

System Validation

System Validation

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: Alan Faisandier, **Contributing Author:** Rick Adcock

System Validation is a set of actions used to check the compliance of any element (a system element, a system, a document, a service, a task, a system requirement, etc.) with its purpose and functions. These actions are planned and carried out throughout the life cycle of the system. Validation is a generic term that needs to be instantiated within the context it occurs. When understood as a process, validation is a transverse activity to every life cycle stage of the system. Particularly during the development cycle of the system, the validation process is performed in parallel with the system definition and system realization processes and applies to any activity and product resulting from this activity. The validation process is not limited to a phase at the end of system development, but generally occurs at the end of a set of life cycle tasks or activities, and always at the end of each milestone of a development project. It may be performed on an iterative basis on every produced engineering element during development and may begin with the validation of the expressed stakeholder needs and requirements. When the validation process is applied to the system when completely integrated, it is often called *final validation*. It is important to remember that while system validation is separate from verification, the activities are complementary and intended to be performed in conjunction.

Contents

Definition and Purpose

Principles

- Concept of Validation Action

 - Why Validate?

 - What to Validate?

- Validation versus Verification

- Validation, Final Validation, and Operational Validation

- Verification and Validation Level per Level

- Verification Actions and Validation Actions Inside and Transverse to Levels

 - Verification and Validation Strategy

Process Approach

- Purpose and Principles of the Approach

- Activities of the Process

- Artifacts and Ontology Elements

- Methods and Techniques

Practical Considerations

- Pitfalls

- Proven Practices

References

- Works Cited

- Primary References

- Additional References

Definition and Purpose

Validation is the confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled. With a note added in ISO 9000:2005: *validation is the set of activities that ensure and provide confidence that a system is able to accomplish its intended use, goals, and objectives (i.e., meet stakeholder requirements) in the intended operational environment* (ISO 2005).

The purpose of validation, as a generic action, is to establish the compliance of any activity output as compared to inputs of the activity. It is used to provide information and evidence that the transformation of inputs produced the expected and *right* result. Validation is based on tangible evidence; i.e., it is based on information whose veracity can be demonstrated by factual results obtained from techniques or methods such as inspection, measurement, test, analysis,

calculation, etc. Thus, to validate a system (product, service, or enterprise) consists of demonstrating that it satisfies its system requirements and eventually the stakeholder's requirements depending on contractual practices. From a global standpoint, the purpose of validating a system is to acquire confidence in the system's ability to achieve its intended mission, or use, under specific operational conditions.

Principles

Concept of Validation Action

Why Validate?

The primary goal of systems engineering (SE) is to develop a solution that meets the needs and requirements of stakeholders. Validation is the process by which engineers ensure that the system will meet these needs and requirements.

A **validation action** is defined and then performed (see Figure 1, below).

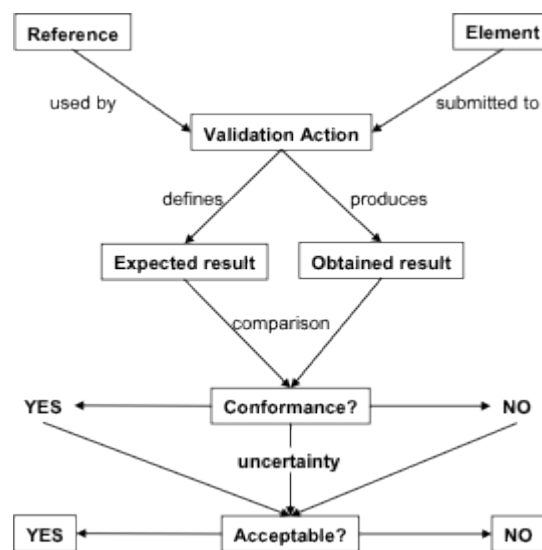


Figure 1. Definition and Usage of a Validation Action. (SEBoK Original)

A validation action applied to an engineering element includes the following:

- Identification of the element on which the validation action will be performed.
- Identification of the reference that defines the expected result of the validation action.

