an equation.

$$\text{Sys A PW(t)} = \text{SSB (pwb)} + \text{SSC (pwc)} + \text{SSD (pwd)}$$

where PW(t) is the total power available for use, SS = subsystem, and pwx = the allocated power use values. In this context, changes to any variables on the right side of the equation will require a change in the other variables to keep the equation balanced. The project team can use either models or spreadsheets to establish these equations. However, each of the subsystems may also allocate/budget their values to the system elements that make up the subsystem, so now there are second order allocations/budgets. Each of the system elements at the lower level may sub allocate/budget their values, resulting in third order allocations/budgeting. There will be cases were there may be a dependency between second or third order allocated/budget values across subsystems.

Using analytical and behavioral models, the project team can do some optimization. However, the budgets are often dynamic as the design progresses for each system element. Some system elements may need less than what was budgeted and other more. These changes have a ripple effect for all dependent budgets up and down the levels of the system architecture. Because of this, it is critical for the project team to manage all budgets from the top using a single integrated model of the SOI.

Due to the dynamic nature of budgeted qualities, many organizations do not include the actual budgeted values within the requirement statements. The budgets are managed separately in a configuration managed form (database, spreadsheet, model). With this approach, the individual design input requirements then include a pointer to this source.

Given there is uncertainty with the budgets, there is inherent risk to the project being able to stay within the allocated budgeted values. One way to help manage those risk is the use of margins and reserves; the different types of margin are highlighted below:

- Development margin or technical margin is defined as the difference between the estimated budgeted value and the actual value at the end of development when the system is delivered. Margins allow for both expected and unexpected change as the design matures over the system development lifecycle. Development margins for resources like mass, power, and time or margins for performance like, accuracy, precision, or rate are defined at the system level and allocated to the parts of the system architecture.

- Operational margin is defined as the difference between what is required during operations and what is actually provided. Operational margin provides additional capability to address unexpected changes, anomalies, security issues, or errors in defining the expected operational environment that may occur during operations. (The need to address operational margins and issues is discussed in several different subject areas including agile systems and resilient systems. The reader is encouraged to explore each of these areas in more debt depending on the needs of the project and customer.)
- Management Reserve is defined to be the portion of the available quantity held back or kept “in reserve” by management or the quantity owner during development and not made available through allocation. Reserves allow management to deal with the unexpected events such as out of-scope demands, unplanned changes, uncertainties as to what is feasible, and other uncertainties.

A key challenge is how to manage budgeted items over the development and operations life of the SOI. At the beginning of a project, it is common to do a preliminary analysis and compute an estimate of the various needs for the quantity of something or needed performance and then communicate these estimates as design input requirements.

A key issue in managing the allocation/budgets during both development and operations is a failure to understand that these initial computations are only approximations based on assumptions that may or may not be correct and may change as the system design matures. Budgeted values should be treated as TBR values (discussed in Section 4.1.2).

#### 3.1.5.2 Traceability

The system and each of the subsystems and system elements within the system physical architecture are represented by a set of requirements and by a set of design output specifications (refer to Figure 3). These relationships are documented via "links" that allow the relationships to be traced between the entities that are linked, this is the concept of **traceability**. Traceability helps establish that the set of requirements has the characteristics defined in the GtWR: C1- Necessary, C10 – Complete, and C11 - Consistent.

Traceability provides the ability to track needs and requirements across the lifecycle from their origin to the activities and deliverables that satisfy them. Traceability can be “bidirectional”, “forward and backward”, or “up and down.” Traceability can be horizontal across the lifecycle as well as vertical between levels of the architecture.

Requirements can have various types of traceability, including:

- **Parent/child** - shows connection of the higher-level allocated requirement that sponsored the need for one or more child requirements.
- *Source* - shows connection to where requirement content derived from (such as a constraint, standard, regulation, MGO, measures, lifecycle concept, risk, a model, or analysis).
- **Allocation** – a trace to the subsystems or system elements a parent requirement is allocated or budgeted to. Note: this is not a requirement-to-requirement link, but a requirement to a logical/physical subsystem or system element, for which child requirements will be defined and then trace back to the allocated parent requirement.
- **Peer** - displays relationships among requirements at the same level, or within the same set. The relationship could be general or grouped by topic (such as connecting requirements of a particular theme, connecting all interface requirements, requirements sharing a common budgeted quantity, connecting all performance requirements relating to a common function, etc.).
- **Interface definition** – provides traceability between an interface requirement and the agreed to definition concerning the interaction across an interface boundary between the SOI and another entity.
- **Dependency** - shows a relationship of a requirement to items it needs to have completed in order to be satisfied (can be other requirements, or representation of specific conditions) or a relationship to another requirement such that a change in one will result in the need to change the other. The other requirement could be in a different set of requirements for a different subsystem or system element.
- **Design** – shows a relationship between a requirement and its design implementation and resulting design output specifications.
- **System Verification -** displays relationship of requirements to verification artifacts or activities that will provide evidence of requirement satisfaction.
- **System Validation** - displays relationship of needs to validation artifacts or activities that will provide evidence of need satisfaction.
- **Model entity** – a trace between a need or requirement and an entity within a model. For example, a functional requirement would be linked to a function within a functional model; a performance characteristic of a function would be linked to that function within the model.

This is not an exhaustive list but shows the kind of traceability that can be established between requirements within a set of requirements, requirements in other sets of requirements, and other artifacts within the project's integrated dataset (needs, models, diagrams, system verification planning artifacts, etc.). Some of this traceability can be supported by use of attributes (such as A2 - Trace to Parent**,** A3 - Trace to Source, A5 – Allocation/budgeting, A32 – Trace to Interface Definition, and A33 – Trace to Peer Requirements, and A37 – Risk (mitigation)) defined in the NRM Section 15.

For complex systems consisting of multiple subsystems and system elements, each with their own sets of requirements, establishing, maintaining, and managing all the traces can be difficult to be effectively done manually using standard office applications. This is one of the reasons toolsets have been developed, enabling traces to be established and managed within a relational database. With the move towards a data-centric approach to managing project data, and the increased use of models, applications have been developed to allow not only traceability within sets of requirements but traceability between requirements and all other project artifacts across all lifecycle stages. Some RMTs enable the user to define a set of traceability rules within the tool. The RMT then can help the user to both establish traceability per the rules, but also alert the user when a rule is not being followed.

### *Parent and Child Trace*

It is a common convention for needs and higher requirements to have "child" requirements (lower-level requirements) link to parent requirements (higher level requirements) as these are often created after the parent requirements are generated. An example of a parent - child trace that can be established from linking is shown in Figure 18. Note that a lower-level requirement may not always have a trace to a parent, it may instead have a different trace, such as to a source (e.g., need, constraint, interface, convention, etc.) which leads to it being considered a derived requirement. All requirements must trace to a source or parent, there should be no "orphan" requirements. All requirements must have a purpose (they are *necessary* as defined in the GtWR) to ensure they add value; a trace to a parent or source is one method to show a requirement is necessary. Another aid to help establish the characteristic *necessary* is the rationale that must be defined for each requirement.

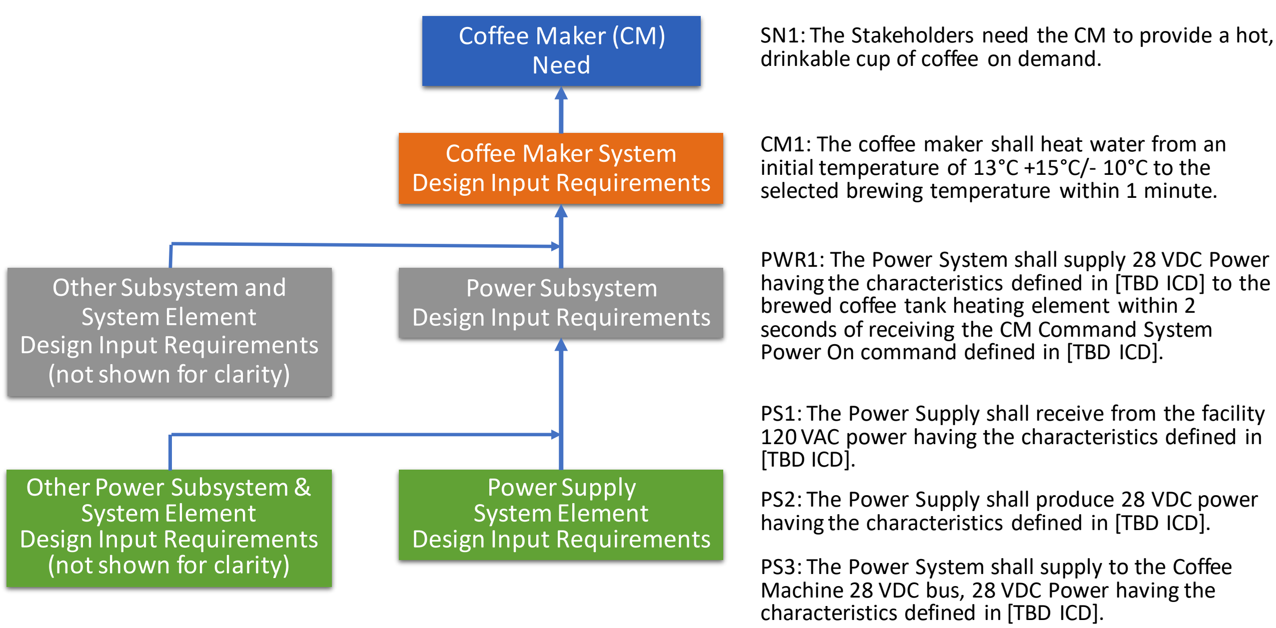



**Figure 18. Example Requirement Parent / Child Trace (from NRM).**

In Figure 18 the Power Supply system element requirements are traced to their allocated parent Power Subsystem requirement, which is traced to its allocated parent system requirement, which is traced to the system need from which it was transformed.

These parent/child relationships can be established and maintained in the toolset as a set of links from the lower-level requirements to the higher-level requirements. In a data-centric practice and toolset, the system need can be easily traced to its source.

Once these links are established, bidirectional traceability exists from the system need to its transformed system requirement, from the system requirement to its child requirements in the Power Subsystem as well as any other child requirements in other subsystems the parent system requirement was allocated. The parent Power Subsystem requirement can be traced to its child requirements in the Power Supply system element as well as any other child requirements in other system elements the parent Power Subsystem requirement was allocated.

In a data-centric practice the system element requirements can be traced to the design output artifacts. All the requirements would also trace to the models from which they were derived as well as trace to all the system verification artifacts that were developed when the requirements were written. All these links are part of the traceability records.

Traceability links are created concurrently with the transformation of needs into requirements and continues as lower levels of subsystem and system element needs and requirements are defined. It is important that when a need or requirement is generated that the associated trace attributes and linkages are be captured in the toolset traceability record; going back and trying to establish traceability after the fact can be difficult.

***Usage of a Trace Matrix***

Being able to visualize the parent/child relationships that result from traceability can be a challenge. A "Trace Matrix" is a common visualization of the traceability record showing the above relationships in tabular form. The display of a trace matrix supports the analysis needed to show that each system need has been transformed into one or more requirements, that each requirement is needed by linking it to the system need it was transformed, a parent, or source as well as showing the requirement's implementation at the next level of the system architecture by displaying its child requirements for each system or system element it was allocated to.

From the example in Figure 18 a trace matrix can be created to show these relationships in a tabular format as shown in Table 8. Many toolsets have the capability of generating a trace matrix automatically based on the information in the traceability records within the project's integrated dataset, such as the example shown in Figure 19.

**Table 8. Example Trace Matrix.**

| System Need | System Requirement | Subsystem Requirement | System Element Requirement |
|---|---|---|---|
| SN1: The Stakeholders need the CM to provide a hot, drinkable cup of coffee on demand. | CM1: The coffee maker shall heat water from an initial temperature of 13°C +15°C/- 10°C to the selected brewing temperature within 1 minute. | PWR1: Upon receipt of the CM Command System Power_On command defined in [TBD ICD], the Power System shall supply 28+/-2 VDC Power to the brewed coffee tank heating element in less than or equal to 2 seconds.<br><br>[Additional Subsystem Requirements would be shown below] | PS1: The Power Supply shall receive facility120 VAC power having the characteristics defined in [TBD ICD].<br><br>PS2: The Power Supply shall produce 28 VDC power having the characteristics defined in [TBD] ICD.<br><br>PS3: The Power System shall supply to the Coffee Machine 28 VDC bus, 28 VDC Power having the characteristics defined in [TBD ICD]. |

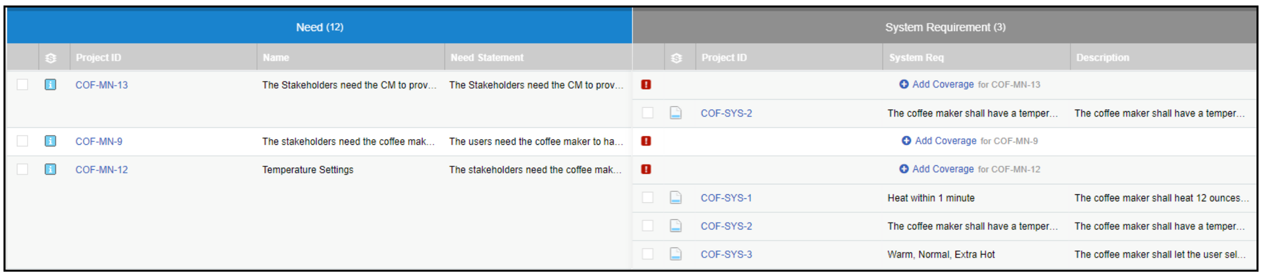



**Figure 19. Example Trace Matrix in a Tool.**

The trace matrix provides a visualization of the traceability record that can be used to manage needs and requirements throughout the system lifecycle. The trace matrix also provides a structure as an aid to help assess and manage changes, as well as assessing consistency and completeness of the requirement set and artifacts the requirements are linked to.

***Using Traces to Support System Verification***

System verification information contained in the verification attributes included in each of the requirement expressions should be traced to the System Verification planning artifacts, such as the System Verification Matrix (SVM). This approach enables system verification planning (such as test and analysis plans). The system verification process is described further in the NRM and the GtVV.

## 3.2 Requirement (Expression) Verification Activities

Requirements verification activities review the requirement expressions and the set of requirements to verify they have the expected characteristics and adhere to the rules that result in those characteristics as defined in the INCOSE GtWR or the developing and customer organization's Guide to Writing Requirements. In terms of verification, the guide represents the organizational requirements for writing well-formed requirements. The requirements verification process will also include inspection of traceability records from the sources from which the requirements were transformed to ensure traceability with those sources as well as an audit of the interface requirements. (Refer to the NRM Section 6.2.3.6for a detailed discussion on performing an interface requirement audit.)

Figure 20 provides the inputs, outputs, activities and enablers for the requirements verification activities from the NRM. The basic steps are:

- Plan for Requirements Verification.
- Perform Requirements Verification.
- Manage Requirements Verification Results.

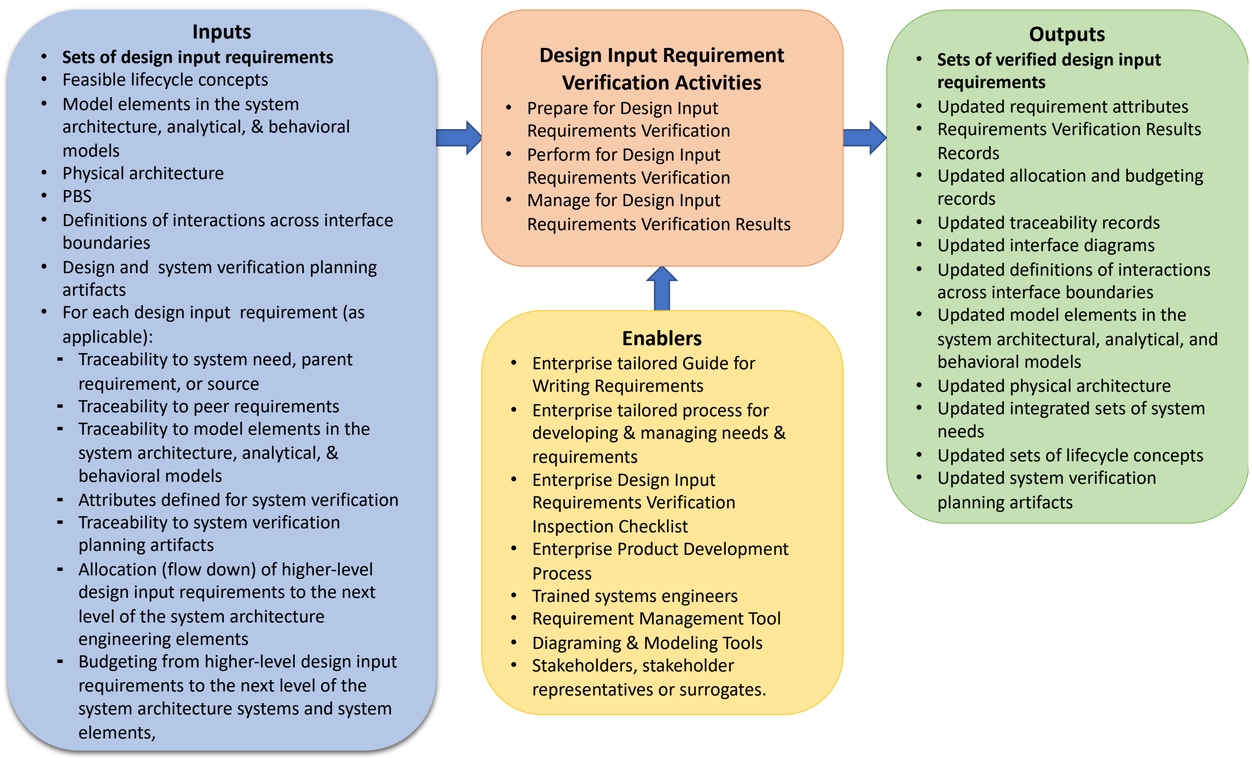



**Figure 20. Requirements Verification IPO Diagram (from NRM).**

### 3.2.1 Enabling Processes and Tools

Enabling processes to support this effort are highlighted below. Organizations vary in their infrastructure to support the lifecycle concepts, needs and requirements definition processes. However, the list below can be used for an organization to assess their maturity level in supporting this effort and options for growth in their supporting infrastructure. The enabling processes and tools for requirements definition, requirements verification, requirements validation, and management consist of the following (depending on the complexity needed):

- Enterprise tailored Guide to Writing Requirements (based on the INCOSE GtWR).
- Enterprise tailored process for developing and managing needs and requirements.
- Tailored checklist for requirements verification criteria and tasks (see examples in Appendix D).
- Enterprise process for product development.
- Trained systems engineers.
- Requirements Management Tool (RMT).
- Diagramming and Modeling tool(s).
- Stakeholder representation.

