System Verification

Lead Authors: *John Snoderly, Alan Faisandier*

System Verification is a set of actions used to check the *correctness* of any element, such as a system element, a system, a document, a service, a task, a requirement, etc. These types of actions are planned and carried out throughout the life cycle of the system. Verification is a generic term that needs to be instantiated within the context it occurs. As a process, verification is a transverse activity to every life cycle stage of the system. In particular, during the development cycle of the system, the verification process is performed in parallel with the system definition and system realization processes and applies to any activity and any product resulting from the activity. The activities of every life cycle process and those of the verification process can work together. For example, the integration process frequently uses the verification process. It is important to remember that verification, while separate from validation, is intended to be performed in conjunction with validation.

☐

Contents

Definition and Purpose
Principles and Concepts
- Concept of Verification Action
  - Why Verify?
  - What to Verify?
- Verification versus Validation
- Integration, Verification, and Validation of the System

Process Approach
- Purpose and Principle 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
- Relevant Videos

Definition and Purpose

Verification is the confirmation, through the provision of objective evidence, that specified requirements have been fulfilled. With a note added in ISO/IEC/IEEE 15288, the scope of verification includes a set of activities that compares a system or system element against the requirements, architecture and design characteristics, and other properties to be verified (ISO/IEC/IEEE 2015). This may include, but is not limited to, specified requirements, design description, and the system itself.

The purpose of verification, as a generic action, is to identify the faults/defects introduced at the time of any transformation of inputs into outputs. Verification is used to provide information and evidence that the transformation was made according to the selected and appropriate methods, techniques, standards, or rules.

Verification is based on tangible evidence; i.e., it is based on information whose veracity can be demonstrated by factual results obtained from techniques such as inspection, measurement, testing, analysis, calculation, etc. Thus, the process of verifying a system (product, service, enterprise, or system of systems (SoS)) consists of comparing the realized characteristics or properties of the product, service, or enterprise against its expected design properties.

Principles and Concepts

Concept of Verification Action

Why Verify?

In the context of human realization, any human thought is susceptible to error. This is also the case with any engineering activity. Studies in human reliability have shown that people trained to perform a specific operation make around 1-3 errors per hour in best case scenarios. In any activity, or resulting outcome of an activity, the search for potential errors should not be neglected, regardless of whether or not one thinks they will happen or that they should not happen; the consequences of errors can cause extremely significant failures or threats.

A **verification action** is defined, and then performed, as shown in Figure 1.

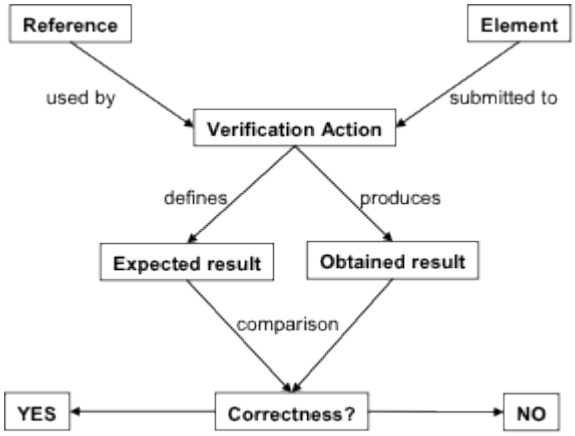

Figure 1. Definition and Usage of a Verification Action. (SEBoK Original)

The definition of a verification action applied to an engineering element includes the following:

- Identification of the element on which the verification action will be performed
- Identification of the reference to define the expected result of the verification action (see examples of reference in Table 1)

The performance of a verification action includes the following:

- Obtaining a result by performing the verification action onto the submitted element
- Comparing the obtained result with the expected result

- Deducing the degree of correctness of the element

What to Verify?

Any engineering element can be verified using a specific reference for comparison: stakeholder requirement, system requirement, function, system element, document, etc. Examples are provided in Table 1.

Table 1. Examples of Verified Items. (SEBoK Original)

Items	Explanation for Verification
Document	To verify a document is to check the application of drafting rules.
Stakeholder Requirement and System Requirement	To verify a stakeholder requirement or a system requirement is to check the application of syntactic and grammatical rules, characteristics defined in the stakeholder requirements definition process, and the system requirements definition process such as necessity, implementation free, unambiguous, consistent, complete, singular, feasible, traceable, and verifiable.
Design	To verify the design of a system is to check its logical and physical architecture elements against the characteristics of the outcomes of the design processes.
System	To verify a system (product, service, or enterprise) is to check its realized characteristics or properties against its expected design characteristics.
Aggregate	To verify an aggregate for integration is to check every interface and interaction between implemented elements.
Verification Procedure	To verify a verification procedure is to check the application of a predefined template and drafting rules.