Performing the validation action includes the following:

- Obtaining a result by performing the validation action onto the submitted element.
- Comparing the obtained result with the expected result.
- Deducing the degree of compliance of the element.
- Deciding on the acceptability of this compliance, because sometimes the result of the comparison may require a value judgment to decide whether to accept the obtained result as compared to the relevance of the context of use.

Note: If there is uncertainty about compliance, the cause could come from ambiguity in the requirements.

What to Validate?

Any engineering element can be validated using a specific reference for comparison, such as stakeholder requirements, system requirements, functions, system elements, documents, etc. Examples are provided in Table 1 below:

Table 1. Examples of Validated Items (SEBoK Original)

Items	Explanation for Validation
Document	To validate a document is to make sure its content is compliant with the inputs of the task that produced the document.
Stakeholder Requirement and System Requirement	To validate a stakeholder requirement is to make sure its content is justified and relevant to stakeholders' expectations, complete and expressed in the language of the customer or end user. To validate a system requirement is to make sure its content translates correctly and/or accurately a stakeholder requirement to the language of the supplier.
Design	To validate the design of a system (logical and physical architectures) is to demonstrate that it satisfies its system requirements.
System	To validate a system (product, service, or enterprise) is to demonstrate that the product, service, or enterprise satisfies its system requirements and/or its stakeholder requirements.
Activity	To validate an activity or a task is to make sure its outputs are compliant with its inputs.

Process

To validate a process is to make sure its outcomes are compliant with its purpose.

Validation versus Verification

The Verification versus Validation section of the System Verification article gives fundamental differences between the two concepts and associated processes. The Table 2 provides information to help understand these differences.

Table 2. Verification and Validation Differences (may vary with context). (SEBoK Original)

Point of View	Verification	Validation
Purpose of the Activity	Detect, identify faults/defects (supplier oriented)	Acquire confidence (end user oriented)
Idea behind the Term	Based on truth (objective/unbiased)	Based on value judgement (more subjective)
Level of Concern	Detail and local	Global in the context of use
Vision	Glass box (how it runs inside)	Black box (application of inputs provides the expected effect)
Basic Method	Fine-tooth comb	Traceability matrix
System (Product, Service, Enterprise)	"Done Right" (respects the state of the art); focus on (physical) characteristics	"Does Right" (produces the expected effect); focus on services, functions
Baseline Reference for Comparison (Product, Service, Enterprise)	System design	System requirements (and stakeholder requirements)
Order of Performance	First	Second
Organization of Activity	Verification actions are defined and/or performed by development/designer team	Validation actions are defined and/or performed by experts and external members to development/designer team

Validation, Final Validation, and

Operational Validation

System validation concerns the global system seen as a whole and is based on the totality of requirements (system requirements, stakeholders' requirements, etc.), but it is obtained gradually throughout the development stage in three non-exclusive ways:

- accumulating the results of verification actions and validation actions provided by the application of corresponding processes to every engineering element;
- performing final validation actions to the complete, integrated system in an industrial environment (as close as possible to the operational environment); and
- performing operational validation actions on the complete system in its operational environment (context of use).

Verification and Validation Level per Level

It is impossible to carry out only a single global validation on a complete, integrated complex system. The sources of faults/defects could be important, and it would be impossible to determine the causes of non-conformance manifested during this global check. Generally, the system-of-interest (SoI) has been decomposed during design in a set of layers of systems. Thus, every system and system element is verified, validated, and possibly corrected before being integrated into the parent system of the higher level, as shown in Figure 2.