### 3.2.2 Step 1: Plan for Requirements Verification

Assuming the enablers are in place and tools are in use, the artifacts listed as inputs in Figure 20 should have been produced and matured during the requirement definition activities to the point where they are ready for the requirements' verification activities. If using a data-centric approach then these artifacts should exist within the project toolset.

While some of the requirements verification activities must be done manually, others may be able to be automated depending on the capabilities of the applications in the project toolset. Natural Language Processing (NLP) applications provide a capability that allow requirement verification to be automated to some extent by automatically checking syntax to conventions. Many of these tools use the characteristics and rules defined in the INCOSE GtWR as a basis for assessing the quality of the requirement statements and sets of requirements. These tools can be used as both a "digital assistant" to aid in the writing requirement statements as well as assess the quality of individual requirement statements and the of the set of requirements. To what extent the project can rely on automated requirement verification depends on a tradeoff between effort and risk. The need for and importance of using automation for requirement verification will increase as systems become more complex and the number of requirements increases.

Using a data-centric approach, completeness and consistency of allocation and traceability can be established within the project toolset, and trace matrices can be reports generated by the tool (refer to Table 8 for an example of a trace matrix). With a well-defined allocation and traceability strategy and requirement attribute strategy, requirement verification in terms of allocation and traceability can be aided greatly using these tools. At a minimum, these tools can generate reports that identify missing traces, requirements that have not been allocated, and attributes not defined. For example, a report could be generated to list all requirements that do not trace to a source, a report that lists all needs that do not have implementing requirements, or all allocated higher-level requirements that do not have child requirements.

Note: While applications within the project toolset can produce allocation and trace matrices or reports, these applications cannot yet determine the accuracy and correctness of the allocations or traces between entities. This assessment must be done manually.

A key preparation activity for requirement verification is the creation of a Requirements Verification Inspection Checklist. This checklist serves as a standard to measure the requirements against and will help guide all the requirement verification activities. Having addressed each of the areas and questions within the checklist will lead to successful completion of the requirements definition and verification activities. The GtWR can be used as a starting point for the organization to develop a checklist defining the specific checks that should be performed on a requirement statement that is reflecting the priorities of the business (An example Requirements Verification Inspection Checklist is included in Appendix D).

### 3.2.3 Step 2: Perform Requirement Verification

There are several activities that should be performed to ensure the individual requirements expressions and integrated set of requirements are well-formed and provide objective evidence they have been verified. Using the Requirements Verification Inspection Checklist as a guide, the project team should:

1. Manually verify that individual requirements expressions and the sets of requirements have the characteristics per the rules defined in the INCOSE GtWR or similar guide. This could be done by using the Requirements Verification Inspection Checklist as a guide to inspect each requirement statement by individuals or as part of a tabletop or peer review. Given the number of characteristics, the number of rules to help ensure the requirements have those characteristics, this task is difficult to do manually especially for large sets of needs.
2. Verify that all terms used within the requirement statements are consistent with the architectural model, project glossary, and data dictionary.
3. Alternately, if in the project toolset, use an Natural Language Processor (NLP) application that provides the capability to automate the verification of the requirements statements in terms of how well they adhere to the rules for writing requirements and sets of requirements as well as checking for consistent use of terminology.
    - For requirements statements with defects, members of the project team will need to examine each defective requirement statement and fix the defects the NLP application identified.
4. Verify that individual requirements expressions have the set of attributes agreed to by the project team defined. The project toolset should be able to produce a report concerning whether any of the attributes are ‘null’- that is, no values have been defined for a given attribute. While a report can tell if an attribute has been defined, it cannot assess the quality or accuracy of the information in the attribute – that assessment will have to be done manually. For example, does the text in the rationale statement include the information expected to be in the rationale?

Does the rationale state why the requirement is necessary?  Is the source of any performance values explained?  This activity helps establish each requirement expression has the characteristic defined in the GtWR C4 – Complete.

5. Verify that individual requirements expressions have the set of system verification attributes agreed to by the project team defined.  The project toolset should be able to produce a report concerning whether any of the system verification attributes are ‘null’- that is, no values have been defined.  While a report can tell if a system verification statement has been defined, it cannot assess the quality or accuracy of the information in the statement– that assessment will have to be done manually.  For example, does the text in the verification statement include the information expected?  This activity helps establish each requirement expression has the characteristic C4 – Complete.
6. Use the project toolset to generate trace matrices to verify each requirement traces to the system need, an allocated parent requirement, or a source from which it was derived.  Alternately, use the project toolset to generate a report that lists all requirements that do not trace to a need, parent, or source.  In a document-based approach, these matrices are often managed manually requiring a lot of time and effort and are often defective.  This activity helps establish each requirement has the characteristic defined in the GtWR C1 – Necessary.
7. Use the project toolset to generate trace matrices to confirm each system need, parent requirement, or source allocated to the SOI has at least one derived requirement that addresses that need, parent, or source.  Alternately, use the project toolset to generate a report that lists all needs, parent requirements, and sources that do not trace to an implementing requirement.  In a document-based approach, these matrices are often managed manually requiring a lot of time and effort and are often defective.  (Bidirectional traceability – if the tool allows a trace from a requirement to a need, parent, or source, it should also include the capability to trace each need.  parent or source to its implementing requirement.)  This activity helps establish each requirement has the characteristic C1 - Necessary and the set of requirements has the characteristic defined in the GtWR C10 – Complete.
8. Perform the interface audit discussed in NRM Section 6.2.3.6.
    - Using the interface diagrams and architectural diagrams developed during concept analysis and maturation activities and refined as part of the transformation of the needs into the requirements as a guide, verify that all interfaces have been addressed and the associated interface requirements are included in the requirement set.  This activity helps establish the set of requirements has the characteristics defined in the GtWR C10 – Complete and C11 - Consistent.
    - For each interface requirement, verify it is clear what the specific interaction is between the SOI and the external system and that the requirement includes a pointer to where that interaction is defined, recorded, and agreed to.  This activity helps establish each interface requirement has the characteristic defined in the GtWR C4 – Complete and C7 – Verifiable.

- For each interface requirement, verify the external system referred to has a corresponding interface requirement or has included the interaction with the SOI being developed in its interface control documentation. This activity helps establish the set of requirements has the characteristics defined in the GtWR C10 – Complete and C11 - Consistent.

9. For the set of requirements, verify there are requirements included that address form, fit, function, quality, and compliance. This activity helps establish the set of requirements has the characteristics defined in the GtWR C10 – Complete and C14 - Able to be Validated.
10. Assess allocation to the next level of architecture. A system will have a hierarchy of requirements, based on the physical architecture. Unless at the system element level of the architecture, the SOI requirements will be allocated to the next level of subsystems or system elements within the SOI architecture. Note: in this context "system element level of requirements" is the point where the project has made a buy, build, code, or reuse determination and no further decomposition of the set of requirements for the system element is needed. Using the project toolset to generate allocation and trace matrices, perform an analysis that verifies:
    - Each of the SOI requirements have been allocated to the next level of the architecture
    - Each allocation is correct and complete, i.e., the requirements were allocated to all applicable subsystems and system elements at the next level of the architecture and each of the allocations were to the correct subsystems and system elements.
    - The resulting dependent child requirements in response to allocations of performance, quality, or resources are linked in order to manage changes to the allocated/budgeted values.

    This activity helps establish the set of requirements has the characteristics defined in the GtWR C10 – Complete and C11 - Consistent.

    Note: Allocation assessments will be easier if the project has developed architectural, analytical and behavioral models during the lifecycle concept analysis and maturation activities discussed in Section 2 and refined those models during the transformation activities from needs to requirements. The models will be refined and elaborated for each subsystem and system element within the SOI architecture until all the requirements for each subsystem and system element has been defined.

***Example***

Below is an example of a coffee maker requirement that is assessed for requirement verification:

| |
|---|
| When the Coffee Temperature is set to Extra Hot, the Coffee Maker shall brew 8.0 oz (+0/-0.5 oz) of coffee in less than or equal to 1 minute. |

Assessment for Requirement Verification:

- This requirement contains what (brew 7.5 to 8 oz of coffee), how well (less than or equal to 1 minute) and under what conditions (coffee temperature set to *Extra Hot*).
- The names "brew", "coffee temperature", "Extra Hot", "oz", "minute" are consistent with the architectural model and data dictionary. Ranges are stated for the volume and time.
- The requirement is singular in a sense that it expresses an expectation for the time to brew a given volume of coffee to a specific temperature (performance attributes of the "brew" function) and desired temperature setting.
- Checking traceability, the requirement has a link to the need(s) from which it was transformed.
- All attributes have been defined.

Based on these verification activities, this requirement has been verified to have the characteristics and conforms to the rules for well-formed requirement statements defined in the GtWR. Based on this assessment, the "requirement verification" attribute can be set to "verified".

### 3.2.4 Step 3: Manage Requirements Verification Results

During this effort, a record is created recording the results and outputs of the requirements verification activities. The outputs include the updated input artifacts and work products shown in Figure 20 (Outputs). Defects found in the set of requirements as well as any defects found in any of the individual requirements should be addressed and corrected before they are baselined.

Applications within the project toolset should allow tracking the status of the requirement verification activities using a dashboard that communicates key metrics related to the requirement verification results. These metrics can be obtained using the requirement attributes within the project's toolset. Making this information accessible will enable all project team members and key stakeholders to have the same view of the maturity and status of the requirement verification activities.

## 3.3 Requirement Validation Activities

The GtWR includes the characteristic C14 - "*Able to be Validated*" for sets of requirements. In the context of requirement validation "*Able to be Validated*" means that the project team or customer will be able to:

- Validate that the resulting SOI that was designed and built/coded per the set of design input requirements.
- Obtain evidence that the SOI meets the set of needs and higher-level allocated requirements from which they were transformed within the constraints (such as cost, schedule, technical, and regulatory compliance) with acceptable risk.

Figure 21 provides the inputs, outputs, activities and enablers for the requirements validation activities from the NRM. The basic steps are:

- Plan for Requirements Validation.
- Perform Requirements Validation.
- Manage Requirements Validation Results.

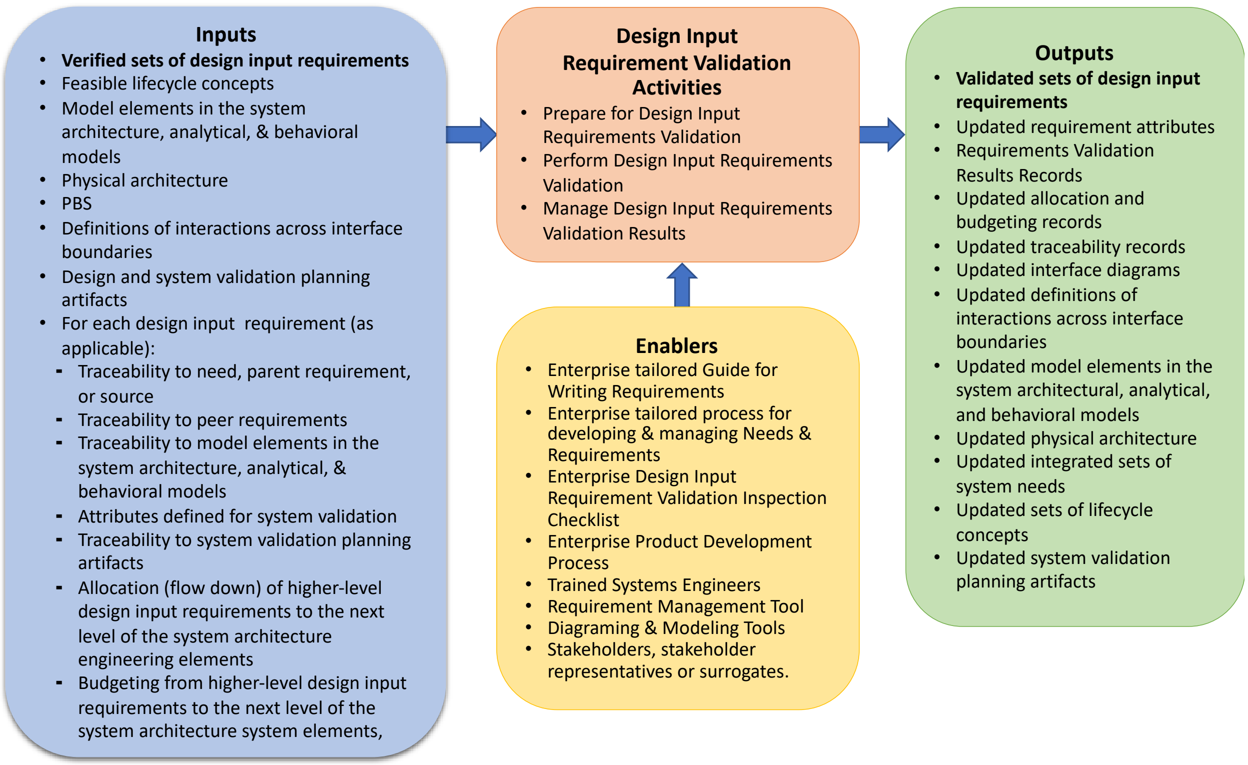



**Figure 21. Requirements Validation IPO Diagram (from NRM).**

### 3.3.1 Enabling Processes and Tools

Organizations vary in their infrastructure to support the lifecycle concepts, needs, and requirements definition processes. However, the list below can be used for an organization to assess their maturity level in supporting this effort and options for growth in their supporting infrastructure. The enabling processes and tools for requirements definition, requirements verification, requirements validation, and management consist of the following (depending on the complexity needed):

- Enterprise tailored Guide to Writing Requirements (based on the INCOSE GtWR).
- Enterprise tailored process for developing and managing needs and requirements.
- Tailored checklist for requirement validation criteria and tasks (see examples in Appendix D).

- Enterprise process for product development.
- Trained systems engineers.
- Requirements Management Tool (RMT).
- Diagramming and Modeling tool(s).
- Stakeholder representation.

### 3.3.2 Step 1: Prepare for Requirements Validation

Why perform requirements validation? Even though individual and sets of requirements may be well-formed, and traceability has been verified, the message they are communicating may not be as intended. It is important that organizations do both requirement verification and requirement validation, failing to do so increases risk that the SOI being developed will fail system validation.

Requirements validation answers the questions: Do we have the right requirements? Are the capabilities communicated by the requirement set sufficient to satisfy the integrated set of needs? Validation also addresses the technical feasibility aspect; it is not the right requirement unless a technical solution exists within the scope of the project with acceptable risk.

Requirements definition activities result in artifacts such as analysis records, diagrams, and models which provide rationale for the transformation that resulted in each of the requirements. Each requirement expression should include rationale to help those validating and implementing the requirement understand the source and intent of what the requirement statement is communicating.

A key preparation activity for requirement validation is the creation of a Requirement Validation Inspection Checklist if one does not already exist. If the organization has a generic checklist, then tailor the checklist to the system being developed and the project's processes. This checklist serves as a standard to measure the requirements against and will help guide the requirements definition and validation activities. Having addressed each of the areas and questions within the checklist will help lead to successful completion of the requirement validation activities. (An example Requirement Validation Inspection Checklist is contained in Appendix D).

