System Requirements Definition

System Requirements Definition

> The printable version is no longer supported and may have rendering errors. Please update your browser bookmarks and please use the default browser print function instead.

Lead Author: *Tami Katz* ***Contributing Authors:*** *Lou Wheatcraft, Mike Ryan*

The System Requirements Definition process transforms the stakeholder view of desired capabilities into a technical, developer view of how the system can achieve those capabilities. System requirements describe requirements which the system-of-interest (SoI) must fulfill to satisfy the stakeholder needs and are expressed in an appropriate combination of well-formed textual statements and supporting models or diagrams. Inputs into this process are the life cycle concepts and integrated set of needs generated during the System Concept Definition activities.

System requirements play major roles in systems engineering, as they:

- Form the basis of system architecture and design activities.
- Form the basis of system integration and verification activities.
- Provide a means of communication between the various project team members that interact throughout the project.

Outputs of the System Requirements Definition process serve as inputs to a number of other technical processes, which include System Design Definition, System Architecture Definition, and System Verification.

Contents

Principles and Concepts

System Requirements Definition uses the outcome of the System Concept Definition activities to address what the system must do so that the integrated set of needs will be realized by the SoI. As shown in Figure 1, this information is then provided as input to the System Architecture Definition and System Design Definition processes, as well as the System Verification process.

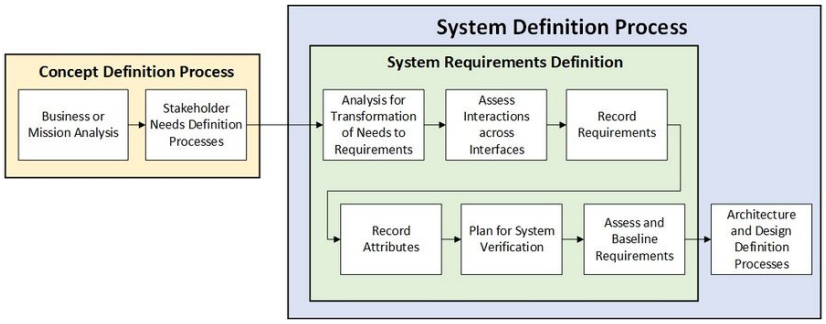

Figure 1. System Requirements Definition is the first activity in the System Definition Process. (SEBoK Original figure derived from ISO 15288-2023)

Defining requirements is not only an exercise in writing it is also an exercise in engineering. Every requirement represents an engineering decision as to what the SoI must do, or a quality the system must have, to meet the needs from which they were transformed. Determination of what the system must do to meet a need is through a process of detailed requirement analysis, which can include the development and use of models, simulations, and prototypes.

A requirement statement is 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." (INCOSE NRM 2022). In this context the use of the term “quality” means an inherent feature or a distinguishing attribute.

There are many types of requirements; this article describes the requirements which are used as inputs into the System Architecture and System Design Definition processes. As such, these requirements can be referred to as design input requirements, which are defined iteratively and recursively as the integrated system is decomposed into subsystems and system elements (the SoI could exist at any level of a system architecture).

Process for Generating System Requirements

Transforming Needs to System Requirements

The System Requirement Definition activities begin with the transformation of the integrated set of needs into a set of requirements for the SoI. These requirements must be appropriate to the level that the SoI exists within the system architecture and communicate “what” the SoI must do to meet the needs, avoiding requirements that state implementation of “how” to achieve the design realization of the physical SoI.

Needs communicate the stakeholder’s perspective concerning their expectations of what they need the system of interest (SoI) to do from an external, black box view.

- Example: The user needs the coffee maker to stop heating water once the user-selected temperature has

been achieved.

The sources of needs are described in Stakeholder Needs Definition. These sources include the stakeholder expectations and needs, drivers and constraints, risk analysis, and life cycle concept analysis and maturation activities. These needs are treated as inputs to the System Requirement Definition process, which results in a set of design input requirements for the SoI.

Requirements communicate the technical, developer perspective concerning what the SoI must do to meet the stakeholder needs from an internal white box view.