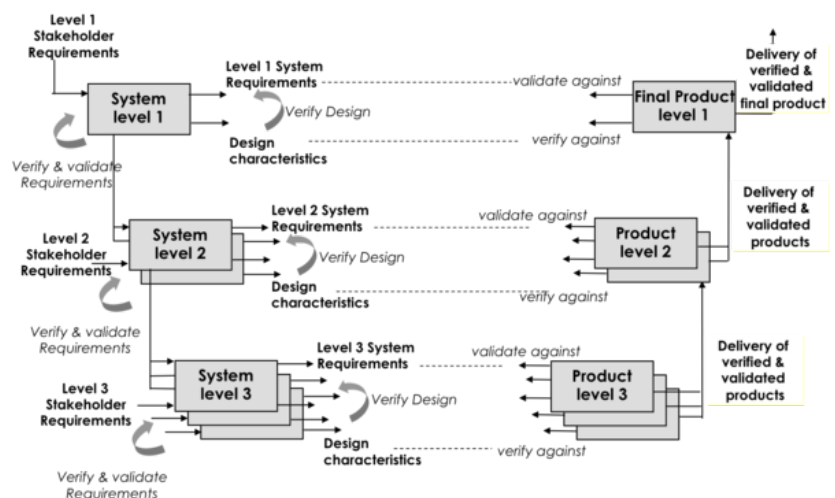


Figure 2. Verification and Validation Level per Level. (SEBoK Original)

As necessary, systems and system elements are partially integrated in subsets in order to limit the number of properties to be verified within a single step. For each level, it is necessary to perform a set of final validation actions to ensure that features stated at preceding levels are not damaged. Moreover, a compliant result obtained in a given environment can turn into a non-compliant result if the environment changes. Thus, as long as the system is not completely integrated and/or doesn't operate in the real operational environment, no result should be regarded as definitive.

Verification Actions and Validation Actions Inside and Transverse to Levels

Inside each level of system decomposition, verification actions and validation actions are performed during system definition and system realization. This is represented in Figure 3 for the upper levels, and in Figure 4 for the lower levels. Stakeholder requirements definition and operational validation make the link between the two levels of the system decomposition.

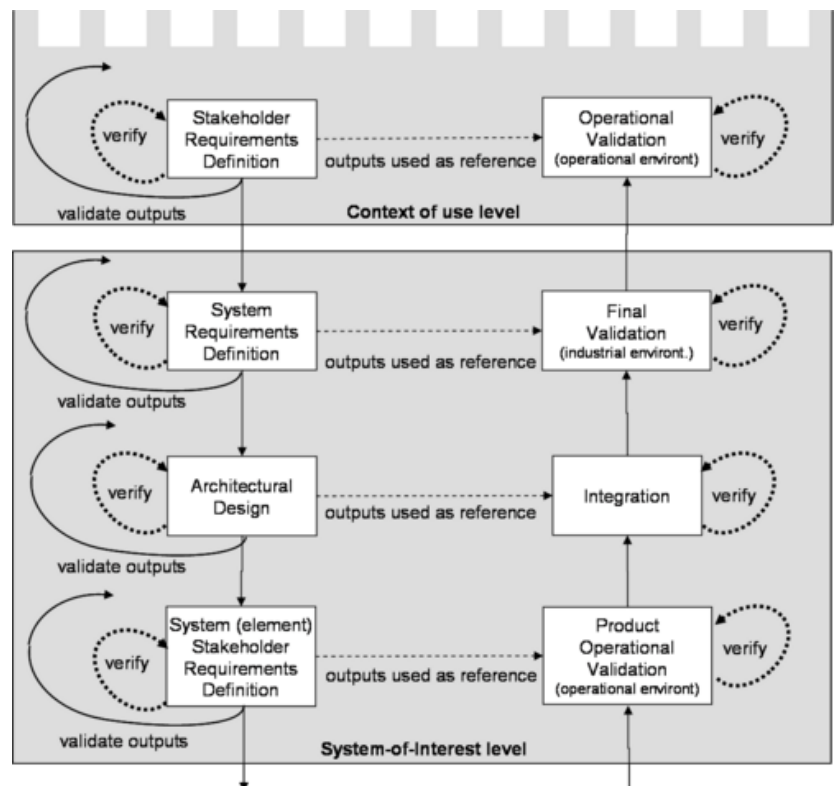


Figure 3. Verification and Validation Actions in Upper Levels of System Decomposition. (SEBoK Original)

Operational validation of system element requirements and products makes the link between the two lower

levels of the decomposition. See Figure 4 below.