### 3.3.3 Step 2: Perform Requirements Validation

The activities associated with requirement validation include the use of inspection and analysis. Validated needs, usage models, business and system analysis records are key inputs to the requirement validation activities. Because the focus of requirement validation is on content and intent concerning what is being communicated, these activities are difficult to automate using automated tools and algorithms.

There are several activities that should be performed to provide objective evidence that the individual and sets of requirements are validated to accurately communicate the intent of the needs, parent requirements, and sources from which they were transformed.

Note: Unlike requirement verification, requirement validation cannot be done without the project team doing the analysis manually – currently none of the NLP applications in a project toolset have the capability to do this type of analysis.

Using the Requirements Validation Inspection Checklist as a guide, the project team should:

1. Use the trace matrices and the rationale attribute to perform an analysis to validate that each requirement statement clearly communicates the intent of system need, allocated parent requirements, or source from which it was derived. This activity helps establish individual requirements have the characteristic defined in the GtWR C8 – Correct.
2. Assess allocation, budgeting, and traceability within the requirements set. A system will have a hierarchy of sets of requirements for each subsystem or system element within the system architecture. Assuming the allocation and traces exist as previously assessed during requirement verification, use the trace matrices to perform an engineering analysis to validate that:
   - The allocations from one level to the next was completed for each requirement and the allocations are correct and complete. This activity helps establish individual requirements have the characteristic defined in the GtWR C8 – Correct.
   - Confirm budgeted values (physical characteristics, performance, quality) are managed such that changes can be made as the design matures, changes are within the budget allocation, and dependent requirements are identified and changed appropriately. This activity helps establish individual requirements have the characteristic defined in the GtWR C8 – Correct and the set of requirements has the characteristics and C11 - Consistent.
   - For each system need, allocated parent requirement, and source, assess whether or not the derived requirement(s) are necessary and sufficient to meet the intent of the system need(s), allocated parent requirement(s), or source they are traced to and from which they were derived. Given that an allocated parent requirement is often allocated to more than one subsystem or system element at the next lower level of the architecture, this means evaluating all child requirements that trace back to the allocated parent requirements. This activity helps establish the set of requirements has the characteristics defined in the GtWR C10 - Complete and C14 – Able to be Validated.
3. For each requirement that references a capability or performance value dependent on a critical technology, confirm the risk attribute has been defined indicating this dependency. Also confirm a Technology Readiness Level (TRL) has been assigned to the technology and there is a plan to mature this technology This analysis helps establish each requirement has the characteristic C6 - Feasible.
4. Confirm the project documented their assessment that the set of requirements is feasible in terms of cost, schedule and technology maturation and has included key product development activities in their cost and schedule estimates, including use of enabling systems, lifecycle concept analysis and maturation, flow down of requirements, design verification, design validation, system integration, system

verification, system validation, and procurement. This analysis helps establish the set of requirements has the characteristic defined in the GtWR C12 - Feasible.

5. In order to reduce the risk of unknowns, using the priority, critically, and Key Driving Requirement (KDR) attributes (See the NRM Section 15), confirm the project has included adequate budget and schedule margins and reserves as well as defined a descope plan. This analysis helps establish the set of requirements has the characteristic defined in the GtWR C12 - Feasible.
6. Confirm with the stakeholders associated with each system need, allocated parent requirement, or source that the message being communicated by the requirement expression is correct and acceptable to ensure the set of requirements are the right requirements, i.e., do they accurately represent the intent of the agreed-to needs, parent requirements, or other sources from which they were transformed? Do the requirements correctly and completely capture what the stakeholders need the system to do in the context of its intended use in its operational environment it terms of form, fit, function, compliance, and quality? This activity helps establish each requirement has the characteristics defined in the GtWR C1 - Necessary and C8 – Correct and that the sets of requirements has the characteristics C10 - Complete, C11 – Consistent, C13 – Comprehensible, and C14 – Able to be Validated.
7. Ensure system verification attributes have been included with each requirements expression addressing the verification method, approach, and success criteria for each requirement. This information must be defined and documented before the set of requirements is baselined. This helps ensure each requirement statement is worded such that the design and the system can be verified to meet the requirement. This activity helps establish each requirement has the characteristics defined in the GtWR C4 – Complete and C7 - Verifiable and that the set of requirements has the characteristics C12 – Feasible and C14 – Able to be Validated.
8. Confirm with the Approving Authorities that the defined verification attributes represents what is necessary for acceptance. This activity helps establish each requirement has the characteristic defined in the GtWR C7 - Verifiable and that the set of requirements has the characteristic C14 – Able to be Validated.

Below is an example of a coffee maker requirement that is assessed for requirement validation:

| |
|---|
| When the Coffee Temperature is set to Extra Hot, the Coffee Maker shall brew 8.0 oz (+0/-0.5 oz) of coffee in less than or equal to 1 minute. |

Trace to Needs:

1. **Need**: The stakeholders need the coffee maker to brew an extra hot 8 oz cup of coffee in less than 1 minute.
2. **Need**: The stakeholders and consumers need the coffee maker to have three temperature settings (warm, hot, extra hot) for the water temperature and control the temperature to within 2 C.

Assessment for Requirement Validation:

- This requirement traces to approved needs, the performance value, and coffee temperature are consistent with the needs, and correctly captures what the stakeholder intends the SOI to do.
- Upon investigation of the technology, it is determined that this requirement is feasible with current technology, enabling this feature to be met within the existing project cost and schedule.

Based on this assessment, the “requirement validation” attribute can be set to “validated”.

### 3.3.4 Step 3: Manage Requirements Validation Results

During this effort, a record is created recording the results and outputs of the requirement validation activities. The outputs include the updated input artifacts and work products shown in Figure 21 (Outputs). Defects found in the set of requirements as well as any defects found must be addressed and corrected before they are baselined.

If it is identified that there is insufficient data to complete requirement validation or lack of an set of needs from which the requirements were transformed, the issue should be recognized as a risk that could cause both design and system validation to be deemed unsuccessful.

Applications within the project toolset should allow tracking of the status of the requirement validation activities using a dashboard that communicates key metrics related to the requirement validation results. These metrics can be obtained using the requirement attributes within the project’s toolset. Making this information accessible will enable all project team members and key stakeholders to have a shared view of the maturity and status of the requirement validation activities.

# 4 NEEDS AND REQUIREMENTS MANAGEMENT

## 4.1 Needs and Requirements Management

Needs and Requirements Management (NRM) is a set of cross-cutting activities spanning all system development lifecycle processes including needs definition, requirement definition, architecture, design, production, integration, system verification, system validation, operations, maintenance, and retirement.

The focus of the NRM activities is to manage the integrated sets of needs and requirements of the system and the subsystems and system elements that make up the SOI to ensure alignment and consistency between those needs and requirements and the project plans, work products, and other PM and SE artifacts generated within the levels of the organization and architecture as well as across all system lifecycle phases.

NRM activities apply to the management of all strategic, operational, system, subsystem and system element lifecycle concepts, needs, and requirements along with the associated artifacts generated within the project, as well as those needs and requirements levied externally on the project from outside the organization by customers or regulatory agencies.

The NRM activities involve the following activities:

- Define, baseline, and oversee the organization's lifecycle concepts, needs, and requirements definition policies, requirements, processes, and work instructions.
- Provide the enablers needed to implement these policies, requirements, processes, and work instructions. Enablers include the IT infrastructure and applications within the project's toolset needed to practice PM and SE from a data-centric perspective.
- Ensure all PM and SE team members are trained in the use of these applications and the concepts associated with practicing PM and SE from a data-centric perspective.
- Provide oversight and insight into the organization's needs and requirements development and management activities.
- Receive, review, and baseline sets of needs and requirements.
- Manage changes to the baselined sets of needs and requirements over the system lifecycle.
- Manage and control the flow down, allocation, and budgeting of requirements between all levels of the of the system architecture.
- Manage traceability: between parent/source and child requirements as well as traceability between dependent peer requirements with links that can be traversed from either direction.

- Ensure vertical alignment and consistency between lifecycle concepts, needs, and requirements at all levels of the SOI architecture.
- Ensure horizontal alignment and consistency between lifecycle concepts, needs, and requirements and the project's plans, work products, and other SE and PM artifacts generated across all system lifecycle phases.
- Monitor, track, and report the development and configuration status of needs and requirements across all levels of the SOI architecture and across the lifecycle.
- Manage and configuration control the work products and artifacts associated with needs and requirements, design, and system integration, system verification, and system validation activities.
- Monitor, track, and report the status of needs, requirements, design, and system integration, system verification, and system validation activities.

Figure 22 provides an overview of the requirements management activities for a SOI. These activities repeat over the various levels of the architecture as needs are transformed to requirements at the system level, and then requirements for the lower-level subsystems and system elements.

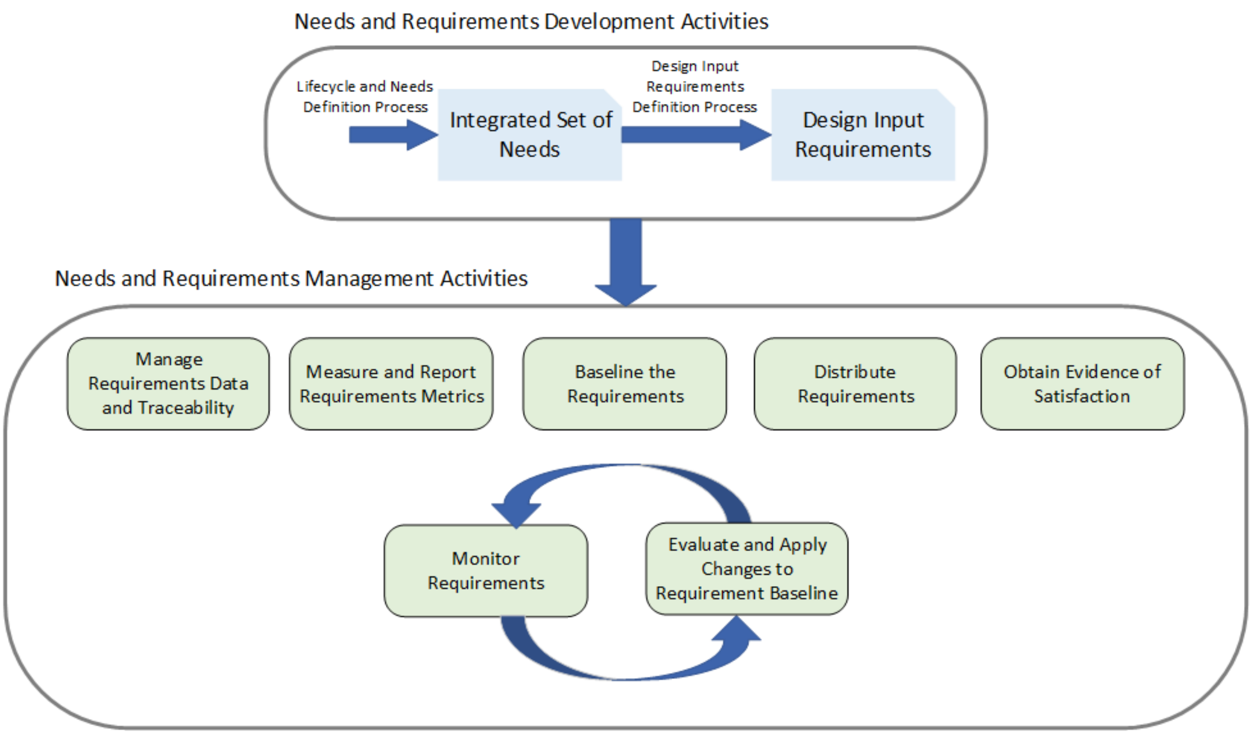



**Figure 22. The Needs and Requirements Management Activities.**

The overall processes, procedures, and work instructions for needs and requirements management will be defined at the operational level of the organization. Key plans include the Project Management Plan (PMP) and a Systems Engineering Management Plan (SEMP). Each project will be expected to develop a PMP and SEMP as part of their project planning activities that is compliant with the business operational level.

The *Project Planning Process* (see the INCOSE SE HB *Project Management Process*) includes the development of a PMP and SEMP that establishes the direction and infrastructure necessary to enable the assessment and control of the project progress and identifies the details of the work and the needed set of personnel, skills, and facilities with a schedule and budget for resources from within and outside the organization needed to produce the SOI.

### 4.1.1 Manage Traceability

As each need is defined, traceability between the need and its sources shown in Figure 7 is established as well as traceability to supportive information such as operational scenarios, validation artifacts, and other project data.

As each requirement is defined, including traceability between the requirement and a need, parent requirement, or source requirement from which it was transformed is established as well as traceability to supportive information such as dependent peer requirements, interfaces, verification artifacts, and other project data. Along with traceability, additional data related to allocation and budgeting of requirements provides information related to the quality of the decomposition at the various levels of the SOI physical architecture. This traceability can be shown in different ways: by reference, by attribute, by toolset linking.

Managing traceability also includes managing the traces between sets of needs and sets requirements for each subsystem and system element that is part of the SOI architecture. System level requirements are allocated to lower-level subsystems and system elements, each having their own sets of lifecycle concepts, needs, and requirements. This results in having parent/child traces as well as peer-to-peer traces for requirements existing in different sets that have some sort of dependency. Ensuring the correctness and completeness of these traces is critical in support of change management discussed later.

To achieve effective management of traceability it is recommended that the principles of integrated data and information requirements management from the NRM be followed to have a data-centric approach to manage the various data elements of the SOI. Attempting to use a purely document-centric approach may work on a simple SOI but will tend to break down as the system becomes more complex; such as increased number of sets of needs and sets of requirements, increased number of interactions between the subsystems and systems elements within the system architecture and the integrated system and its operating environment, and with multiple teams and organizations involved in the development effort.

### 4.1.2 Managing the Unknowns and Unresolved

When initially generating an set of needs and set of requirements, there is often a set of parameters which may require future analysis to determine their actual value. The initial draft set can still be reflected with a designation (of "To Be...(TBX)) that notes values will be populated, or confirmed, at a later time, this allows the end user to work to initial information concurrently with the analyses or investigations required to resolve those values.

One common method of approach is to use the "To Be Determined" (TBD) indication within the need or requirement statement, in place of an actual value. See the following for an example:

**Original Stated Need:** The users need to get brewed coffee from the coffee machine quickly.

In assessing the need it is determined that the word “quickly” is ambiguous, and there is no content for conditions and volume of brewed coffee to be delivered. The project team does analysis to determine under what conditions the coffee will be produced, and the amount for a feasible the volume. Further work is needed to assess technologies before they will be able to define a feasible value for the time it will take to heat and dispense that volume.

With the updated information, the Need statement is updated and an initial requirement statement for the coffee machine is generated:

**Updated Need:** The user needs to get a 8 oz cup of brewed, Extra Hot coffee from the coffee machine in less than [TBD] minutes.

**Initial Requirement:** When the Coffee Temperature is set to Extra Hot, the Coffee Maker shall brew 8.0 oz (+0/-0.5 oz) of coffee in less than or equal to [TBD] minutes.

In this example, the original need was expressed in a way that is not able to be validated because of the ambiguity and questionable feasibility. Working with the stakeholders a more concise need statement was defined addressing the volume to be delivered (8 oz) however the time (quickly), was not able to be agreed to when the need was baselined. The resultant initial requirement was therefore captured with the agreed to volume value with a TBD time value as a placeholder.

After further analysis it was determined the user need based on stakeholder discussions for the time is “within 1 minute.” However, it was not actually confirmed that for this volume, that time is technically feasible. Because of this, it is noted that further analysis is needed concerning the volume of heated coffee to be delivered within the time and align it with what is technically feasible.

A method to reflect this is to use the "To Be Resolved" (TBR) indication. The initial requirement could be updated as follows:

**Updated Requirement**: When the Coffee Temperature is set to Extra Hot, the Coffee Maker shall brew 8.0 oz (+0/-0.5 oz) of coffee in less than or equal to 1 minute [TBR].

*The [TBR] indicates that further research is required to ensure this time is technically feasible.*

The work to resolve the TBR would then involve either prototype effort to ensure this value is achievable, research into the field to understand if this is a nominal value based on benchmarking, performing a model of the activity with simulation, or other type of effort.

Once this issue has been resolved, the TBD in the need and TBR in the requirement can be removed once the actual agreed to and feasible time value has been defined. If the need had been baselined with the TBD, then a change request would be submitted for the change and agreed to by the stakeholders.

The above example reflected a simple use of TBD and TBR to show that the confidence in the value is low, and that the teams responding to the needs and requirements obtain awareness that further work is required to establish the feasible values. This work is often referred to as "TBX Management", as it provides indication of the effort required to resolve the various TBDs and TBRs in the need or requirement (or even throughout the entire sets of needs and requirements).

Note: The usage of "To Be Specified" (TBS) can be used when an entire set of needs or set of requirements is still to be provided. Some organizations use “To be Computed” (TBC) for cases where the identification of a value is dependent on an analysis or computation that has yet to be completed.