Verification versus Validation

The term *verification* is often associated with the term *validation* and understood as a single concept of *V&V*. Validation is used to ensure that *one is working the right problem*, whereas verification is used to ensure that *one has solved the problem right* (Martin 1997). From an actual and etymological meaning, the term verification comes from the Latin *verus*, which means truth, and *facere*, which means to make/perform. Thus, verification means to prove that something is *true* or correct (a property, a characteristic, etc.). The term validation comes from the Latin *valere*, which means to become

strong, and has the same etymological root as the word *value*. Thus, validation means to prove that something has the right features to produce the expected effects. (Adapted from "Verification and Validation in plain English" (Lake INCOSE 1999).)

The main differences between the verification process and the validation process concern the references used to check the correctness of an element, and the acceptability of the effective correctness.

- Within verification, comparison between the expected result and the obtained result is generally binary, whereas within validation, the result of the comparison may require a judgment of value regarding whether or not to accept the obtained result compared to a threshold or limit.
- Verification relates more to one element, whereas validation relates more to a set of elements and considers this set as a whole.
- Validation presupposes that verification actions have already been performed.
- The techniques used to define and perform the verification actions and those for validation actions are very similar.

Integration, Verification, and Validation of the System

There is sometimes a misconception that verification occurs after integration and before validation. In most cases, it is more appropriate to begin verification activities during development or implementation and to continue them into deployment and use.

Once the system elements have been realized, they are integrated to form the complete system. Integration consists of assembling and performing verification actions as stated in the integration process. A final validation activity generally occurs when the system is integrated, but a certain number of validation actions are also performed parallel to the system integration in order to reduce the number of verification actions and validation actions while controlling the risks that could be generated if some checks are excluded. Integration, verification, and validation are intimately processed together due to the necessity of optimizing the strategy of verification and validation, as well as the strategy of integration.

Process Approach

Purpose and Principle of the Approach

The purpose of the verification process is to confirm that the system fulfills the specified design requirements. This process provides the information required to effect the remedial actions that correct non-conformances in the realized system or the processes that act on it - see ISO/IEC/IEEE 15288 (ISO/IEC/IEEE 2015).

Each system element and the complete system itself should be compared against its own design references (specified requirements). As stated by Dennis Buede, *verification is the matching of [configuration items], components, sub-systems, and the system to corresponding requirements to ensure that each has been built right* (Buede 2009). This means that the verification process is instantiated as many times as necessary during the global development of the system. Because of the generic nature of a process, the verification process can be applied to any engineering element that has conducted to the definition and realization of the system elements and the system itself.

Facing the huge number of potential verification actions that may be generated by the normal approach, it is necessary to optimize the verification strategy. This strategy is based on the balance between what must be verified and constraints, such as time, cost, and feasibility of testing, which naturally limit the number of verification actions and the risks one accepts when excluding some verification actions.

Several approaches exist that may be used for defining the verification process. The International Council on Systems Engineering (INCOSE) dictates that two main steps are necessary for verification: planning and performing verification actions (INCOSE 2012). NASA has a slightly more detailed approach that includes five main steps: prepare verification, perform verification, analyze outcomes, produce a report, and capture work products (NASA December 2007, 1-360, p. 102). Any approach may be used, provided that it is appropriate to the scope of the system, the constraints of the project, includes the activities of the process listed below in some way, and is appropriately coordinated with other activities.

Generic inputs are baseline references of the submitted element. If the element is a system, inputs are the logical

and physical architecture elements as described in a system design document, the design description of internal interfaces to the system and interfaces requirements external to the system, and by extension, the system requirements. **Generic outputs** define the verification plan that includes verification strategy, selected verification actions, verification procedures, verification tools, the verified element or system, verification reports, issue/trouble reports, and change requests on design.