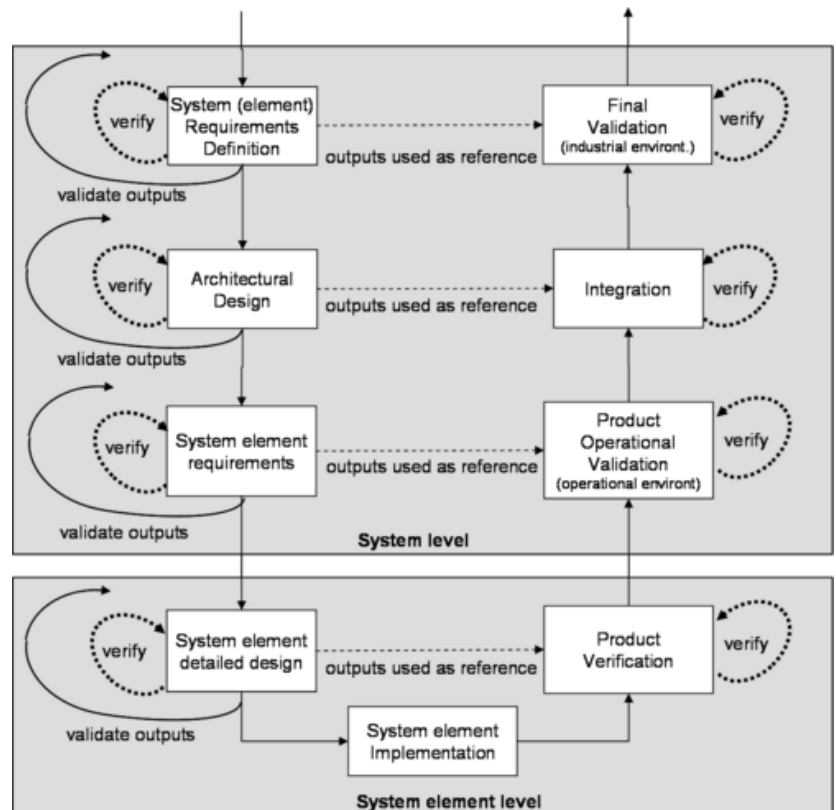


Figure 4. Verification and Validation Actions in Lower Levels of System Decomposition. (SEBoK Original)

Note: The last level of system decomposition is dedicated to the realization of system elements and the vocabulary and number of activities may be different from what is seen in Figure 4.

Verification and Validation Strategy

The difference between verification and validation is especially useful for elaborating on the integration strategy, the verification strategy, and the validation strategy. In fact, the efficiency of system realization is gained by optimizing the three strategies together to form what is often called the verification and validation strategy. This optimization consists of defining and performing the minimum number of verification and validation actions but detecting the maximum number of errors/faults/defects and achieving the maximum level of confidence in the system. The optimization takes into account the risks potentially generated if some verification actions or validation actions are excluded.

Process Approach

Purpose and Principles of the Approach

The purpose of the validation process is to provide objective evidence that the services provided by a system in use comply with stakeholder requirements and achieve its intended use in its intended operational environment (ISO/IEC/IEEE 15288 2015). The validation process performs a comparative assessment and confirms that the stakeholder requirements are correctly defined. Where variance is identified, it is recorded to guide future corrective actions. System validation is ratified by stakeholders (ISO/IEC/IEEE 15288 2015).

The validation process demonstrates that the realized end product satisfies its stakeholders' (customers' or other interested parties') expectations within the intended operational environments with validation performed by anticipated operators and/or users (NASA 2007, 1-360). Each system element, system, and the complete SoI are compared against their own applicable requirements (system requirements and stakeholder requirements). This means that the validation process is instantiated as many times as necessary during the global development of the system.