- Example: The Coffee Maker shall stop the heating function once the selected Brew Temperature has been achieved.

The process of generating a requirement statement is more than changing the subject of the need statement, it is an analysis of what the system must do to achieve what is needed. There are many types of analyses and tools that are used to make this determination:

- functional analysis,
- interface analysis,
- data flow diagrams,
- performance analysis, and
- needs to transformation matrix.

Requirement analysis for functions and performance can be performed by using system models (supported by Model-Based Systems Engineering (MBSE) applications). Key enabling model views include activity diagrams, sequence diagrams, state machine diagrams, and structure diagrams (for hierarchy and interfaces). An example functional analysis is shown in Figure 2.

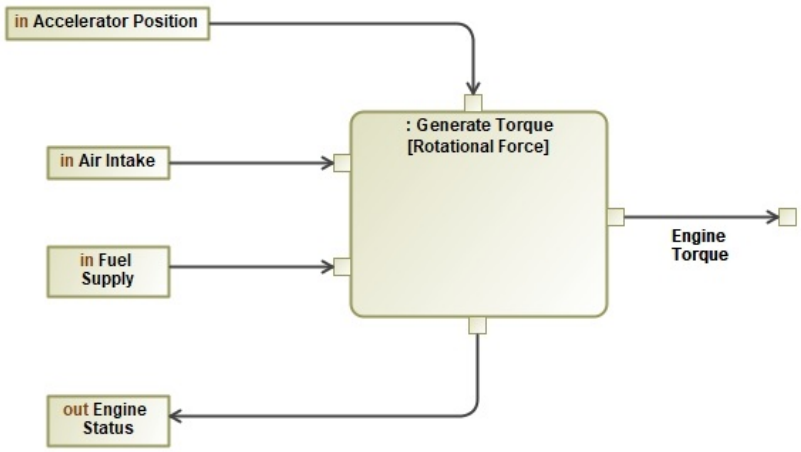

Figure 2. Example Function Analysis using a SysML Activity Diagram. (SEBoK Original)

Close and frequent coordination with the stakeholders is necessary to ensure the transformation is accurate and traceability is maintained. This results in a set of system requirements communicating measurable characteristics which can form the basis for system realization. A detailed analysis of a single need statement may result in multiple requirements expressing what the system must do to meet it, including definition of measurable performance criteria (INCOSE NRM 2022).

Use of Attributes

An attribute is additional information associated with an entity which is used to aid in its definition and management. Well-chosen attributes, when properly defined and tracked, can enable correct interpretation and management of requirements throughout the system life cycle. There are several useful attributes to consider:

- rationale,
- trace to source or parent,
- system verification success criteria,
- owner,
- category,
- status,
- criticality, and
- priority.

The use of the rationale attribute helps communicate why the requirement is needed, any assumptions made, the source of numbers, the results of related design studies, or any other related supporting information. This supports further requirements analysis and

decomposition, as well as identifying the source of any requirement value. Defining the system verification success criteria helps ensure the requirement is well-formed, is verifiable, and aids in the planning for the project's verification program (INCOSE NRM 2022).

Categorizing Requirements

It is useful to organize the requirements into categories, grouping similar types of requirements together within a set. An example set of categories is shown in Table 1. While organizations may have different categorizations, for the set of requirements to be complete *each* category topic in Table 1 must be addressed.

Table 1. Example Types of Requirement Categories. (Derived from the INCOSE Needs and Requirements Manual)