The tracking and resolution effort for TBXs can be managed using common project management techniques. One method to control these within the sets of requirements, showing a value for each TBX (e.g., TBD1, TBR3, etc.). Using a document-centric approach, the project can include a table in the needs or requirement document showing all TBXs and associated unknowns, with some notion of a closure date, person or organization responsible for addressing the TBX, along with rationale as to why the value has yet to be defined and baselined. At a minimum, the following notation could be considered “TBX_<by Whom>_<When>”, for example “TBD_Supplier_PDR” or “TBD_Customer_CDR+3 months”.

If multiple sets of needs and requirements exist, then the work to resolve the TBXs may become complex and involve several resources and actions. In this case, a comprehensive TBX tracking mechanism would be helpful, such as using the RMT to connect all TBXs to a common database, where management plans such as forward work, assignee, and closure plan is managed (using a data-centric approach of capturing all TBXs within a requirement set).

TBXs are able to be connected directly to the needs or requirements they impact, aligning the resolution to the impacted need/requirement statement. Some management tools also have the ability to print out a report with the exported specification to provide the summary table for those TBXs captured within the requirements of that specification from the master TBX module (this approach is demonstrated in Figure 23 and Table 9). Keeping track of the TBXs allows for awareness of the maturity of the needs and requirements throughout the development lifecycle and is a valuable metric to evaluate completion of the requirement set.

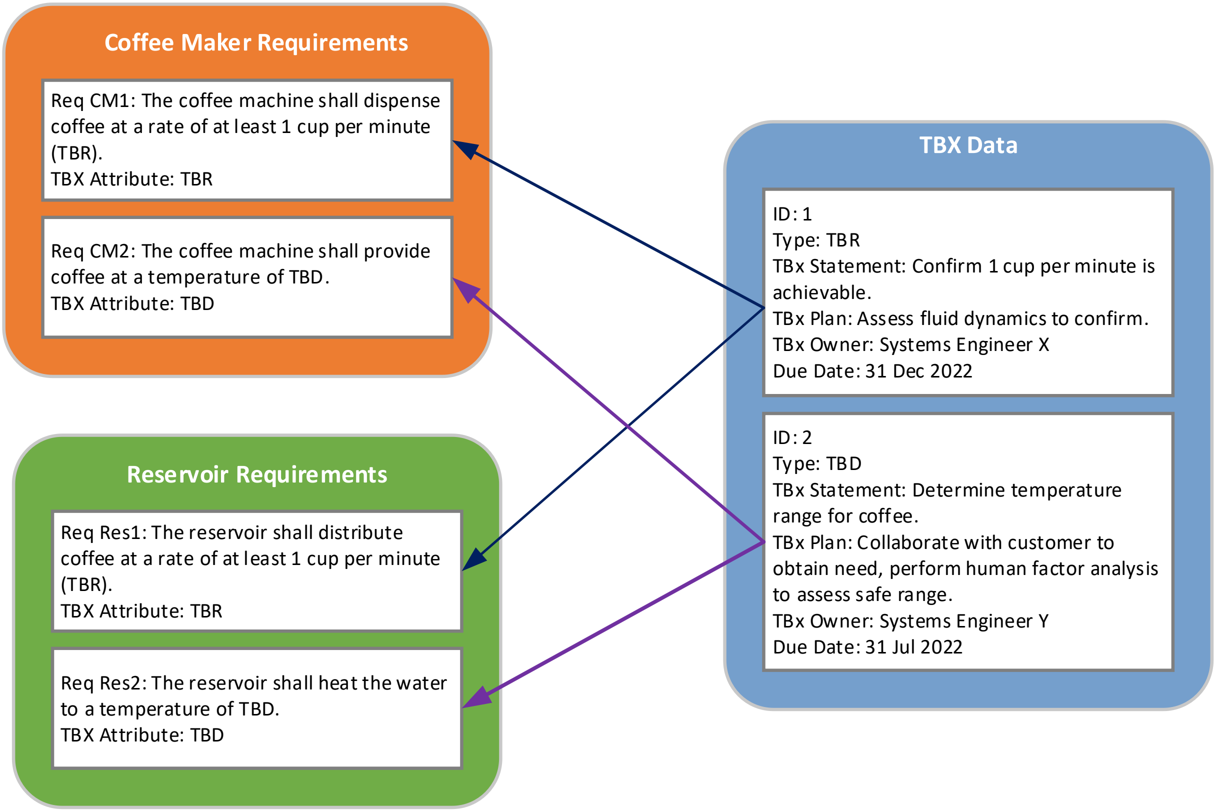



**Figure 23. Example of a Comprehensive TBX Management Approach to Address Entire Set of Needs and Requirements.**

**Table 9. Example TBX Tracking Matrix.**

| TBX ID | Type | Statement | Impacted Reqt(s) | Closure Plan | Owner | Due Date | Status |
|---|---|---|---|---|---|---|---|
| **1** | TBR | Confirm 1 cup per minute is achievable. | CM1, Res1 | Assess fluid dynamics to confirm. | Systems Engineer X | 31 Dec 2022 | Open |
| **2** | TBD | Determine temperature range for coffee. | CM2, Res2 | Collaborate with customer to obtain need, perform human factor analysis to assess safe range. | Systems Engineer Y | 31 Jul 2022 | Open |

When a set of requirements is outsource to a supplier and there are TBXs in the set of requirements, then supplier's role in the resolution of the TBXs must be clearly addressed in the contract as the resolution of the TBXs represents additional time, cost, and resources.

### 4.1.3 Generating a Needs and Requirements Definition and Management Plan

On many projects a SEMP contains structured information describing how the systems engineering effort, in the form of tailored processes and activities, for one or more lifecycle stages, will be managed and conducted in the organization for the SOI to be produced.

From a needs and requirements definition and management perspective, the SEMP:

- Provides a high-level description of the needs and requirements definition and management activities and how they fit into the overall system development lifecycle processes.
- Provides description of organizational roles and their association to the needs and requirements definition and management activities.
- Establishes how the project will develop, maintain, and manage the set of needs and requirements, and the relationship of that information to the other SE artifacts and PM work products generated across all lifecycle stages.
- Includes key definitions of the needs and requirements definition activities and identification of all artifacts and work products generated as part of these activities as well as the major deliverables of the project at each lifecycle stage.
- Addresses the form of the work products (paper vs. electronic), the applications to be included in the project SE toolset to be used to generate and maintain and configuration manage the artifacts and work products and underlying data and information, and the IT infrastructure needed.
- Defines measures that will be used to manage definition of the SOI.
- Defines attributes for needs and requirements expressions that will be used to manage the definition and management of the needs and requirements.
- Defines relationships between the needs and requirements definition and management activities defined in the Needs and Requirement Management Plan (NRMP) and the system verification and system validation activities defined in the project's Integration, Verification, and Validation Plan.
- Manages the work products and artifacts associated with needs, requirements, design, and system verification and system validation activities.
- Defines a project ontology.

Both the PMP and SEMP identify the measures and reports that will be used to manage and track progress of the lifecycle activities.  These reports help define the data and information needed to be managed within the project’s integrated sets of data and information.  Knowing which data and information will be included in the reports helps inform the formation of the project master schema to which individual sets of data and databases will conform.

Supplementing the PMP and SEMP is the Needs and Requirements Definition and Management (NRDM) Plan.  The NRDM Plan focuses on the needs and requirements definition and management activities to be conducted, artifacts to be developed, and deliverables to be produced. Key provisions of the NRDM Plan can include, but are not limited to following:

- Information on how needs and requirements are defined and managed for the project.
- Method of TBX tracking and resolution.
- Change control processes.
- Stakeholder engagement and peer review activities.
- Project artifacts produced from the effort.
- Schedule of development, reviews, and artifact generation.
- Tools and resources required for the effort.

Requirements definition and management activities do not exist in isolation from other project management process activities. A successful needs and requirements management requires integrated execution with the other activities defined in the PMP and SEMP. It is critical to get buy-in of all the provisions in the NRDM Plan from all the key stakeholders addressed in the plan to reduce the risk of stakeholders failing to support the work to be performed within budget and schedule constraints. Getting this buy-in will result in an approved NRDM Plan. Appendix B provides a sample template outline showing contents within the NRDM Plan.

### 4.1.4 Needs and Requirements Configuration Management

The concept of change management is addressed in ISO/IEC/IEEE 15288 [5], and with respect to needs and requirements in ISO/IEC/IEEE 29148 [6]. Additionally, the INCOSE SE Handbook provides additional information in the section that describes the CM Process.

With respect to needs and requirements, the sections below address some strategies for ensuring an initial baseline is created, communicated, monitored, assessed for changes and impacts, implementing the change, and communicating any applied changes.

#### 4.1.4.1 Needs and Requirements Baseline Activities

A SOI is developed in progressive waves of maturity, resulting in iterative and recursive needs and requirements definition over a project lifecycle. At an early phase of a product's needs and requirements definition effort, the sets of needs and requirements are controlled in a configuration managed "baseline". A baseline is an agreed-upon description of the attributes of a product at a point in time, which serves as a basis for change. It is these baselines which the design will be based and will be verified and validated against.

When the set of needs and requirements for the system (and its elements within the system architecture) are completed, they will undergo a process to review their quality and content per the needs verification and needs validation activities discussed in

Section 2.2. Likewise, the requirements undergo a similar activity (Section 3.2 and Section 3.3).

The needs and requirements are assessed for any issues concerning their content and quality and clear up any misunderstandings as to what they are communicating. Once all issues and misunderstandings are resolved the sets of needs and requirements are “signed off” by the project stakeholders and are baselined at some type of formal gate review along with other PM and SE artifacts and work products appropriate to the specific lifecycle stage.

The baselining of the set of needs and requirements is normally the responsibility of a change control board (CCB) or configuration control board (CCB). A CCB is a formally chartered group of operational level stakeholders responsible for reviewing, evaluating, approving, delaying, or rejecting changes to the project in addition to recording and communicating such decisions. Not all projects require the use of a CCB. A project in a heavily regulated industry or one with numerous components, interfaces, risks, and stakeholders may require the use of a formal CCB more than a project without those characteristics. Upon approval, the technical baseline documentation is placed under formal configuration control, this often occurs at a designated time or project review early in the project lifecycle.

This initial baseline is not the finished and complete set of product requirements, it is an initial controlled configuration which allows product development to proceed based on a common framework of needs and requirements. As needs and requirements change during the project, the concept of **change management** takes on a high level of importance. Change management is the process by which the proposed need or requirement changes go through a defined impact assessment, review and approval process, using needs and requirements tracing and version management. Upon acceptance of the change then a new baseline is captured and communicated to the stakeholders.

Once approved, the set of needs and requirements are baselined, any changes are proposed through the CCB. The CCB may choose to have change requests under a prescribed dollar amount approved by business operational level stakeholders and/or the sponsor/customer, and change requests over the threshold approved by the CCB. Those thresholds usually are defined by impact on cost, schedule, or deliverables and should be defined in the project’s Configuration Management Plan.

#### 4.1.4.2 Document and Communicate Baselined Needs and Requirements

Communicating the SOI needs and requirements baseline keeps project stakeholders apprised of the current state and maintains a good level of insight and collaboration. This is especially important with the increasing complexity of projects and programs. The overall state of the needs and requirements management activities, key metrics, and overall status of needs and requirements definition activities should be captured and communicated to stakeholders in accordance with the project’s Communications Management Plan. The metrics to be captured and communicated as part of needs and requirements management should be captured and maintained within the project’s integrated data set. The metrics to be monitored will be communicated via reports and dashboards shared with the project's stakeholders.

#### 4.1.4.3 Needs and Requirements Monitoring and Controlling

Needs and requirements monitoring (insight) and controlling (oversight) iteratively monitors the status of needs and requirements definition activities and manages their baseline over the SOI development lifecycle. In the context of the needs and requirements definition, activities include monitoring and controlling so changes are approved and managed.

The needs and requirements attributes and traceability are used as part of needs and requirements monitoring to help control the projects risks, budget, and schedule. A key part of monitoring and controlling the needs and requirements includes linkages between needs and implementing requirements and parent and child requirements as well as traceability between the requirements and design output specifications.

A major part of monitoring and controlling needs and requirements includes managing the traces between sets of needs and requirements for each subsystem and system element that is part of the SOI architecture. Maintaining the correctness and completeness of these traces is critical in support of change management.

Changes to needs and requirements must be managed as they evolve, identifying inconsistencies that may occur among them, as well as with other project plans, work products, and engineering artifacts. Prior to baselining, changes to needs and requirements will occur as they are initially defined and matured. It is imperative that all changes be thoroughly evaluated to determine the impacts on the cost, schedule, architecture, design, design output specifications, system integration, system verification, system validation, interfaces, lifecycle concepts, and higher and lower-level requirements. The technical team should also ensure that the approved changes are communicated in a timely manner to all relevant stakeholders.

Once the needs have been verified, validated, reviewed, and approved in a gate review and requirements have been verified, validated, reviewed, and approved in the System Requirements Review (SRR) or similar gate review, they are baselined and placed under formal configuration control, after which any suggested change initiates the CM process.

Needs and requirements changes in later lifecycle stages are more likely to cause significant impacts to cost and schedule due to possible changes in the design and rework to parts of the system that have already been built or coded because of the change. The possibility of these impacts must be a major consideration when doing change impact assessment.

#### 4.1.4.4 Needs and Requirements Change Management Process

Change management is of critical importance across all lifecycle stages. Using a data-centric approach to SE, traceability results in a 3-dimensional spider web of relationships both vertically (between levels) of needs and requirements, horizontally between dependent requirements at a given level both within the SOI architecture as well as other SOIs at that level, and horizontally between SE lifecycle artifacts. This spider web is represented within the project's integrated data and information dataset.

Because all the data and information developed across all lifecycle stages are linked together and included in this dataset, it is much easier for the project team to assess a change in any data item within the dataset to determine whether the change could have impacts to other data items within the dataset. A change in any of the information represented by a system need, could result in a change in one or more other needs. Each system need is represented by one or more requirements that were transformed from those needs. Therefore, a change in any of the needs could have a domino effect through the requirements representing those needs. Likewise, a change in a requirement will impact the design and resulting design output specifications.

A process must be defined to address all changes to the needs and requirements at the beginning of the project within the NRDM Plan. Usually, changes are managed within an organization's existing CM process. Managing change involves following the official CM process to ensure that changes are submitted, assessed, and dispositioned per the process.

The CM process also ensures that any needs, requirements, design, design output specifications and other project artifacts that are affected by the change are updated when a change is approved.

Change requests, impact analysis, other artifacts created during the CM process, as well as updates to traceability should be captured and the configuration managed in accordance with the CM process defined in the NRDM Plan. Capturing this information electronically within the project's toolset results in status and disposition of change requests being readily available allowing project participants/stakeholders to evaluate all change requests to the baseline over the life of the project, be informed on the status of changes, and implement changes when approved.

#### 4.1.4.5 Change Impact Analysis

Changes need to be analyzed, which typically involve performing an impact analysis to evaluate the proposed change, if approved, in relation to how it will affect other needs, requirements, design, risks, the system, the project, and the program as well as impacts if not approved. The systems engineer, project manager, and other key engineers and subject matter experts participate in the CM process to assess the impact of the change including cost, performance, programmatic, security, and safety.

Performing functional and sensitivity analyses will ensure that the needs and requirements are feasible and correctly allocated. Rigorous needs and requirement verification and validation will ensure that the requirements can be satisfied and conform to mission objectives.

Effective change management requires a process that assesses the impact of the proposed changes prior to approval and implementation of the change. For this effort, the baseline configuration is used for comparison of the changed configuration, and resources and tools are used to assess impacts to the baseline.

When performing change impact analysis, fundamental considerations are:

- Timing of the proposed change and impacts to work in progress.
- Scope of change and impacts to SOI in definition and associated interfaces.
- Requirement allocations and impact of change to development teams and suppliers of lower-level subsystems and system elements.
- Impact of the change to other needs and requirements and other artifacts linked to the need or requirement that a change is proposed.

If the change is not understood, if impacts to all areas that respond to the changing needs or requirements are not captured, or if the change is not communicated, then the needs and requirements can become misaligned at multiple portions of the project. When a need or requirement change occurs at the system level, it permeates down to lower-level subsystems and system elements through change orders.

Based on allocation and impact of the changed need or requirement it may go all the way to component level, as shown in Figure 18. This set of "change orders" may occur with every change, and at different levels based on where a change can occur. This is one of the impacts that is assessed for acceptance of a change, as the costs associated with these change orders can be a significant cost to the overall project.

Figure 24 highlights areas of hidden costs based on how needs and requirements are developed and implemented on a project. When providing needs and requirements to the teams on a project, the trend is to ensure the developing organizations have the needs and requirements as soon as possible so that they can start their efforts and uncover design solutions sooner, allowing this design information to be used for further refinement of the lifecycle concepts, needs and requirements.