Activities of the Process

To establish the verification strategy drafted in a verification plan (this activity is carried out concurrently to system definition activities), the following steps are necessary:

- Identify verification scope by listing as many characteristics or properties as possible that should be checked. The number of verification actions can be extremely high.
- Identify constraints according to their origin (technical feasibility, management constraints as cost, time, availability of verification means or qualified personnel, and contractual constraints that are critical to the mission) that limit potential verification actions.
- Define appropriate verification techniques to be applied, such as inspection, analysis, simulation, peer-review, testing, etc., based on the best step of the project to perform every verification action according to the given constraints.
- Consider a tradeoff of what should be verified (scope) taking into account all constraints or limits and deduce what can be verified; the selection of verification actions would be made according to the type of system, objectives of the project, acceptable risks, and constraints.
- Optimize the verification strategy by defining the most appropriate verification technique for every verification action while defining necessary verification means (tools, test-benches, personnel, location, and facilities) according to the selected verification technique.
- Schedule the execution of verification actions in the project steps or milestones and define the configuration of elements submitted to verification

actions (this mainly involves testing on physical elements).

Performing verification actions includes the following tasks:

- Detail each verification action; in particular, note the expected results, the verification techniques to be applied, and the corresponding means required (equipment, resources, and qualified personnel).
- Acquire verification means used during system definition steps (qualified personnel, modeling tools, mocks-up, simulators, and facilities), and then those used during the integration step (qualified personnel, verification tools, measuring equipment, facilities, verification procedures, etc.).
- Carry out verification procedures at the right time, in the expected environment, with the expected means, tools, and techniques.
- Capture and record the results obtained when performing verification actions using verification procedures and means.

The obtained results must be analyzed and compared to the expected results so that the status may be recorded as either *compliant* or *non-compliant*. Systems engineering (SE) practitioners will likely need to generate verification reports, as well as potential issue/trouble reports, and change requests on design as necessary.

Controlling the process includes the following tasks:

- Update the verification plan according to the progress of the project; in particular, planned verification actions can be redefined because of unexpected events.
- Coordinate verification activities with the project manager: review the schedule and the acquisition of means, personnel, and resources. Coordinate with designers for issues/trouble/non-conformance reports and with the configuration manager for versions of the physical elements, design baselines, etc.

Artifacts and Ontology Elements

This process may create several artifacts such as:

- verification plans (contain the verification strategy)
- verification matrices (contain the verification action, submitted element, applied technique, step of execution, system block concerned, expected result, obtained result, etc.)
- verification procedures (describe verification actions to be performed, verification tools needed, the verification configuration, resources and personnel needed, the schedule, etc.)
- verification reports
- verification tools
- verified elements
- issue / non-conformance / trouble reports
- change requests to the design

This process utilizes the ontology elements displayed in Table 2 below.

Table 2. Main Ontology Elements as Handled within Verification. (SEBoK Original)

Element	Definition Attributes (examples)
Verification Action	A verification action describes what must be verified (the element as reference) on which element, the expected result, the verification technique to apply, on which level of decomposition.
Verification Procedure	Identifier, name, description A verification procedure groups a set of verification actions performed together (as a scenario of tests) in a gin verification configuration.
Verification Tool	Identifier, name, description, duration, unit of time A verification tool is a device or physical tool used to perform verification procedures (test bench, simulator, cap/stub, launcher, etc.).
	Identifier, name, description

Verification Configuration	A verification configuration groups all physical elements (aggregates and verification tools) necessary to perform a verification procedure.
Risk	Identifier, name, description 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 in engineering). A risk is the combination of vulnerability and of a danger or a threat. An argument that provides the justification for the selection of an engineering element.
Rationale	Identifier, name, description (rationale, reasons for defining a verification action, a verification procedure, for using a verification tool, etc.)

Methods and Techniques

There are several verification techniques to check that an element or a system conforms to its design references or its specified requirements. These techniques are almost the same as those used for validation, though the application of the techniques may differ slightly. In particular, the purposes are different; verification is used to detect faults/defects, whereas validation is used to provide evidence for the satisfaction of (system and/or stakeholder) requirements. Table 3 below provides descriptions of some techniques for verification.