Category	Description
Function/ Performance	The primary functions and associated performance that the SoI needs to perform in terms of its intended use. The functions address the capabilities and features the stakeholders expect the SoI to have; performance addresses how well, how many, how fast attributes of the function. Many of the primary functions involve interactions (interfaces) between the SOI and systems external to the SOI. All critical and high priority needs would be included in this category.
Fit/Operational	Requirements dealing with functions that deal with a secondary or enabling capabilities, functions, and interactions between the SoI and external systems needed for the system to accomplish its primary functions. This includes functions concerning the ability of the system to interface with, interact with, connect to, operate within, and become an integral part of the macro system it is a part. Fit includes human system interactions and interfaces as well as both the induced and natural environments (conditions of operations, transportation, storage, maintenance). For example, needs associated with safety, security, power, cooling, transportation and handling, storage, maintenance, and disposal.
Form	Physical Characteristics. The shape, size, dimensions, mass, weight, and other observable parameters and characterizes that uniquely distinguish a system. For software, form could address programming language, lines of code, memory requirements.
Quality	Fitness for use. For example, various “-ilities” such as reliability, testability, operability, availability, maintainability, operability, supportability, manufacturability, and interoperability.
Compliance	Conformance with design and construction standards and regulations.

See Table 2 for example requirement statements,

attributes and categories.

Table 2. Example Requirement Statements. (SEBoK Original)

Reqt ID	Requirement Title	Requirement	Rationale	Category
R1	Variable Temperature Settings	The Coffee Maker shall have two user-selectable settings for coffee brewing: Warm: 96°C +/- 2°C, Hot: 100°C+/- 2°C.	Based on focus group inputs for selectable temperature; values are from analysis associated with consumer research surveys and safety regulations.	Function/Performance
R2	Prohibit Brew if Container Missing	If coffee container is not fully inserted into the brew location, the Coffee Maker shall prohibit brew function.	Protective measure related to off-nominal use case.	Function/Performance
R3	Operational Life	The Coffee Maker shall have an operational life greater than or equal to 3 years.	Analysis shows expected operational service of 1,000 hours over three years of usage, ensuring appliance lasts through the warranty period.	Quality
R4	User Inputs	The Coffee Maker shall limit the number of user inputs to: Power On, Brew Temperature, Brew Size, Brew Start.	User inputs are assessed from focus groups to minimum set of inputs to achieve coffee maker full set of life cycle concepts.	Fit/Operation

Requirements At Levels Within the Hierarchy

Requirements definition is an iterative and recursive

process performed concurrently with the Architecture Definition Process. Upon determination of the system hierarchy, a requirement tree can be established showing the system level requirements and the supporting requirements at the lower-level elements of the hierarchy. Figure 3 shows an example flow from system requirement to lower-level requirements, and an example requirement tree is shown in Figure 4.