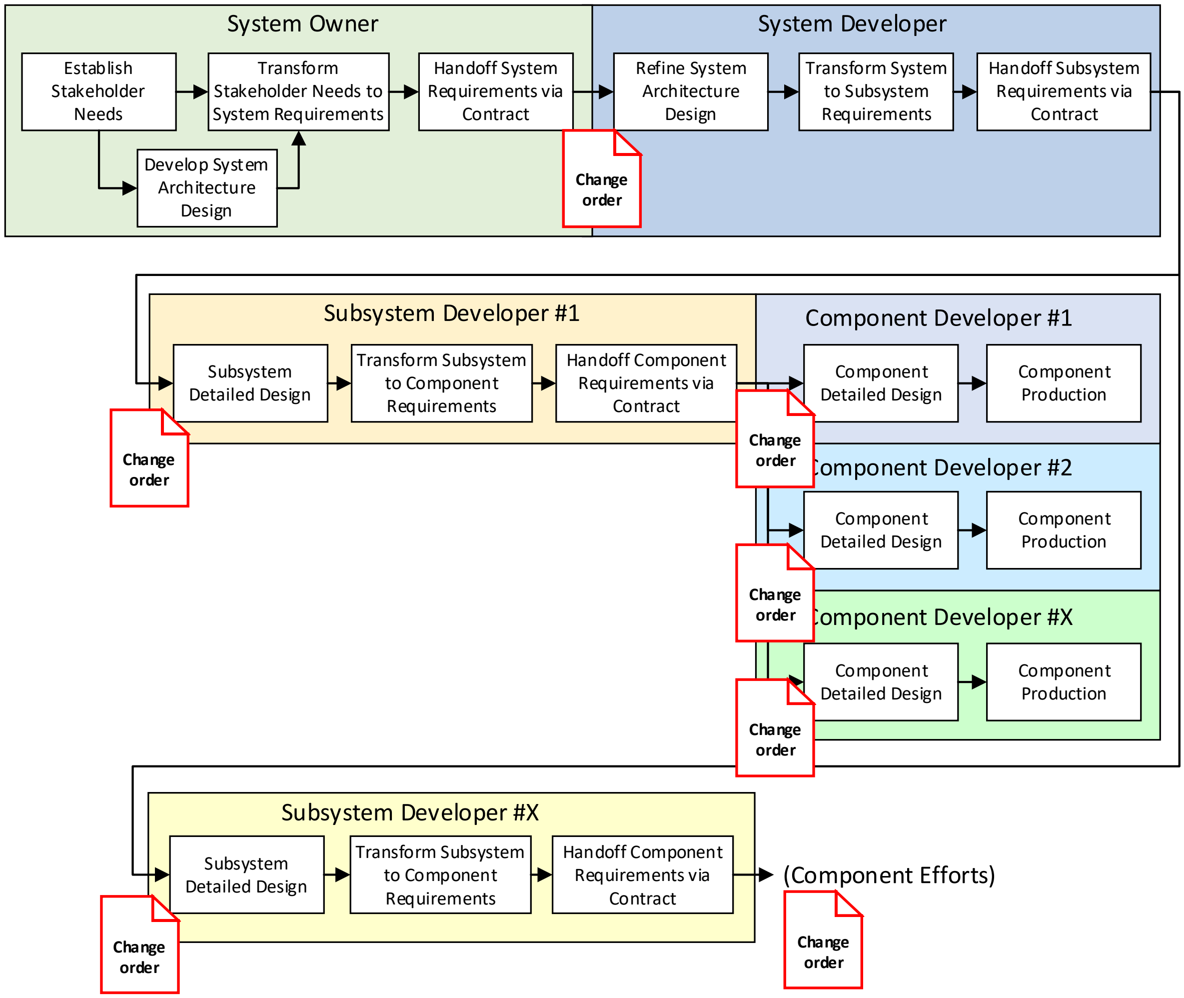



**Figure 24. Impact of Requirement Changes with Multiple Development Organizations.**

However, consideration should be made to address the stability of the needs and requirements prior to levying them on the development organizations, particularly if these are contracted suppliers, as unstable needs and requirements will drive changes that could impact the project significantly in rework time and associated cost.

When a change to a system level need or requirement is proposed, it is necessary to complete an impact analysis to evaluate the proposed change in relation to how it will affect other needs, requirements, design, design output speculations, system, the project, and the program. The impact analysis assesses a proposed change that includes the identification of the risks associated with the change, the work required to incorporate the change, and the schedule and cost implications.

The impacts of changes in traceability are assessed and updated as changes are approved. One should consider how a proposed change may impact the value of the solution to be delivered and whether the change continues to address the business need, mission, goals, objectives, and measures that were approved by the stakeholders.

A key benefit of completing an impact analysis is that it allows for changes within the project to be considered in an integrated fashion, thereby reducing project and product risk, which often arises from changes being made without consideration to the effect on the program, project, and the end product.

Typical tools used to analyze the change impacts are as follows:

- **Needs and Requirements Management Tools:** The tools used to manage the needs and requirements provide the attributes and tracing ability to understand areas of impact for proposed changes. The misalignment of requirement changes in one area can be uncovered by assessing the change across the parent/child requirements and trace to the integrated set of needs, ensuring the change enables the system to meet its objectives, or introduce additional changes that may need to be evaluated. The traceability record within the tools is key to accessing change impacts both vertically between levels as well as horizontally across the lifecycle.
- **Diagramming and Modeling Tools**: Diagrams and models that form the underlying analysis from which the needs and requirements were derived provide the context of the needs and requirements and their relationships to other entities within the model. Changes in one part of the model often have an impact on other parts of the model and their needs and requirements. For complex systems, changes concerning these relationships are difficult to define, understand, and manage.
- **Performance Tools:** This tool provides the ability to assess key performance margins for the system and the current status of the margin. For example, the propellant performance margin will provide the necessary propellant available versus the propellant necessary to complete the mission. Changes should be assessed for their impact on performance margins.
- **Evaluation Subject Matter Expert and Stakeholder List:** This list is developed by the project office to ensure that the appropriate persons are evaluating the changes and providing impacts that could result from implementing or not implementing the change. All changes need to be routed to the appropriate individuals to ensure that the change has had all impacts identified. This list will need to be updated periodically.
- **Cost and Schedule Prediction Tools:** To ensure that the changes align with the scope of the project, tools providing project resources are used to assess whether the change requires additional resources or schedule.
- **Risk Tracking System:** The risk tracking system can be used to identify risks to the project and the cost, schedule, and technical aspects of the risk. Changes to the baseline can affect the consequences and likelihood of identified risk or can introduce new risk to the project. A threats list is normally used to identify the costs associated with all the risks for the project. Project reserves are used to mitigate the appropriate risk. Analyses of the reserves available versus the needs identified by the threats list assist in the prioritization for reserve use.

The process for managing needs and requirements changes should consider the distribution of information related to the decisions made during the change management process.

The CM Process needs to communicate the requirements change decisions to the affected organizations. During a CM board meeting the participants will assess and approve a change, and take actions to update documentation and information within the databases that are to be included as part of the change package. These actions should be tracked to ensure that affected documentation and information is updated in a timely manner. A major challenge is establishing and maintaining a single source of truth.

Note: Beware of changes that are fast tracked through the CM processes. Often fast-tracked changes do not involve the proper degree of change impact analysis that is used to identify possible impacts due to the change. As a result, the fast-tracked change could be implemented with unintended negative consequences. Another type of change to avoid are deferred changes. These are changes that are approved, but their implementation is deferred to a later time, e.g., after a key milestone or moved into a different budget cycle. When the deferred change was approved it was approved for the system configuration at that time. The danger is that, depending on the deferred time period, other changes could have been approved and implemented resulting in a different system configuration. When the deferred change is implemented, it again could have unintended negative consequences because of the changed configuration.

#### 4.1.4.6 Assessing Impacts of Change Across Lifecycle Stages

A quote from Gentry Lee, Jet Propulsion Laboratory, is: "*A good systems engineer knows the partial derivative of everything in respect to everything*." This means that a change that occurs anywhere in a system's integrated/federated dataset could have an impact on other artifacts represented in that dataset. A change to a stakeholders' needs and requirements could impact a lifecycle concept, which could impact multiple needs, which could impact multiple requirements, which could impact design and implementing design output specifications, which would impact the system. Any of these changes could impact production as well as system integration and system verification and validation planning, especially cost and schedule. When assessing changes, the project must assess the impacts to the system verification and system validation programs and associated artifacts.

What are the impacts to system verification and system validation planning when a level 3 requirement changes? Does that change impact its parent requirement(s) at level 2 and associated verification artifacts? Related/dependent peer requirements and associated verification artifacts at the same level? Level 4 requirements and associated verification artifacts? How will that requirement change impact the architecture and design? What is the ripple effect of that one change? How will that change impact the design output specifications? Production? Will software need to be recoded? Hardware re-built to the changed design output specifications. What will be the total cost of that change? How will that change impact the schedule?

For example, several of the authors were part of a large government program where requirements existed to six to seven levels across multiple projects and architectures within each project that made up the program. Due to changes in mission and funding (primarily due to politics), several key changes were mandated at the top levels of management. At the program level, it seems that only four or five requirements for the overall system architecture would be changed. What is the big deal? Because of the ripple effect, thousands of requirements at the lower levels were impacted along with design implementations of those requirements. All the associated system verification and system validation artifacts for those requirements were impacted as well. Sadly, the costs associated with the top-level program requirements were too great, and the program was canceled.

For today's increasingly complex, software-centric systems, it is very difficult, if not impossible, for a human brain to comprehend all the relationships and impacts due to change. Because of this, it is critical that projects have the tools necessary to develop the integrated/federated dataset, enabling them to manage these relationships across all lifecycle stages and PM and SE artifacts generated at each stage. For highly regulated systems, e.g., medical devices or nuclear based systems, being able to manage the relationships and assess the impacts of changes, no matter which level, and documenting the impacts are mandatory for certification or qualification for use.

Because of these possible effects of change, it is critical that the developing organization's CM office consider the impacts of changes across the entire web of data and associated artifacts, especially all the artifacts that are part of the project's system verification and system validation programs and associated artifacts.

#### 4.1.4.7 Assessing Needs and Requirements Scope Creep

Needs and requirements instability is a leading cause of scope creep, and the needs and requirements monitoring and controlling activities ensure that requested changes are processed through the project's change control process.

"Scope creep" and "requirements creep" are the terms used to describe the subtle way that needs and requirements grow incrementally during the course of a project. The tendency for the set of requirements is to increase in size during the course of development, resulting in a SOI that is more expensive and complex than originally intended.

However, some of the needs and requirements creep involves new needs and requirements that did not exist when the sets were baselined, and were not anticipated, during Needs and Requirements Definition activities. These new needs and requirements are the result of evolution, and if a relevant SOI is to be developed, they cannot be ignored.

There are several techniques for avoiding or at least minimizing scope and requirements creep:

- In the needs elicitation phase, work with the stakeholders to bring out both implicit as well as explicit needs and requirements that might otherwise might not be stated.

- Define a set of lifecycle concepts that has been through the lifecycle concepts analysis and maturation activities and agreed-to by the customer and relevant stakeholders. In addition to operations, address all lifecycle stages of the SOI as well as alternate nominal and off-nominal cases.
- Establish a process for configuration control and change management as part of the CM Process. This will determine which stakeholders have the authority to submit changes formally to the CCB and the type of changes that will be allowed. Maintaining configuration control is essential to having a control over the project budget and schedule.
- Establish a process for assessing changes and their impacts on the rest of the system and other artifacts within the project's data and information model. Compare this impact with the consequences of not approving the change.
- Determine the feasibility of implementing a change in terms of budget, schedule, technology, and risk. If it cannot be accommodated within the established resource margins, then the change most likely should be disapproved.
- Determine criticality – if the change does not involve the ability to address a critical need, then the change most likely should be disapproved.

One of the authors attended a briefing of a project that was getting an award for their success in developing a system that met the stakeholder needs within cost and schedule constraints. When the project manager was asked what was key to the project success, the project manager said that he had extremely strict rules for change management. His philosophy was that if it is not broke, is safe, or is note breaking a law – then leave it alone! It did not take long for members of the project team to limit any proposed changes to those that met these criteria. While this seems drastic, it worked.

## 4.2 Needs and Requirements Management Methodology and Examples

Management of needs can take similar form to the management of requirements; the mechanism to manage can take the form of document or data-centric using different tools. For a simple SOI, the document-centric approach may be sufficient, using a spreadsheet to capture the needs, their source, stakeholder, and other attributes. There is a point where the complexity of the SOI and number of needs and requirements as well as organizations and teams will prove too great to manage this data - at that point it is recommended that a tool be used to manage the data and connect to other sources of SOI data and artifacts across the development lifecycles; two example tool methods are provided in this section.

As the integrated set of needs can address different aspects of the SOI, it is advised that they be organized into function, fit, form, quality, and compliance groupings (described in Section 2.1.7). The use of attributes is also advised to help ensure the needs are viewed within the right context. As an example, the priority of the need should be understood from the stakeholder perspective, recorded as an attribute, and managed for changes in priority or for assurance in meeting the need.

**Why is it worth managing the integrated set of needs compared to only managing the requirements?** Primarily the needs reflect the reason for and intent of the requirement(s) transformed from the need and having it defined enables the resultant system to be aligned with the integrated set of needs for the SOI. Management of the set of needs ultimately supports system validation activities and it is the set of needs that represent what is necessary for acceptance. In addition, the quality of the sets of requirements is directly related to the quality of the needs from which they were transformed.

This is shown in Figure 9, and further expanded upon in the example shown in Figure 25. The key to effective management of the set of needs is to use tools to help ensure alignment with the overall set of SOI data and information from which they were derived (see Figure 7), to help manage the set of needs and system validation against those needs, and to establish and maintain traceability between the needs and resulting requirements.

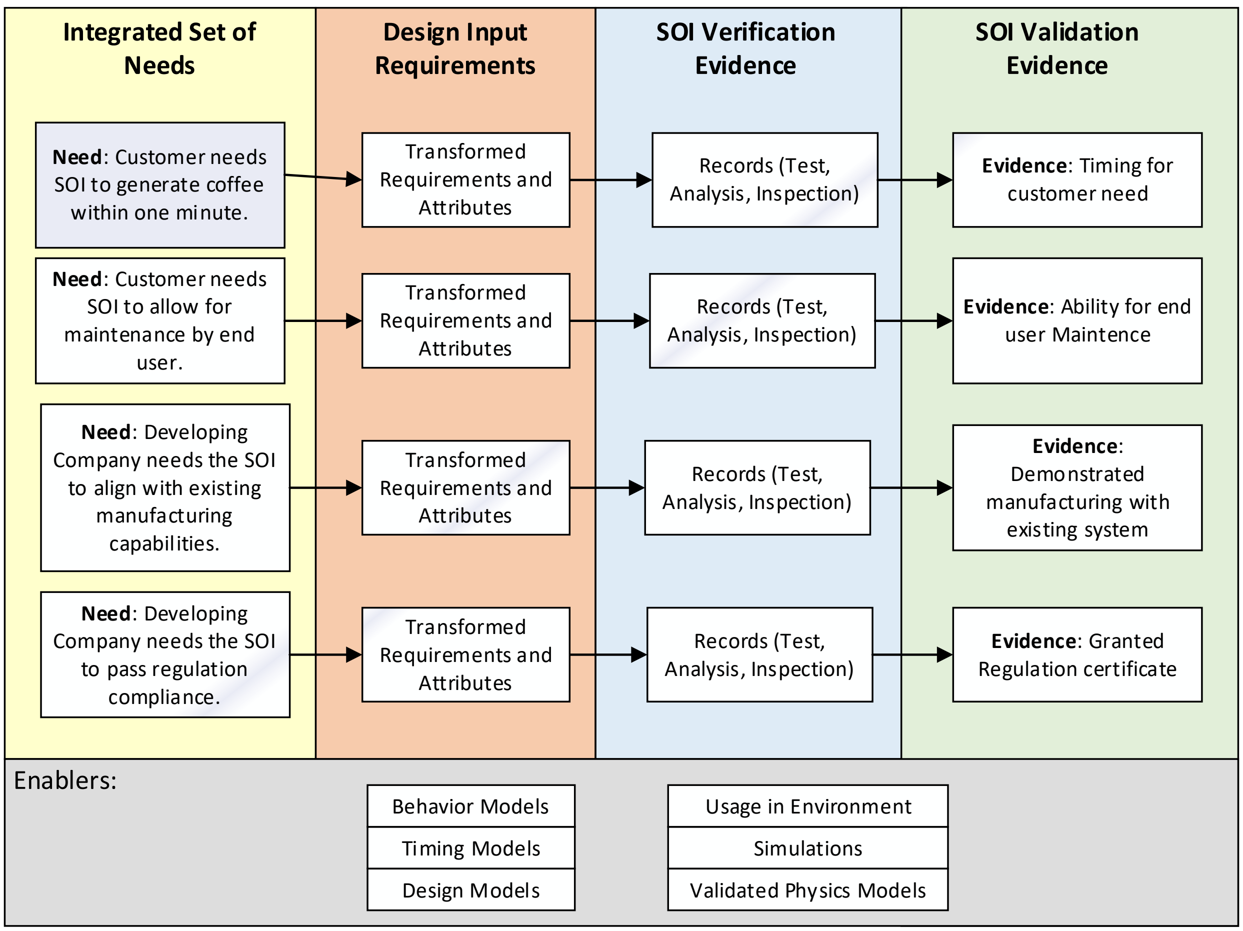



**Figure 25. Example of Management of Needs with Associated Data in Support of System Validation.**

### 4.2.1 Management and Trace Using a System Model

The definition of needs is enabled by using a system model, this was described earlier in Section 2. One approach to management of the set of needs is to use the same toolset which defines the system model. This enables the needs to be related directly to the users, stakeholders, use cases, MGOs, measures, drivers and constraint, risks, and lifecycle concepts from which they were derived.