In order to ensure that validation is feasible, the implementation of requirements must be verifiable onto a defined element. It is essential to ensure that requirements are properly written, i.e., quantifiable, measurable, unambiguous, etc. In addition, verification/validation requirements are often written in conjunction with stakeholder and system requirements and provide a method for demonstrating the implementation of each system requirement or stakeholder requirement.

Generic inputs are references of requirements applicable to the submitted element. If the element is a system, inputs are system requirements and stakeholder requirements.

Generic outputs are the validation plan that includes validation strategy, selected validation actions, validation procedures, validation tools, validated elements or systems, validation reports, issue/trouble reports, and change requests on requirements or on the system.

Activities of the Process

Major activities and tasks performed during this process

include the following:

- Establish a validation strategy (often drafted in a validation plan). This activity is carried out concurrently to system definition activities:
 - Identify the validation scope that is represented by (system and/or stakeholder) requirements; normally, every requirement should be checked as the number of validation actions can be high.
 - Identify constraints according to their origin (technical feasibility, management constraints as cost, time, availability of validation means or qualified personnel, and contractual constraints that are critical to the mission) that limit or increase potential validation actions.
 - Define appropriate verification/validation techniques to be applied, such as inspection, analysis, simulation, review, testing, etc., depending on the best step of the project to perform every validation action according to constraints.
 - Consider a trade-off of what should be validated (scope) while taking into account all constraints or limits and deduce what can be validated objectively; selection of validation actions would be made according to the type of system, objectives of the project, acceptable risks, and constraints.
 - Optimize the validation strategy to define the most appropriate validation technique for every validation action, define necessary validation means (tools, test-benches, personnel, location, and facilities) according to the selected validation technique, schedule the execution of validation actions in the project steps or milestones, and define the configuration of elements submitted to validation actions (this is primarily about testing on physical elements).
- Perform validation actions, including the following tasks:
 - Detail each validation action. In particular, note the expected results, the validation technique to be applied, and the corresponding means necessary (equipment, resources, and qualified personnel).
 - Acquire validation means used during the system

definition steps (qualified personnel, modeling tools, mocks-up, simulators, and facilities), then those means used during integration and final and operational steps (qualified personnel, validation tools, measuring equipment, facilities, validation procedures, etc.).

- Carry out validation procedures at the right time, in the expected environment, with the expected means, tools, and techniques.
- Capture and record results obtained when performing validation actions using validation procedures and means.
- Analyze the obtained results and compare them to the expected results. Decide if they comply acceptably. Record whether the decision and status are compliant or not, and generate validation reports and potential issue/trouble reports, as well as change requests on (system or stakeholder) requirements as necessary.
- Control the process using following tasks:
 - Update the validation plan according to the progress of the project; in particular, planned validation actions can be redefined because of unexpected events.
 - Coordinate validation activities with the project manager regarding the schedule, acquisition of means, personnel, and resources. Coordinate with the designers for issue/trouble/non-conformance reports. Coordinate with the configuration manager for versions of physical elements, design baselines, etc.

Artifacts and Ontology Elements

This process may create several artifacts, such as:

- a validation plan (contains the validation strategy)
- a validation matrix (contains for each validation action, submitted element, applied technique, step of execution, system block concerned, expected result, obtained result, etc.)
- validation procedures (describe the validation actions to be performed, the validation tools needed, the validation configuration, resources, personnel, schedule, etc.)
- validation reports

- validation tools
- the validated element
- issue, non-conformance, and trouble reports
- change requests on requirements, products, services, and enterprises

This process utilizes the ontology elements of Table 3.