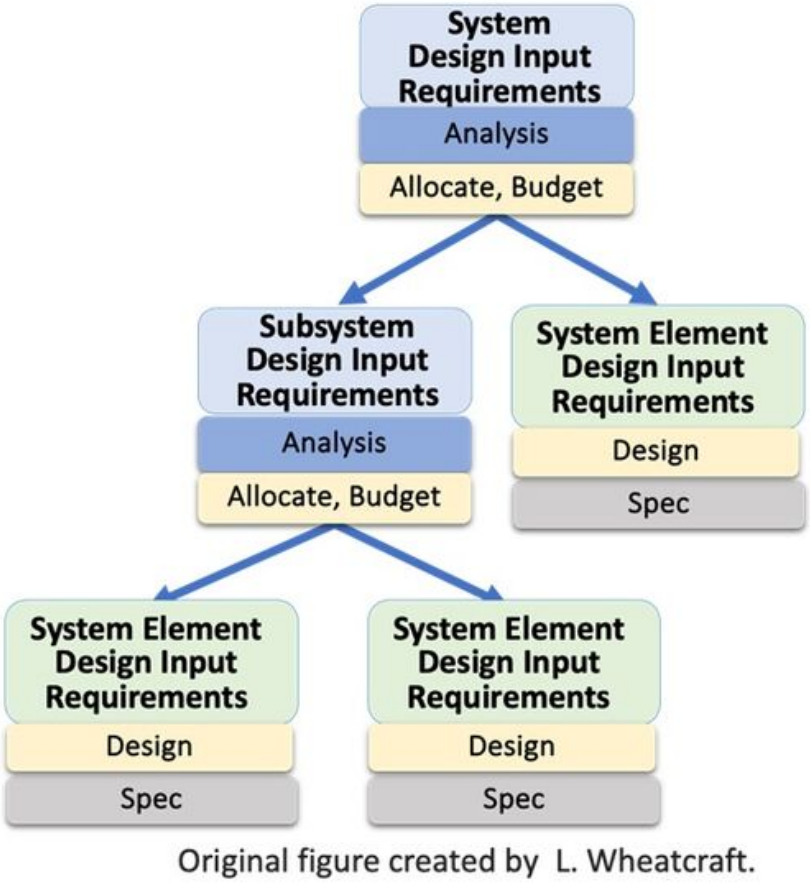

Figure 3. Example Flow of System Requirements with to Lower-level Requirements. This figure is reprinted with permission of Lou Wheatcraft. All other rights are reserved by the copyright owner.

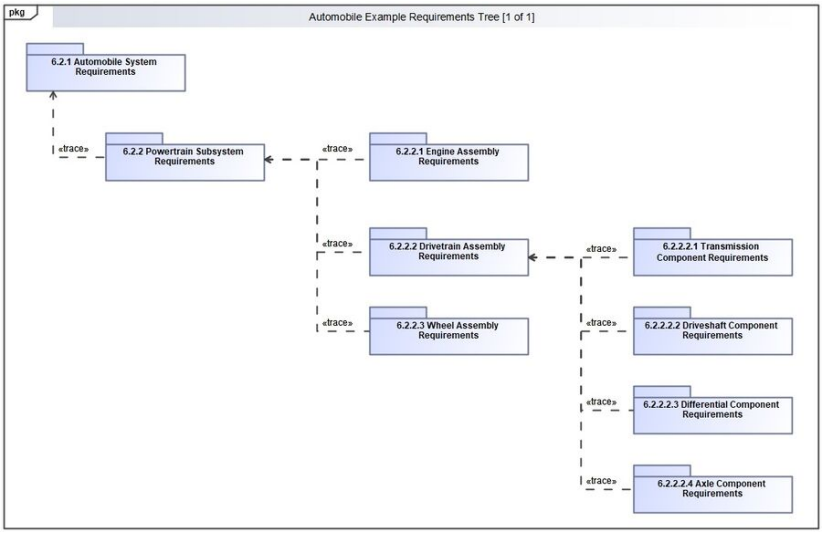

Figure 4. Example Requirement Tree using a SysML Package Diagram. (SEBoK Original)

Allocation is the process by which the requirements at one level of the physical architecture are assigned to those entities at the next lower level of the architecture that have a role in the implementation of the allocated requirement. This involves an analysis where the project team determines what "role", if any, each subsystem or system element at the next level of the architecture has in the implementation of the requirement being allocated. Requirements generated at the lower level are referred to as child requirements of an allocated parent requirement.

An important concept associated with allocation is budgeting. Budget refers to the total value of a parameter defined at the system level. The system requirement value may be decomposed to ensure the lower-level elements contribute their portion, enabling the system level to achieve its total value. Budgeted quantities result in requirements that have a dependency - a change in one will result in the need to change another. Budgeted values can include mass (weight), power usage, bandwidth, time, and quality attributes).

Principles Governing System Requirements

Addressing Interfaces and Interactions

For each part of the architecture, an external interface diagram provides the ability to address interactions between the SoI and external systems (Figure 5).