Figure 26 provides how this organization could look in a system model, where the set of needs are managed with the problem domain (logical model). These are located at the highest abstraction of the model, separate from the SOI and lower-level requirements, however the traceability of these needs can be captured with the requirements and SOI elements using the model linking capability (Figure 22, Figure 27, and Figure 24).

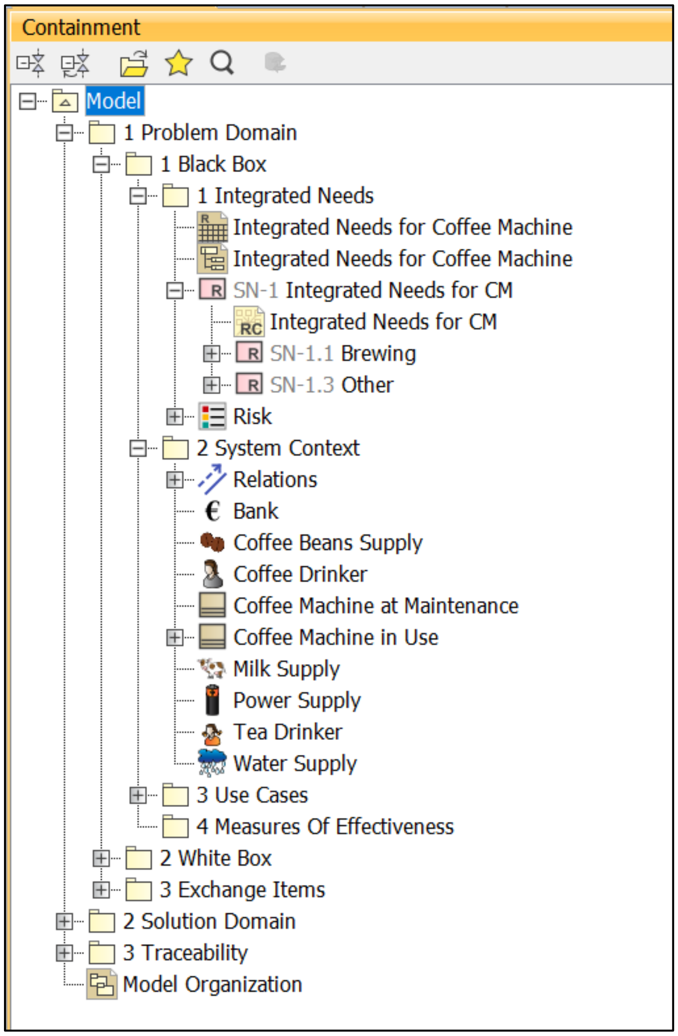
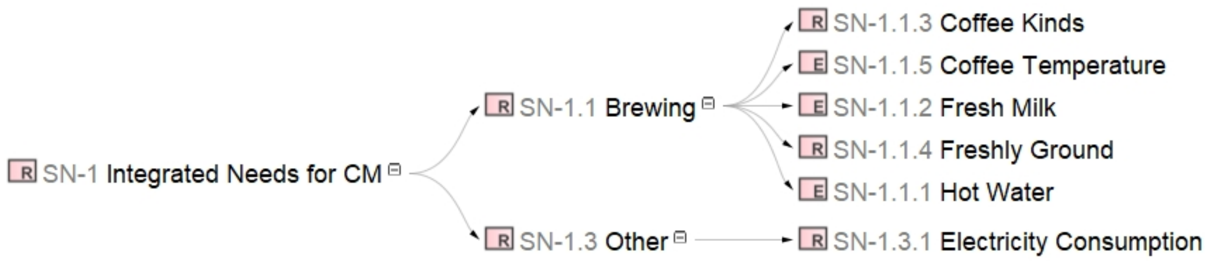
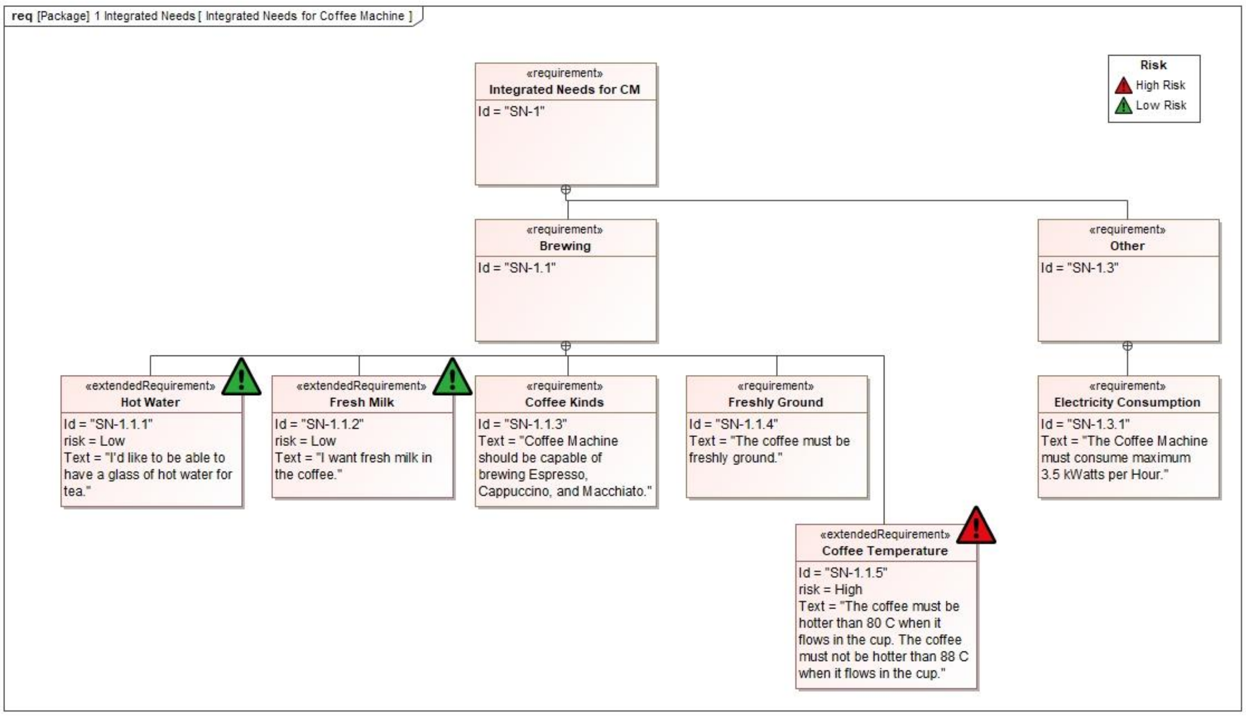



**Figure 26. Example System Model Organization of the Integrated Set of Needs.**




| # | Name | Text |
|---|---|---|
| 1 | SN-1 Integrated Needs for CM | |
| 2 | SN-1.1 Brewing | |
| 3 | SN-1.1.3 Coffee Kinds | Coffee Machine should be capable of brewing Espresso, Cappuccino, and Macchiato. |
| 4 | SN-1.1.5 Coffee Tempera | The coffee must be hotter than 80 C when it flows in the cup. The coffee must not be hotter than 88 C when it flows in the cup. |
| 5 | SN-1.1.2 Fresh Milk | I want fresh milk in the coffee. |
| 6 | SN-1.1.4 Freshly Ground | The coffee must be freshly ground. |
| 7 | SN-1.1.1 Hot Water | I'd like to be able to have a glass of hot water for tea. |
| 8 | SN-1.3 Other | |
| 9 | SN-1.3.1 Electricity Consu | The Coffee Machine must consume maximum 3.5 kWatts per Hour. |

**Figure 27. Example System Model Capture of the Set of Needs.**

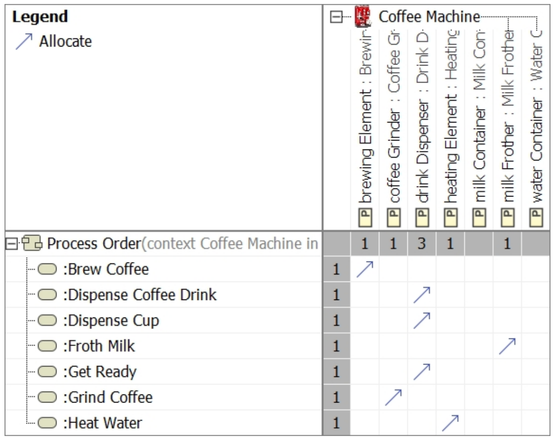



**Figure 28. Example System Model Trace of Needs to System Elements.**

Modern system modeling tools are able to show attributes, enabling the set of needs to be managed with associated data, such as external source (such as reference to standards), priority, etc.

### 4.2.2 Management and Trace in a Requirements Management Tool

Many modern tools developed for management of requirements are actually capable of inclusion of other sources of project data, such as the example shown in Figure 29 and Figure 30. In this approach, the set of needs is captured in the set of data and traced to the associated requirements, records, etc. These tools are also capable of connecting data to other tools, such as language-based system models, allowing the needs to be traced to their use cases and activities.

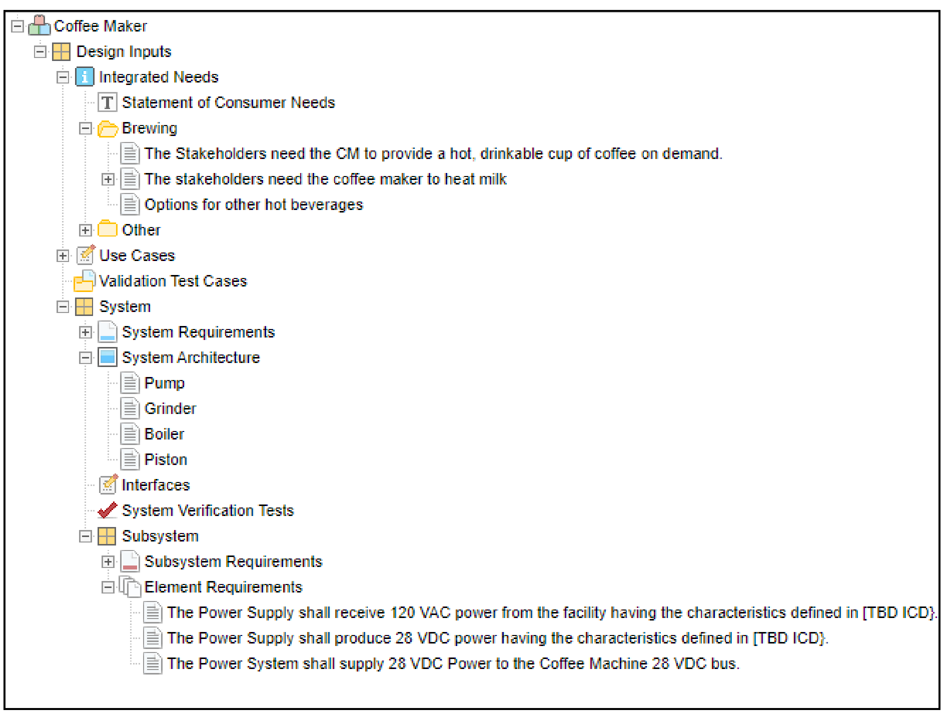



**Figure 29. Example Requirement Tool Organization of Needs and Requirements.**

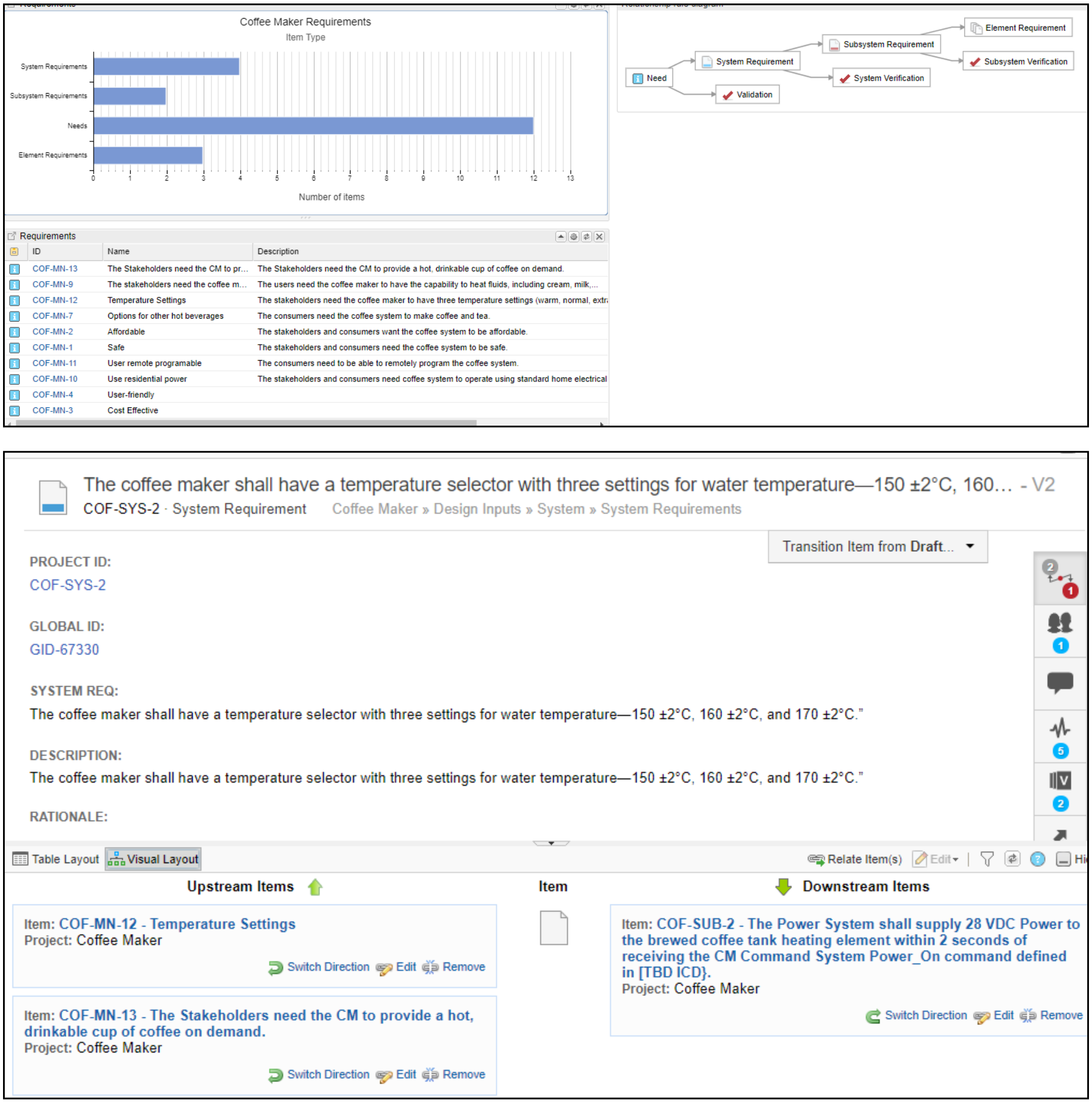



**Figure 30. Example Requirement Tool Capture and Trace of Needs and Requirements.**

## 4.3 Needs and Requirements Management Best Practices

Requirements are often associated with a larger quantity of data that needs to be managed, and therefore this section focus is on best practices associated with management and associated ability to capture, baseline and configuration manage that data. Some fundamental approaches are highlighted in Table 10, helping a project team obtain an understanding of some optimal approaches in determining methods and tools for management of their needs and requirements.