Table 3. Main Ontology Elements as Handled within Validation. (SEBoK Original)

Element	Definition
Validation Action	<p data-bbox="1007 640 1369 674">Attributes (examples)</p> <p data-bbox="901 689 1445 860">A validation action describes what must be validated (the element as reference), on which element, the expected result, the verification technique to apply, on which level of decomposition.</p>
Validation Procedure	<p data-bbox="901 913 1315 947">Identifier, name, description</p> <p data-bbox="901 960 1461 1099">A validation procedure groups a set of validation actions performed together (as a scenario of tests) in a given validation configuration.</p>
Validation Tool	<p data-bbox="901 1153 1477 1225">Identifier, name, description, duration, unit of time</p> <p data-bbox="901 1238 1477 1373">A validation tool is a device or physical tool used to perform validation procedures (test bench, simulator, cap/stub, launcher, etc.).</p>
Validation Configuration	<p data-bbox="901 1431 1315 1464">Identifier, name, description</p> <p data-bbox="901 1478 1461 1579">A validation configuration groups the physical elements necessary to perform a validation procedure.</p>
Risk	<p data-bbox="901 1632 1315 1666">Identifier, name, description</p> <p data-bbox="901 1680 1453 1852">An event having a probability of occurrence and a gravity degree on its consequence onto the system mission or on other characteristics (used for technical risk engineering).</p>
	Identifier, name, description, status

An argument that provides the justification for the selection of an engineering element.

Rationale

Identifier, name, description (rationale, reasons for defining a validation action, a validation procedure, for using a validation tool, etc.)

Methods and Techniques

The validation techniques are the same as those used for verification, but their purposes are different; verification is used to detect faults/defects, whereas validation is used to prove the satisfaction of (system and/or stakeholder) requirements.

The **validation traceability matrix** is introduced in the stakeholder requirements definition topic. It may also be extended and used to record data, such as a validation actions list, selected validation techniques to validate implementation of every engineering element (stakeholder and system requirements in particular), expected results, and obtained results when validation actions have been performed. The use of such a matrix enables the development team to ensure that selected stakeholder and system requirements have been checked, or to evaluate the percentage of validation actions completed.

Practical Considerations

Key pitfalls and good practices related to system validation are described in the next two sections.

Pitfalls

Some of the key pitfalls encountered in planning and performing system validation are provided in Table 4.

Table 4. Major Pitfalls with System Validation. (SEBoK Original)

Pitfall	Description
Start validation at the end of the project	A common mistake is to wait until the system has been entirely integrated and tested (design is qualified) to perform any sort of validation. Validation should occur as early as possible in the [product] life cycle (Martin 1997).

Use only testing	Use only testing as a validation technique. Testing requires checking products and services only when they are implemented. Consider other techniques earlier during design; analysis and inspections are cost effective and allow discovering early potential errors, faults, or failures.
Stop validation when funding is diminished	Stop the performance of validation actions when budget and/or time are consumed. Prefer using criteria such as coverage rates to end validation activity.

Proven Practices

Some good practices gathered from the references are provided in Table 5.

Table 5. Proven Practices with System Validation.
(SEBoK Original)

Practice	Description
Start Validation Plan Early	It is recommended to start the drafting of the validation plan as soon as the first requirements applicable to the system are known. If the writer of the requirements immediately puts the question to know how to validate whether the future system will answer the requirements, it is possible to: <ul style="list-style-type: none"> • detect the unverifiable requirements • anticipate, estimate cost, and start the design of validation means (as needed) such as test-benches, simulators • avoid cost overruns and schedule slippages
Verifiable Requirements	According to Buede, a requirement is verifiable if a "finite, cost-effective process has been defined to check that the requirement has been attained." (Buede 2009) Generally, this means that each requirement should be quantitative, measurable, unambiguous, understandable, and testable. It is generally much easier and more cost-effective to ensure that requirements meet these criteria while they are being written. Requirement adjustments made after implementation and/or integration are generally much more costly and may have wide-reaching redesign implications. There are several resources which provide guidance on creating appropriate requirements - see the system definition knowledge area, stakeholder requirements, and system requirements topics for additional information.