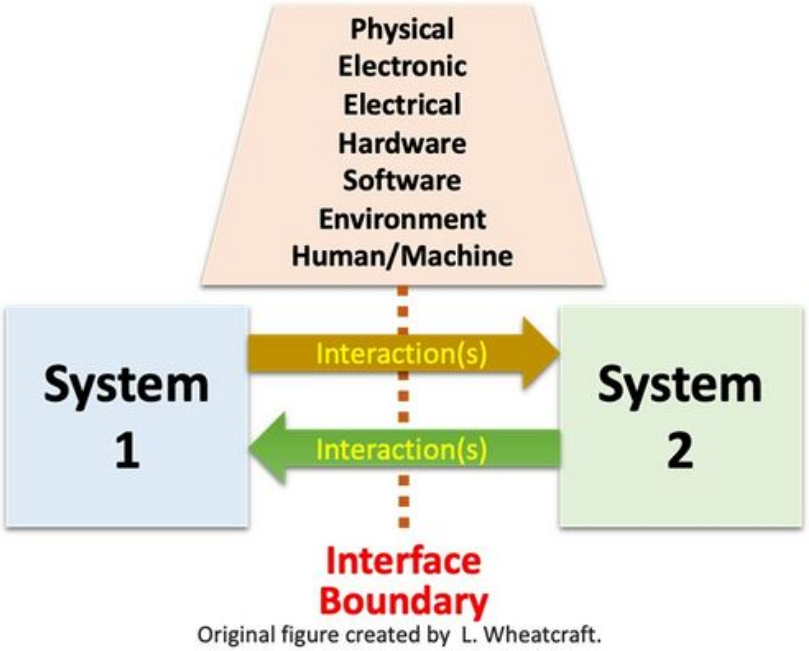

Figure 5. Interfaces Address Interactions between Systems.

The phrase “interface requirement” refers to the specific form or template for a functional requirement that deals with an interaction of a system across an interface boundary with another system. Writing interface requirements is a three-step process (INCOSE NRM 2022):

1. Identify the interface boundaries and interactions across those boundaries.
2. Define the interactions across the interface boundaries (the characteristics of what is crossing the boundary, and media involved).
3. Write the interface requirements.

Example interface requirement:

- The System shall send telemetry to the Ground System as defined in ICD 123, Table X.

Model Form of Requirements

Requirements can be recorded and managed via a document, database, or in the form of a model:

- Models provide the capability to address completeness in terms of functions, inputs to those functions, sources of those inputs, outputs, and customers (destinations/users) for those outputs.

- Models help facilitate communication by making complex systems and processes easier to understand (a picture is worth a thousand words….).
- Models enable the capture of interdependencies of requirements to other aspects of the system dataset, such as inputs, needs, architecture, test cases, etc.
- Models enable quantitative analyses and execution of simulations.

In MBSE, the model form of requirements can be expressed as a diagram (Figure 6) or within a table in the modeling tool (Figure 7), using well-formed textual statements as part of a requirement model element. Refer to Requirements Management for further guidance on usage of tools associated with textual and model requirements.

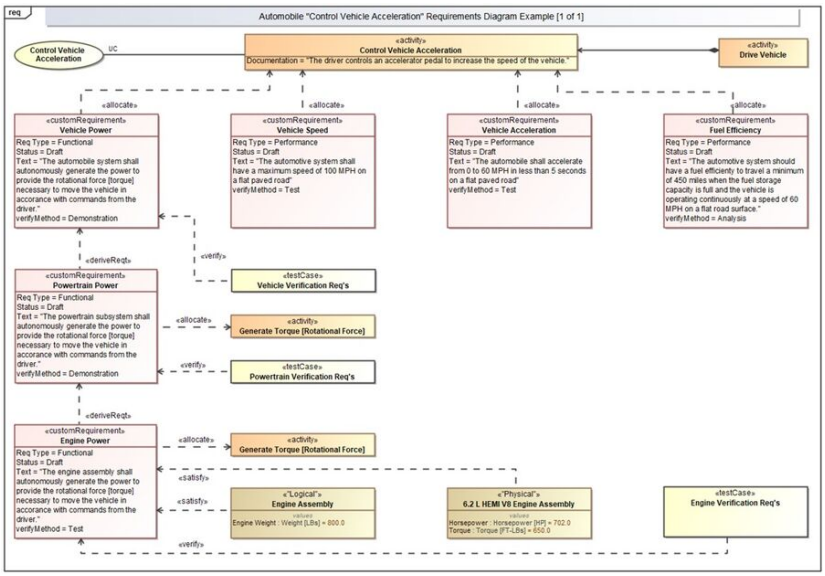

Figure 6. Example of a Model Based Requirement Diagram. (SEBoK Original)