### Table 10. Recommended Needs/Requirements Management Considerations[7].

| Process Step | Considerations and Recommendations |
|---|---|
| 1. Select whether a "document-centric" or "data-centric" management approach will be applied. | Project complexity (technical performance and product structure), document quantity, existing processes at the organization, required effort to develop updated processes.<br><br>For complex projects with significant amount of needs, requirements and standards, the selection of a data-centric approach is advised to realize cost and schedule savings compared to the investment of establishing this process for a project. |
| 2. Select a Needs and Requirements Management Tool and a modeling tool for the project. | Quantify of needs and requirements, expectation on maturity and change evolution based on the product being developed, existing tools, costs to purchase any new tools, associated training costs and learning curve schedule impacts.<br><br>For new projects developing complex systems with a significant number of needs and requirements, the selection of a user-friendly and collaborative requirements management tool and modeling tool that can share data and information is advised to realize cost savings compared to the cost of a new tool and associated training required. |
| 3. Ensure needs and requirement quantity and quality is addressed during the needs requirements definition effort. | Quantity of requirements, project complexity in product structure, number of outside organizations receiving the requirements; usage of standards such as INCOSE Guide to Writing Requirements provides the methods to achieve requirement quality.<br><br>The amount of time spent ensuring the needs and requirements sets are minimized and consolidated for the project may vary, but for complex projects with a significant number of needs and requirements it is advised that the time invested early to improve the quality of the needs and requirements sets (reduce overlaps, minimize the number of needs and requirements to sets of needs and requirements that are necessary, complete, consistent, correct, and feasible) will yield benefits in labor costs and schedule later in the project lifecycle. |
| 4. Select the timing to levy needs and requirements on any suppliers that will be developing systems and system elements. | Stability of the needs and requirements, heritage of the supplier, anticipated cost of future change cycles, anticipated costs associated with any schedule delays of starting project efforts.<br><br>Based on needs and requirements maturity, evaluate whether to 1) wait on establishing the contracts, 2) bring on the suppliers with a contract to help advance the preliminary lifecycle concepts and resulting needs and requirements while working early lifecycle concepts and needs definition activities, or 3) establish the contract with an official set of requirements based on the customer's baselined set of needs.<br><br>For highly unstable sets of needs and requirements, it is advised to either wait to establish the contract or to use an approach to have the supplier support the needs and requirement definition activities; this approach could realize cost savings associated with future change cycles and rework by the supplier.<br><br>For cost optimization, the Acquirer should invest more in the lifecycle concepts and needs definition activities before developing a set of requirements that will be given to a Supplier. |

### 4.3.1 Attributes for Requirements Management

Requirements management uses data as a way to assess requirements progress, stability, quality, and source, this data is referred to as **requirement attributes**.

Per the NRM, an attribute is additional information included with a need or requirement statement which is used to aid in the definition and management of that need or requirement. There are several requirement attributes recommended in the NRM Section 15 that are key to help maintain high-quality sets of needs and requirements, a sample from NRM is shown in Table 11.

**Table 11. Sample Attributes of Requirements Statements (from NRM).**

| | |
|---|---|
| **Attributes to Help Define the Requirement and its Intent** | A19 - Owner |
| A1 - Rationale | A24 - Approval Date |
| A2 - Trace to Parent* | A25 - Date of Last Change |
| A3 - Trace to Source* | A26 - Stability |
| A4 - States and Modes | A27 - Responsible Person |
| A5 - Allocation/Budgeting | A28 - Need or Requirement Verification Status |
| A6 – System Verification or Validation Success Criteria | A29 - Need or Requirement Validation Status |
| A7 - System Verification or Validation Strategy | A30 - Status (of the Need or Requirement) |
| A8 - System Verification or Validation Method | A31 - Status (of Implementation) |
| A9 - System Verification or Validation Responsible Organization | A32 - Trace to Interface Definition* |
| A12 - Condition of Use | A33 - Trace to Peer Requirements* |
| A17 - Originator/Author | A34 – Priority |
| | A35 – Criticality or Essentiality |

* Note: the above attributes dealing with traceability are often established within the tool

While not all of the attributes are needed for every project, the NRM provides recommendations on a minimum set, as well as how to apply them and considerations of use.

Ultimately, these attributes provide enabling capabilities for managing requirements on a project and allow for situational awareness of the requirements completeness and stability.

### 4.3.2 Managing Needs and Requirements for a Product Line

When developing a product line, or an element within a product line, needs and requirements management will need to introduce an ability to manage product variant needs and requirements and consider needs and requirements that are common to all variants as well as needs and requirements that are unique to a specific variant in support of the overall product line needs and requirements management activities.

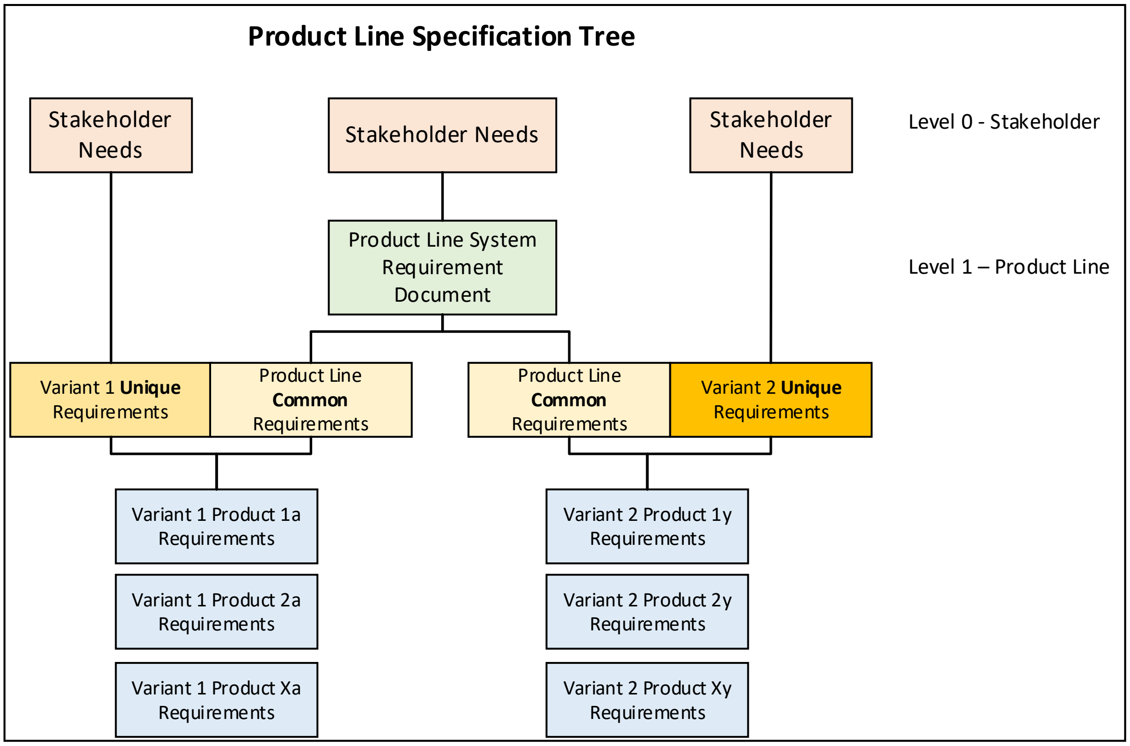



**Figure 31. Example of a Product Line Requirement Tree showing Inter-relationships among the Variants.**

Individual variants of a product line may have common requirements that exist for all products in the product line, as well as unique needs requirements that exist only for that variant (as shown in Figure 31). For example, needs and requirements dealing with quality and compliance may be common to all variants. Some functions may be common to all variants but have variations in performance. Other functions may be unique to a specific variant.

For the coffee maker, an example would be for the development of the coffee maker product line where one variant is a smaller model for individual consumption, another variant for larger model designed for a larger household, and an even larger variant to be used in an office setting with many employees; the developing organization will ensure the common product line features are included in each variant as well as address the unique aspects reflected in the individual variants. For the definition of needs and requirements within a product line, the needs include both the overall product line needs as well as needs associated with each variant.

Needs and requirements management in this type of effort will need to address traceability of each variant need and requirement and ensure that changes are assessed against impact to the variant as well as the overall product line. Additionally, any changes of the product line common needs and requirements will need to consider impacts to all of the variants or allow for tailored needs and requirements to be created for the variant products that will keep the original needs and requirements. Some organizations will use attributes to manage requirement variants. Some examples of these attributes from the NRM are shown in Table 12.

Table 12. Sample Attributes of Needs and Requirements Statements (from NRM).

| | |
|---|---|
| **Attributes to Help Define the Requirement and its Intent**<br>A41 - Applicability<br>A42 - Region<br>A43 - Country<br>A44 – State/Province | A45 - Market Segment<br>A46 - Business Unit<br>A47 - Product Line<br>A48 - Product Line Common Needs or Requirements<br>A49 - Product Line Variant Needs or Requirements |

Referring to Figure 27, there could be an attribute used for needs and requirements that are common to all variants, and an attribute to indicate needs and requirements that apply to a specific variant. In addition to A47 in Table 12 for product line, there could be attribute A48 to indicate needs or requirements common to all variants of the product line and attribute A49 would include list of each variant used to indicate needs and requirements unique to a specific variant.

Using this approach, the project's toolset could be used to form specific sets of needs and requirements for each variant. Each set would include both the needs or requirements common to the variant as well as the needs or requirements specific to that variant.

### 4.3.3 Managing Needs and Requirements for Internal Teams

When defining needs and requirements for external organizations, the needs and requirements are communicated formally in the contract via the Statement of Work (SOW) and a needs or requirements document. The contract then clearly communicates what is necessary for acceptance, enabling proof of contractual compliance for the established service or product.

However, when establishing needs and requirements for SOI systems and system elements being developed within the same company or organization, some efficiencies may be gained by looking for a less formal, data-driven approach towards the formality of needs and requirement definition and management and associated compliance, verification, and validation activities.

A recommendation is for the organization to establish a method of needs and requirements management, change control, and communications that enables all project members to ensure they are working to the correct needs and requirements, the communication to stakeholders is established, and discussions on any change or compliance impacts are able to be held and assessed against related needs and requirements. For some organizations this could be achieved entirely within a project toolset, as shown in the earlier section, where others may need a more formal method between organizations to establish provider/customer deliverables.

### 4.3.4 Managing Requirements for Heritage or Off-The-Shelf Products

One area to consider is whether a formal set of requirements is required for specific products within the organization that have already been developed using a capabilities-based approach (heritage products), or already exist and are being purchased as an off-the-shelf (OTS) product.

Section 12 of the NRM provides a comprehensive approach to ensure existing products can be reconciled against a set of needs and requirements for an SOI. When determining the needs and requirements for the OTS, the project team uses an external view the OTS with a focus on form, fit, function, quality, and compliance and address the following questions:

- What functionality and level of performance is needed when the SOI is operating within the SOI's operational environment?
- What are the allowable induced environments for the OTS?
- What are the interface boundaries, how are they defined, and what are the characteristics of each interaction across those boundaries?
- What is the allowable envelope, mass, power requirements, and other physical attributes requirements for the OTS?
- What quality (-ilities) requirements need to be met by the OTS?
- What standards and regulatory requirements does the OTS need to comply?

Because the heritage or OTS candidates were not developed to the project's specific needs and requirements, not all needs and requirements defined for the OTS product will be able to be met. A framework for evaluation of COTS hardware assemblies is presented in Figure 32. This framework includes evaluation considerations to address customer and project stakeholder satisfaction as well as addressing any needs for insight / oversight from the customer.

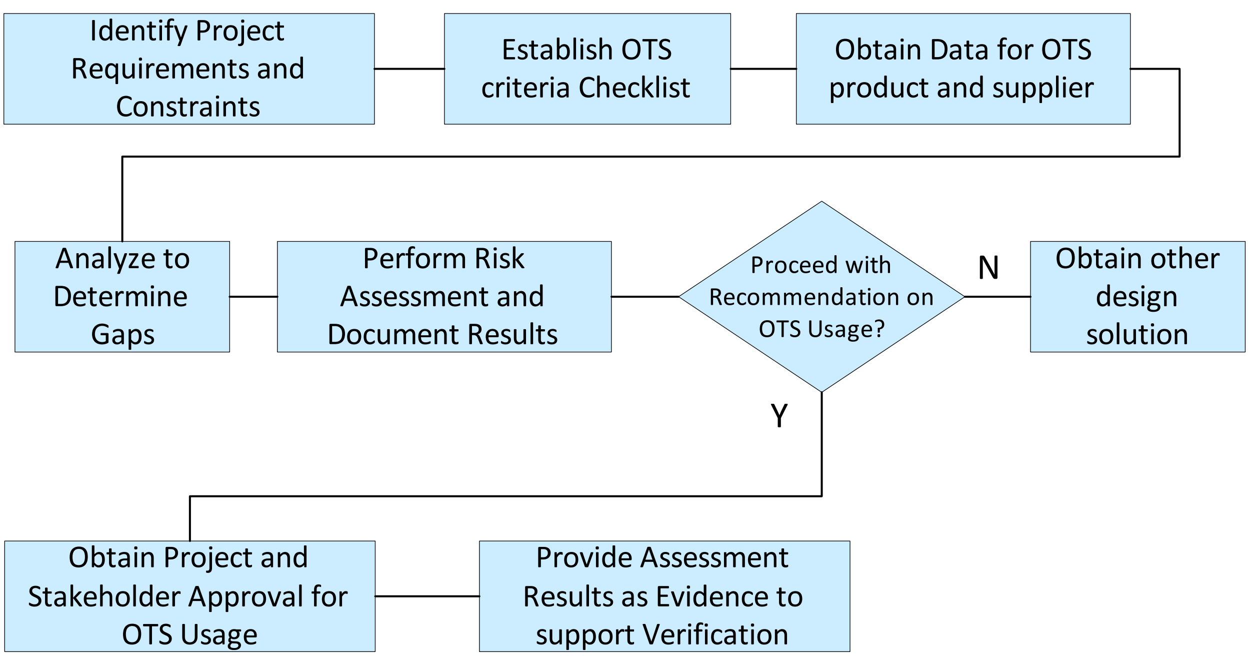



**Figure 32.** **Framework for OTS Evaluation[8].**

Understanding the rationale, priority, and critically for all needs and requirements allows the project team to make tradeoffs during their evaluation and selection process. The capture and management of the requirements for these OTS products is often done at the integrated level, where the interaction of the OTS product can be assessed within the integrated SOI (such as through integrated test or analysis), and the logistics of compliance of the OTS product can be captured against the higher level needs and requirements using existing data from the heritage OTS product and any additional considerations related to unique environments, needs, or capabilities of the SOI.

### 4.3.5 Connection of Needs and Requirements Data to System Verification and System Validation Activities

To ensure the project's system verification and system validation programs are managed effectively to meet milestones, budget, and schedule, a series of metrics should be developed to track and manage the planning and performance of the project's system verification and system validation programs across all lifecycle stages.

Examples of metrics include:

- Number of needs and requirements that have the Success Criteria and Strategy defined and Method identified.
- The number of needs and requirements that have all system verification and system validation attributes defined.
- The number of system verification and system validation Events that have been scheduled.
- The number of system verification and system validation Execution Records that have been completed (one for each system verification and system validation activity).
- Percent of system verification and system validation Instances that have been closed (status and outcomes).
- Burn-down plan for system verification and system validation Instances closures (against needs and requirements).
- Number of variances, concessions, waivers, and deviations.

System verification and system validation metrics can be generated by the project's toolset to produce reports or dashboards that communicate the current status of the project's system verification and system validation programs to the project team and to the customers. Metrics need to be generated as often as necessary to provide the project team and the customers' insight into the status of the project's system verification and system validation programs. The closure of hundreds (or thousands) of system verification and system validation Instances can involve a large percentage of a project's resources, budget, and schedule.

These metrics can be generated from various sources including information from the project's scheduling tools, the attributes defined for the individual needs and requirements expressions, as well as any special metrics contained in the project's integrated/federated datasets developed to help manage the project's system verification and system validation programs.

### 4.3.6 Considerations on Tools and Data Exchange

Over the years multiple software applications have been developed to address management of needs and requirements; an RMT can enable a project's success with its execution, verification and validation of the end product. As the number of software applications has been growing over the past few years, the INCOSE organization has partnered with Project Performance International (PPI) to develop an online, searchable systems engineering tools database, https://www.incose.org/setdbtest/system-engineering-tools-database.

Desired capabilities and features of needs and requirements management tools include definition, collaboration, change control, and trace to other project data. It is recommended that tools are more effective if they are setup with project configuration and templates beforehand, ensuring the project team spends their time on the definition management of the needs and requirements and not infrastructure development of the management tool.

While projects may vary in their needs and ability to afford different solutions, a recommendation is to determine the overall complexity and scale of the needs and requirement data, the number of project team members using the tool, the anticipated number of changes and change reviewers that should be interfacing with a tool, and the need to connect requirement data with other project data, such as system model or test procedures. Considerations to use in tool selection include:

- Requirements specification and prioritization - ability to add, edit, delete and prioritize requirements easily.
- Traceability and dependencies - ability to create relationships between requirements and change the data model to reflect the traceability needed in the organization.
- Stakeholder management, review and collaboration - ability to give feedback on requirements or initiate workflows to approve requirements.
- Change control - ability to baseline requirements, track changes after a baseline, or revert requirements set back to a baseline.
- Visual Modeling - ability to create and edit models in the tool or link requirements to visual models.
- Import/export and reporting - ability to import to / export from MS Word, Excel, Visio or other sources and report the requirements, models or subset of either group.
- Lifecycle concept, needs, and requirements definition support - ability to set up own templates and object types to support a methodology with things like checklists, issues, risks or constraints.
- Task / Iteration management - ability to track definition tasks on requirements, set release or iteration dates, or create burndown charts.
- Licensing, support and tool administration - flexible licensing for the tool, adequate support materials and ease of maintenance.