Table 3. Verification Techniques. (SEBoK Original)

Verification Technique	Description

Inspection	Technique based on visual or dimensional examination of an element; the verification relies on the human senses or uses simple methods of measurement and handling. Inspection is generally non-destructive, and typically includes the use of sight, hearing, smell, touch, and taste, simple physical manipulation, mechanical and electrical gauging, and measurement. No stimuli (tests) are necessary. The technique is used to check properties or characteristics best determined by observation (e.g. paint color, weight, documentation, listing of code, etc.).
Analysis	Technique based on analytical evidence obtained without any intervention on the submitted element using mathematical or probabilistic calculation, logical reasoning (including the theory of predicates), modeling and/or simulation under defined conditions to show theoretical compliance. Mainly used where testing to realistic conditions cannot be achieved or is not cost-effective.
Analogy or Similarity	Technique based on evidence of similar elements to the submitted element or on experience feedback. It is absolutely necessary to show by prediction that the context is invariant that the outcomes are transposable (models, investigations, experience feedback, etc.). Similarity can only be used if the submitted element is similar in design, manufacture, and use; equivalent or more stringent verification actions were used for the similar element, and the intended operational environment is identical to or less rigorous than the similar element.

Demonstration	Technique used to demonstrate correct operation of the submitted element against operational and observable characteristics without using physical measurements (no or minimal instrumentation or test equipment). Demonstration is sometimes called 'field testing'. It generally consists of a set of tests selected by the supplier to show that the element response to stimuli is suitable or to show that operators can perform their assigned tasks when using the element. Observations are made and compared with predetermined/expected responses. Demonstration may be appropriate when requirements or specification are given in statistical terms (e.g. mean time to repair, average power consumption, etc.).
Test	Technique performed onto the submitted element by which functional, measurable characteristics, operability, supportability, or performance capability is quantitatively verified when subjected to controlled conditions that are real or simulated. Testing often uses special test equipment or instrumentation to obtain accurate quantitative data to be analyzed.
Sampling	Technique based on verification of characteristics using samples. The number, tolerance, and other characteristics must be specified to be in agreement with the experience feedback.

Practical Considerations

Key pitfalls and good practices related to this topic are described in the next two sections.

Pitfalls

Some of the key pitfalls encountered in planning and performing System Verification are provided in Table 4.

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

Pitfall	Description

Confusion between verification and validation	Confusion between verification and validation causes developers to take the wrong reference/baseline to define verification and validation actions and/or to address the wrong level of granularity (detail level for verification, global level for validation).
No verification strategy	One overlooks verification actions because it is impossible to check every characteristic or property of all system elements and of the system in any combination of operational conditions and scenarios. A strategy (justified selection of verification actions against risks) must be established.
Save or spend time	Skip verification activity to save time.
Use only testing	Use only testing as a verification 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 verifications when funding is diminished	Stopping the performance of verification actions when budget and/or time are consumed. Prefer using criteria such as coverage rates to end verification activity.

Proven Practices

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

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

Practice	Description
Start verifications early in the development	The earlier characteristics of an element are verified in the project, the easier the corrections are to do and the consequences on schedule and cost will be fewer.
Define criteria ending verifications	Carrying out verification actions without limits generates a risk of drift for costs and deadlines. Modifying and verifying in a non-stop cycle until arriving at a perfect system is the best way to never supply the system. Thus, it is necessary to set limits of cost, time, and a maximum number of modification loops back for each verification action type, ending criteria (percentages of success, error count detected, coverage rate obtained, etc.).

Involve design responsible with verification	Include the verification responsible in the designer team or include some designer onto the verification team.

References

Works Cited

Buede, D.M. 2009. *The Engineering Design of Systems: Models and Methods.* 2nd ed. Hoboken, NJ, USA: 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.

Lake, J. 1999. "V & V in Plain English." International Council on Systems Engineering (INCOSE) 9th Annual International Symposium, Brighton, UK, 6-10 June 1999.

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

Primary References

INCOSE. 2012. *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, 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), December 2007. NASA/SP-2007-6105.

Additional References

Buede, D.M. 2009. *The Engineering Design of Systems: Models and Methods,* 2nd ed. Hoboken, NJ, USA: John Wiley & Sons Inc.

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

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

MITRE. 2011. "Verification and Validation." in *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. "Measurement and Analysis Process Area" in *Capability Maturity Model Integrated (CMMI) for Development, version 1.2.* Pittsburgh, PA, USA: Software Engineering Institute (SEI)/Carnegie Mellon University (CMU).

Relevant Videos

- Systems Engineering-Test, Evaluation, and Validation

< Previous Article | Parent Article | Next Article >
SEBoK v. 2.11, released 25 November 2024