Name	Text Type	Derived From	Req Type	Text	Status	Function Allocation	Satisfied By	Verify Method	Documentation
17 Powertrain Requirements	Header								Powertrain Subsystem Specification
17.1 Powertrain Power	Requirement	12.1 Vehicle Power	Functional	The powertrain subsystem shall autonomously generate the power to provide the rotational force [torque] necessary to move the vehicle in accorance with commands from the driver.	Draft	Generate Torque [Rotational Force]	Engine Assembly, 6.2 L HEMI V8 Engine Assembly	Demonstration	
17.2 Powertrain Speed	Requirement	12.2 Vehicle Speed	Performance	The powertrain subsystem shall generate sufficient power to maintain a maximum vehicle speed of 100 MPH on a flat paved road	Draft	Generate Torque [Rotational Force]	Engine Assembly, 6.2 L HEMI V8 Engine Assembly	Test	
17.3 Powertrain Acceleration	Requirement	12.3 Vehicle Acceleration	Performance	Te powertrain subsystem shall generate sufficient power to accelerate the vehicle from 0 to 60 MPH in less than 5 seconds on a flat paved road	Draft	Generate Torque [Rotational Force]	Engine Assembly, 6.2 L HEMI V8 Engine Assembly	Test	

Figure 7. Example of a Model Based Requirement Table. (SEBoK Original)

Addressing Unknowns

When initially generating a set of requirements, there is often a set of parameters which may require future analysis to determine their actual value. One common method to do this is to use a placeholder in the requirement statement to indicate the requirement is

drafted, yet further work is needed before it is considered complete. When a value is unknown, a common approach is to use the "To Be Determined" (TBD) indication within the requirement statement, in place of an actual value. When a value is determined but confidence is low (e.g., feasibility has not been assessed), a method to reflect this is to use the "To Be Resolved" (TBR) indication next to the value. Together, TBRs and TBDs are referred to generically as "TBXs". Keeping track of TBXs allows for awareness of the maturity of the requirements throughout the life cycle and is a valuable metric to evaluate completion and maturity of the integrated set of needs and set of design input requirements.

TBXs can lead to risk if not addressed by the project team prior to significant work in establishing the design solution. TBX management is the process of identifying the TBXs in a requirement set, establishing the work needed to resolve closure of the issue, identifying the owner of the work, and systematically iterating through and completing the actions to enable removal of these placeholders in the requirements.

Traceability

Traceability is a powerful way to manage the design input requirements, especially across levels and across subsystems and system elements within a specific level as well as manage the requirements across the life cycle. Traceability helps ensure the requirements trace to their needs, and that they are correct and complete. Traceability also helps identify allocation of requirements to lower-level system elements of the architecture. Traceability is used to access any impacts of changes to requirements or other types of data across the life cycle, enabling insight into whether that change is beneficial or costly.

Defining a traceability relationship model at the beginning of the project enables an understanding of which relationships will be established and managed via traceability throughout the development of the SoI. An example traceability model is shown in Figure 8 (INCOSE NRM 2022).

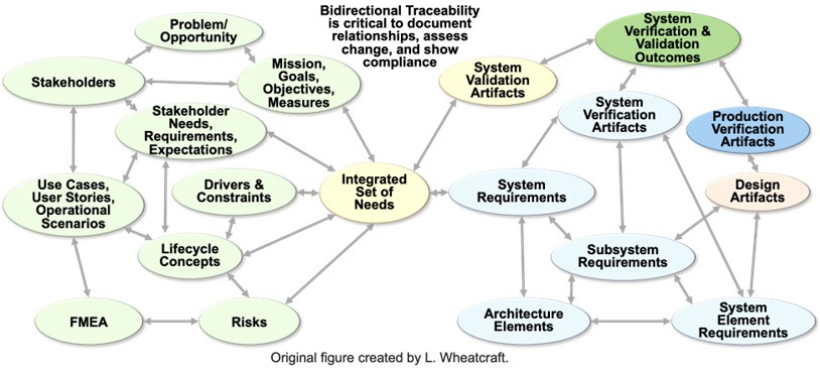

Figure 8. Example Traceability Model. This figure is reprinted with permission of Lou Wheatcraft. All other rights are reserved by the copyright owner.