- Scalability, integrations and ease of use - how intuitive is the tool, ability to scale, ability to integrate with other applications.
- Software application infrastructure - existence of a digital engineering approach towards data-integration, the set of applications and servers within an organization, and the maintenance needs of the applications.

See the NRM Section 16 for a more detailed discussion concerning features the project's toolset should have.

# APPENDIX A: ACRONYMS AND ABBREVIATIONS

| Acronym | Description |
|---|---|
| CCB | Configuration Control Board |
| CDR | Critical Design Review |
| CM | Configuration Management |
| ConOps | Concept of Operations |
| CPSC | USA Consumer Product Safety Commission |
| CR | Concept Review |
| FDA | USA Food and Drug Authority |
| FFBD | Functional Flow Block Diagram |
| GtNR | Guide to Needs and Requirements |
| GtWR | Guide to Writing Requirements |
| GtVV | Guide to Verification and Validation |
| HB | Handbook |
| ICD | Interface Control Document |
| INCOSE | International Council on System Engineering |
| IPO | Input/Output |
| I-RDM | Information based Requirements Development and Management |
| IT | Information Technology |
| KDR | Key Driving Requirement |
| LI | Leading Indicators |
| MBD | Model-Based Design |
| MBSE | Model-Based Systems Engineering |
| MGO | Mission, Goals, Objectives |
| MOE | Measures of Effectiveness |
| MOP | Measures of Performance |
| MOS | Measures of Suitability |
| NASA | National Aeronautics and Space Administration |
| NRM | Needs and Requirements Manual |
| NRMP | Needs and Requirements Management Plan |
| NLP | Natural Language Processing |
| NPR | NASA Procedural Requirements |
| OpsCon | Operational Concept |
| OSLC | Open Services for Lifecycle Collaboration |
| PBS | Product Breakdown Structure |
| PDR | Preliminary Design Review |
| PDD | Process Definition Documents |
| PMI | Project Management Institute |
| PMP | Project Management Plan |

| Acronym | Description |
|---|---|
| RE | Requirement Engineer |
| RM | Requirements Management |
| RMT | Requirements Management Tools |
| RWG | Requirements Working Group |
| SDR | System Design Review |
| SE | System Engineering |
| SEBoK | SE Book of Knowledge |
| SEI | Software Engineering Institute |
| SEMP | Systems Engineering Management Plan |
| SNR | Stakeholder Needs and Requirements |
| SOI | System of Interest |
| SRR | System Requirements Review |
| SRD | System Requirement Document |
| TBC | To Be Computed |
| TBD | To Be Determined |
| TBR | To Be Resolved |
| TBS | To Be Specified |
| TBX | To Be [generic indication] |
| TDP | Technical Data Package |
| TPMs | Technical Performance Measures |
| TRL | Technology Readiness Level |
| USA | United States of America |
| V&V | Verification and Validation |
| VGO | Vision, Goals, and Objectives |
| WI | Work Instruction |

# APPENDIX B: NEEDS AND REQUIREMENTS DEFINITION AND MANAGEMENT PLAN CONTENT

This section supplies recommended content for a NRDM Plan. The template should follow the organization's approach and format for generation of a plan. Key provisions of the NRDM Plan can include, but are not limited to following:

- Provide details needed to implement the needs and requirements definition and management activities consistent with the project requirements defined in the PMP and SEMP concerning how the needs and requirements definition and management activities of the project will be conducted.
- Define the oversight (control) and insight (monitoring) of the project's needs and requirements definition and management activities.
- Identify the relevant stakeholders who will be involved in the needs and requirements definition activities.
- Assign roles and responsibility, authority, and resources needed to define needs and requirements, perform the needs and requirements management activities, and develop the needs and requirements management work products.
- Define the artifacts and work products that need to be developed as part of needs and requirements definition and management.
- Identify and manage the relationships between the set of needs and requirements, and other project and system engineering artifacts and work products throughout the system development lifecycle.
- Provide a workflow for performing the needs and requirements definition and management activities and delivery of needs and requirements definition and management artifacts and work products.
- Define how the set of needs and requirements will be developed, maintained, tracked, managed, verified, validated, and reported.
    - Define how, and the form in which, the needs and requirements will be recorded.
    - Define the attributes that will be defined and maintained as part of each need and requirement expression.
    - Define the measures, metrics, and reports that will be used to monitor and control the definition of the set of needs and requirements.
    - Define the review and approval processes for baselining the integrated sets of needs and sets of requirements.
    - Define the system verification and system validation Success Criteria, Strategy, and Method for each need and requirement against which the SOI will be verified and validated.

  - Describe the process for the flow-down (allocation) and budgeting of requirements from one level of system architecture to lower-level subsystems and system elements and the management of those allocations.
  - Describe the process for establishing traceability between levels of requirements (parent/child, child/parent, source/child child/source) as well as peer-to-peer traceability between dependent requirements at the same level both within a subsystem or system element and with other subsystems and system elements.
- Define the CM process for changes to needs and requirements.
- Define the level of CM/data management control for all needs and requirements definition and management artifacts and work products.
- Identify the training for those who will be performing the needs and requirements definition and management activities.

# APPENDIX C: NEEDS ELICITATION QUESTIONS

| Topic | Stakeholder / Source |
|---|---|
| **Problem/Opportunity Statement** | |
| Do you agree with the problem/opportunity statement? Why/Why not? | |
| Why do you think the new system is needed? | |
| If you know of similar existing systems, what would you like to see different? Output? Function? Performance? | |
| **MGO/Measures** | |
| Are there any other changes they would like to make? | |
| Are other measures that need to be defined? | |
| Do they think the values stated in the MGOs and measures are feasible? If not, why? | |
| **Lifecycle Stages** | |
| What Lifecycle stages are you involved in? | |
| **Use Cases or Scenarios** | |
| How do you see the SOI used? | |
| What is the "day-in-the-life" of the SOI for the Lifecycle you are involved? | |
| Can you think of any alternate nominal scenarios? | |
| Can you think of any off-nominal scenarios? | |
| **Capabilities, Functions, Performance** | |
| How many users/operators? | |
| What capabilities/functions/performance do the users need? What are the priority of each (Musts, nice to have, etc.)? What is the criticality of each? | |
| Do the functions have to occur within a certain amount of time? | |
| What are the inputs and outputs for each function? What is the nominal state for each input/output vs. off-nominal state? | |

| Topic | Stakeholder / Source |
|---|---|
| What operational environment does the function occur? | |
| Does the function have to occur repeatably? For how many times? What triggers each function? | |
| Any constraints for the SOI to consider? | |
| **Interactions/Interfaces with External Systems** | |
| What external systems does the SOI interact with across all Lifecycle stages.? Identify each interaction. | |
| **Quality** | |
| Usability—complexity? with or without training? | |
| Reliability—to what level? Over what length of time? Mean time to failure? | |
| Safety and Security—what are the threats? Hazards? What needs to be protected? Etc. | |
| Serviceability/Sustainability—through user updates, system upgrades, etc. Mean time to repair? | |
| Scalability—how many users? How much data and information? | |
| Testability/inspectability—built-in? With provided tooling and/or equipment? | |
| Availability—to how many? For how long? Over what time periods? | |
| Are there any other quality (-ilities) that need to be addressed? | |
| **Safety/Security** | |
| Are there any safety or security needs that should be considered? | |
| **Compliance** | |
| What regulations and standards that the SOI and project must comply? | |
| **Issues/Risks/Opportunities** | |
| What Issues/Risks/Opportunities do you see with this SOI? | |
| **Assumptions** | |

| Topic | Stakeholder / Source |
|---|---|
| Of the information you provided what assumptions have you made? | |
| **Rationale** | |
| For any need statement you have provided, what is the rationale? | |

# APPENDIX D: CHECKLISTS

## D1. Sample Need Verification Checklist

1. **Establish the need expressions are well-formed:** Verify individual needs expressions and the sets of needs have the characteristics per the rules defined in the INCOSE GtWR or similar guide; this can be done manually, through use of a natural language processor (NLP), or through use of another autonomous tool.
2. **Verify each need expression contains a complete set of attributes: Are** the attributes agreed to by the project team defined? The project toolset should be able to produce a report concerning completeness and data defined for each attribute. For example, does the text in the rationale statement include the information expected to be in the rationale? Does the rationale state why the need statement is necessary? Is the source of any numbers explained?
3. **Establish the needs expressions are well formed such that the system will be able to be validated to meet the needs as written:** Verify individual needs expressions have the set of system validation attributes defined and have values agreed to by the project team. For example, does the text in the validation attribute that defines the validation success criteria include the information expected?
4. **Establish that the integrated set of needs is complete:** Use the project toolset to generate trace matrices to confirm traceability of each need to one or more input artifacts (sources shown in Figure 7) and confirm each need traces to at least one source from which it was derived. Use the project toolset to generate trace matrices to confirm each source has at least one derived system need that addresses that source. (Traceability – if the tool allows a trace from a need to its source, it should also include the capability to trace each source to its implementing lifecycle concept and associated need statement(s)).
   - For the set of MGOs, there should be a trace to a need addressing the MGOs.
   - For each measure there should be a trace to a need that addresses that measure.
   - For each standard and regulation there should be a trace to a need for each standard and regulation.
   - For each SOI level function within the functional architectural and behavioral models there should be a trace to a system need that addresses that function.
   - For each higher-level requirement or sets of requirements allocated or budgeted to the SOI there should be a trace to a need that addresses that requirement or sets of requirements (or a lifecycle concept that address those requirements.)
   - Confirm the project has done risk assessments and for each risk that will be mitigated, the project has established traceability between the risk and the lifecycle concepts that define a concept for mitigation of that risk and traceability to the need that addresses that mitigation concept.
   - Referring to the External Interface Diagrams, Context Diagrams, Boundary Diagrams, or functional models for the SOI verify there are needs that address each of the interfaces and that each need that addresses an interface traces back to the source that identified that interface.

## D2. Sample Need Validation Checklist

1. **Establish that the needs are correct:** Use the trace matrices to perform an analysis to validate that each needs expression clearly communicates the intent of those sources from which it was derived.
2. **Establish the integrated set of needs is complete:** Assess whether the integrated set of needs are necessary and sufficient to meet the intent of the sources they are traced to and derived from. (*Note: there will be cases where a need or group of needs satisfies more than one source.).*
3. **Establish each need is feasible:** For each need that references a capability or performance value dependent on a critical technology, confirm the risk attribute has been defined indicating this dependency. Also confirm a TRL has been assigned to the technology and there is a plan to mature this technology.
4. **Establish the integrated set of needs is feasible:** Confirm the project documented their assessment that the integrated set of needs is feasible in terms of cost, schedule and technology maturation and has included key product development activities in their cost and schedule estimates, including use of enabling systems, lifecycle concept analysis and maturation, requirement definition, design verification, design validation, system integration, system verification, system validation, and manufacturing or coding.
5. **Establish that the integrated set of needs are correct:** Confirm with the stakeholders associated with each source that the message being communicated by each need statement is correct and acceptable to ensure the set of needs are the right needs, i.e., do they accurately represent the agreed-to sources and lifecycle concepts from which they were transformed? Do the needs correctly and completely capture what the stakeholders need the SOI to do in the context of its intended use in its operational environment, when operated by its intended users, in terms of form, fit, function, compliance, and quality? Will a SOI that is validated to meet the needs, result in the stakeholder's real-world expectations to be met? There is no substitute for the stakeholders validating that individual needs and the integrated set of needs represents what is necessary for acceptance.
6. **Establish that what is necessary for acceptance has been defined and agreed to:** For each need expression, ensure system validation attributes have been defined and included in the need expression addressing the validation Success Criteria, Strategy, Method, and organization responsible for system validation. This information must be defined and documented before the integrated set of needs is baselined. This also helps ensure each need is worded such that the requirements, design, and the system can be validated to meet the needs.
7. **Establish that the needs associated with interfaces are well formed such that the SOI will be able to be validated to meet the needs as written:** For each need that addresses an interface with an external system, check to see if the specific interactions have been defined and documented in some type of interface definition type document or record, e.g., Interface Control Document (ICD), Data Dictionary, etc., or there is a plan in place to define these interactions.

## D3. Sample Requirement Verification Checklist

Below is a set of basic checks that can be used as a starting point for creating a tailored requirement verification checklist. It is assumed that requirements are organized in sets that applies to the SOI or any of its subsystems or system elements (within brackets are the characteristics in the GtWR that contributes to compliance):

1. Does the requirement statement follow the organizations agreed-to template for writing requirements? Consider the requirement type if the boilerplate is tailored to different types of requirements (functional/performance, interface, quality, etc.) (C9)
2. Does the statement contain the basic elements of: entity, what, how well and under what conditions? A requirement should have one 'how well' statement to be singular, there are however many ways to express how well, can be a number or a list of things. (C3, C4, C5, C10)
3. Are entity names and function names consistent with the system architecture model? (C3, C8)
4. Is the requirement constructed so that compliance can be determined by observing the behavior at the boundary of the entity it applies to? Essentially the 'how well' part of the requirements should refer to something observable on the boundary of the entity, not on other entities. (C2)
5. Is required traceability in place? Basic traceability to needs, parent requirements, architecture model elements, system analysis records, and design/system verification methods (C1, C4, C6, C8, C7)
6. Have the required agreements been completed, articulated by lifecycle state attributes. Verify to established business rules. (C4, C6)
7. Are a sufficient set of attributes defined for the requirement considering the lifecycle state of the project? Verify to established business rules. (C4)
8. Are entity names, function names, terms and units used consistently throughout the set? (C11)

## D4. Sample Requirement Validation Checklist

Below is a set of basic checks that can be used as a starting point for creating a tailored requirement validation checklist, it is assumed that requirements are organized in Sets that applies to the SOI or any of its subsystems or system elements. This requirement validation checklist enables the project team to confirm the resultant set of design input requirements aligns with the integrated set of needs and are complete.

- Do we have a complete set of system functions? Review traceability between needs and use cases and system functions. Review if use cases are sufficiently addressed by functional/performance requirements.
- Are all requirements traceable to one or more needs, parent requirement, or source?
- Do lifecycle analysis and maturation records exist that justifies the transformation of a need to one or more requirements that will result in a system that will sufficiently satisfy the need?
- Do all functional/performance requirements trace to a function allocated to the entity?
- Do system analysis records exist that confirms technical feasibility of the requirement to an acceptable level of risk while considering project scope and schedule?
- Has the set of requirements been matured through the proper reviews and agreement processes according to business rules considering the lifecycle state of the project?
- Are all interactions across all interface boundaries represented by an interface requirement? Have all interactions been defined and referenced within the interface requirements?
- Do requirements sufficiently express the intent of the needs from which they were transformed considering the foreseeable set of operating scenarios (including rainy day scenarios, what variation will be seen on interfaces)?
- Are the requirements written in a language understood by the developing organization?
- Does the set of requirements contain any conflicting or inconsistent requirements?

# APPENDIX E: REFERENCES

[1] INCOSE Needs and Requirements Manual, INCOSE-TP-2021.002-01, 2022.

[2] INCOSE Systems Engineering Handbook (INCOSE-TP-2003-002-04), 2015.

[3] INCOSE Guide to Writing Requirements, 2022.

[4] INCOSE Guide to Verification and Validation, INCOSE-TP-2021-004-01, 2022.

[5] ISO/IEC/IEEE 15288, 2015.

[6] ISO/IEC/IEEE 29148, 2018.

[7] T. Katz, Cost Optimization in Requirements Management for Space Systems, 2021.

[8] T. Katz, "Evaluation of COTS Hardware Assemblies for use in Risk Averse, Cost Constrained Space-based Systems," in *INCOSE IS2019*, 2019.

# APPENDIX F: COMMENT FORM

| | |
|---|---|
| Reviewed Document: | |
| Name of submitter (first name & last name): | |
| Date Submitted: | |
| Contact information (email address): | |
| Type of submission (individual/group): | |
| Group name and number of contributors (if applicable) | Please read examples carefully before providing your comments (and delete the examples provided.) |

| Comment sequence number | Commenter name | Category (TH, TL, E, G) | Section number (e.g., 2.1.1, no alpha) | Specific reference (e.g., paragraph, line, Figure no., Table no.) | Issue, comment and rationale (rationale must make comment clearly evident and supportable) | Proposed Changed/New Text -- MANDATORY ENTRY (must be substantial to increase the odds of acceptance) | Importance Rating (R = Required, I = Important, T = Think About for future version) |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

Submit comments to Working Group chair. Current WG chair will be listed at:

http://www.incose.org/techcomm.html . If this fails, comments may be sent to info@incose.org (the INCOSE main office), which can relay to the appropriate WG, if so requested in the comment cover page.

(BACK COVER)

INCOSE Publications Office
7670 Opportunity Road, Suite 220
San Diego, CA 92111-2222 USA