Document Validation Actions	It is important to document both the validation actions performed and the results obtained. This provides accountability regarding the extent to which system, system elements, and subsystems fulfill system requirements and stakeholders' requirements. These data can be used to investigate why the system, system elements, or subsystems do not match the requirements and to detect potential faults/defects. When requirements are met, these data may be reported to organization parties. For example, in a safety critical system, it may be necessary to report the results of safety demonstration to a certification organization. Validation results may be reported to the acquirer for contractual aspects or to internal company for business purpose.
Involve Users with Validation	Validation will often involve going back directly to the users to have them perform some sort of acceptance test under their own local conditions.
Involve	Often the end users and other relevant stakeholders are involved in the validation process.

The following are elements that should be considered when practicing any of the activities discussed as a part of system realization:

- Confusing verification and validation is a common issue. Validation demonstrates that the product, service, and/or enterprise as provided, fulfills its intended use, whereas verification addresses whether a local work product properly reflects its specified requirements. Validation actions use the same techniques as the verification actions (e.g., test, analysis, inspection, demonstration, or simulation).
- State who the witnesses will be (for the purpose of collecting the evidence of success), what general steps will be followed, and what special resources are needed, such as instrumentation, special test equipment or facilities, simulators, specific data gathering, or rigorous analysis of demonstration results.
- Identify the test facility, test equipment, any unique resource needs and environmental conditions, required qualifications and test personnel, general steps that will be followed, specific data to be collected, criteria for repeatability of collected data,

and methods for analyzing the results.

References

Works Cited

Buede, D. M. 2009. *The engineering design of systems: Models and methods*. 2nd ed. Hoboken, NJ: John Wiley & Sons Inc.

INCOSE. 2012. *INCOSE Systems Engineering Handbook*, version 3.2.2. San Diego, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TP-2003-002-03.2.2.

ISO/IEC/IEEE. 2015. *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. ISO/IEC/IEEE 15288:2015.

NASA. 2007. *Systems Engineering Handbook*. Washington, D.C.: National Aeronautics and Space Administration (NASA), NASA/SP-2007-6105, December 2007.

Primary References

INCOSE. 2012. *Systems Engineering Handbook*, version 3.2.2. San Diego, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TP-2003-002-03.2.2.

ISO/IEC/IEEE. 2015. *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. ISO/IEC 15288:2015.

NASA. 2007. *Systems Engineering Handbook*. Washington, D.C.: National Aeronautics and Space Administration (NASA), NASA/SP-2007-6105, December 2007.

Additional References

Buede, D.M. 2009. *The engineering design of systems:*

Models and methods. 2nd ed. Hoboken, NJ: John Wiley & Sons Inc.

DAU. February 19, 2010. *Defense Acquisition Guidebook (DAG)*. Ft. Belvoir, VA, USA: Defense Acquisition University (DAU)/U.S. Department of Defense.

ECSS. 2009. Systems engineering general requirements. Noordwijk, Netherlands: Requirements and Standards Division, European Cooperation for Space Standardization (ECSS), ECSS-E-ST-10C. 6 March 2009.

MITRE. 2011. "Verification and Validation." *Systems Engineering Guide*. Accessed 11 March 2012 at [[1]].

SAE International. 1996. *Certification considerations for highly-integrated or complex aircraft systems*. Warrendale, PA, USA: SAE International, ARP475.

SEI. 2007. *Capability maturity model integrated (CMMI) for development, version 1.2, measurement and analysis process area*. Pittsburg, PA, USA: Software Engineering Institute (SEI)/Carnegie Mellon University (CMU).

< [Previous Article](#) | [Parent Article](#) | [Next Article](#) >

SEBoK v. 2.10, released 06 May 2024

Retrieved from

"https://sandbox.sebokwiki.org/index.php?title=System_Validation&oldid=71533"

This page was last edited on 2 May 2024, at 22:39.