Planning for System Verification

A best practice is to do early planning for how the project will perform system verification. Defining the system verification success criteria, strategy, and method when the requirements are generated helps ensure the requirement statements are worded properly, are within the scope of the project plans, and demonstrate they have the characteristic of "verifiable". This information can be captured within the attributes defined for the requirement.

Assessing the Requirements

Requirements should be assessed to gauge whether they are well-formed and that they meet the intent of the needs from which they were transformed. Concepts behind assessment of the requirements are captured with the terms "requirement verification" and "requirement validation", which are not the same concepts described for system verification and system validation.

- Requirement verification activities address: Are the requirements well-formed, i.e., do the individual requirements and sets of requirements have the characteristics and follow the rules defined in the organizations criteria for well-formed requirements?
- Requirement validation activities address: Do we have the right requirements, i.e., do the requirements clearly communicate the intent of the needs, parent requirements, and other sources from which they were transformed, in a language understandable by the architectural definition, design definition, and manufacturing/coding teams? Validation addresses if

the requirements are correct and are also achievable.

There are several methods that can be used to verify and validate requirements, such as standard peer review techniques and comparison of each requirement against a set of requirements characteristics and the integrated set of needs. For requirements communicated in a textual form, guidelines exist for writing well-formed requirements; they include recommendations about the syntax of requirements statements, wording (exclusions, representation of concepts, etc.), and characteristics (specific, measurable, achievable, feasible, testable, etc.) (INCOSE GtWR 2023).

System Requirement Definition Artifacts

The System Requirements Definition process results in a variety of artifacts:

- well-formed requirements recorded in models, databases, or documentation,
- requirement tree,
- traceability of requirements to other data,
- defined interactions across boundaries,
- performance budgets, and
- initial system verification plans.

Requirements Management

Requirements Management (RM) is performed to ensure alignment of the system, subsystem, and system element requirements with other representations, analyses, and artifacts of the system across the life cycle. It includes providing an understanding of the requirements, obtaining commitment, managing changes, maintaining bi-directional traceability among the requirements and with the rest of the system definition, and alignment with project resources and schedule. The Requirements Management article provides additional guidance on methods for performing requirements management.

References

Works Cited

INCOSE. 2022. *INCOSE Needs and Requirements Manual (NRM)*, version 1.1. INCOSE-TP-2021-002-01.

INCOSE. 2023. *INCOSE Guide to Writing Requirements (GtWR)*, version 4. INCOSE-TP-2006-004-04.

Primary References

ISO/IEC/IEEE. 2018. *Systems and Software Engineering - Requirements Engineering*. Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC)/ Institute of Electrical and Electronics Engineers (IEEE), ISO/IEC/IEEE 29148.

ISO/IEC/IEEE. 2023.*Systems and Software Engineering - System Life Cycle Processes.* Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC), Institute of Electrical and Electronics Engineers (IEEE). ISO/IEC/IEEE 15288:2023.

INCOSE. 2023. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, version 5.0. Hoboken, NJ, USA: John Wiley and Sons, Inc, ISBN: 978-1-118-99940-0.

INCOSE. 2022. *INCOSE Needs and Requirements Manual (NRM)*, version 1.1. INCOSE-TP-2021-002-01.

INCOSE. 2022. *INCOSE Guide to Needs and Requirements (GtNR)*, version 1. INCOSE-TP-2021-003-01.

INCOSE. 2023. *INCOSE Guide to Writing Requirements (GtWR)*, version 4. INCOSE-TP-2006-004-04.

Additional References

INCOSE. 2022. *INCOSE Guide to Verification and Validation (GtVV)*, version 1. INCOSE-TP-2021-004-01.

Relevant Videos

- INCOSE Requirements Working Group YouTube Channel

< Previous Article | Parent Article | Next Article >
SEBoK v. 2.10, released 06 May 2024

nition&oldid=71621"