Verification Notes

29148-2018 6.5.2.2 Prepare for verification

Task: Select appropriate verification methods or techniques and associated criteria for every verification action. This activity is facilitated if you associate verification methods as requirements are created, these methods should be documented. This documentation may include requirements verification and traceability matrix or verification statements in a verification plan.

A good verification method should consider:

- How - identify which verification method to be applied (see list below).
- Who - identify the organization or person with the lead responsibility for performing the verification, such as a contractor, subcontractor, vendor, product team, or supplier.
- When - designate a time in the program plan when the verification is to be done. This should be event-based, and not a calendar date, accomplishment.
- Where - specify any unique venue and environment needed for the verification activity.

Four standard verification methods to use to obtain the objective evidence that the requirements have been fulfilled: inspection, analysis or simulation, demonstration, and test.

Inspection - An examination of items against applicable documentation to confirm compliance with requirements. It is used to verify properties best determined by examination and observation (e.g., - pain color, weight, etc.). The inspection typically includes the use of sight, hearing, smell, touch, and taste simple physical manipulation. Including documents and drawings is good practice to compare between what is required versus what is being inspected.

Analysis (including modeling and simulation) - Analysis is the use of analytical data or simulations under defined conditions to show theoretical compliance. It is used to test where realistic conditions cannot be achieved or are not cost-effective, and can be used as a means to establish the appropriate requirement, specification, or derived requirement is met by the proposed solution. Additionally, the analysis may be based on similarity by reviewing similar items prior to verification and confirming that its verification status can be transferred to the present system element (Note: Similarity can only be used if the items are similar in design, manufacture, and use). Identifying a generic name of the analysis, analytical or computer tools, or numeric methods; to validate the source of input data; and how raw data is to be analyzed is good practice.

Demonstration - is a qualitative exhibition of functional performance, with no or minimal instrumentation or test equipment. The demonstration uses a set of test activities to show that system or system element response to stimuli is suitable or to show that operators can perform with allocated functions when using the system. They may be appropriate when requirements or specifications are given in statistical terms (e.g, mean time to repair, average power, consumption, etc.). As good practice, state witnesses for the purpose of collecting evidence of success, what general steps are to be followed, and what special resources are needed.

Test - is an action in which the operability, supportability, or performance capability of an item is quantitatively verified under controlled conditions that are real or simulated. These verifications often use special test equipment or instrumentation to obtain accurate quantitative data for analysis. As good practice, state witnesses for the purpose of collecting evidence of success and identify test facility, test equipment, unique resource needs, and environmental conditions.

29148-2018 6.5.2.3 Manage results of verification

Task: Maintain traceability of the verified [software] system elements.

Requirements Traceability is frequently used for tracing a requirement back to the source of the requirement. In requirements traceability, verification methods and information are associated with the requirements to indicate how the system or system element is to be verified to show it meets the requirements.

29148-2018 9.6.19 Verification

Provide the verification approaches and methods planned to qualify the software. The information items for verification are recommended to be given in a parallel manner with the information items in 9.6.10 to 9.6.18

Could provide my list of requirements, design constraints, and software system attributes.

SEBOK System Verification

Verification is the confirmation, through the provision of objective evidence, that specified requirements have been fulfilled.

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.

What to verify currently:

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.

Definitions - Validation is used to ensure that one is working on the right problem, whereas **verification** is used to ensure that one has solved the problem right (Martin 1997).

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.

Methods and techniques -

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.

Proven Practices -

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.

MITRE Systems Engineering Guide p 356

This was pulled from the System-Level Requirements Checklist found on page 354 and was also used for the characteristics of the BLRB here: https://wangola.atlassian.net/wiki/spaces/DB/blog/2021/02/14/114196695/Requirements+Notes#Characteristics-of-this-set-of-system%2Fsoftware-requirements

Verification approaches for all system performance and sustainability requirements are
complete and appropriate. Every requirement must have a verification method identified.
If a requirement cannot easily be verified by a direct **inspection**, **measurement**, or one-time **demonstration** of the requirement, the verification requirement should include an expanded **test** criteria description to ensure that there is no disagreement later in the program. This can include describing the number of trials, statistical criteria to be used, conditions of the test such as simulated inputs, etc

The four fundamental methods of requirement verification

The four fundamental methods of verification are Inspection, Demonstration, Test, and Analysis.

Inspection is the nondestructive examination of a product or system using one or more of the five senses (visual, auditory, olfactory, tactile, taste).

The demonstration is the manipulation of the product or system as it is intended to be used to verify that the results are as planned or expected.

The test is the verification of a product or system using a controlled and predefined series of inputs, data, or stimuli to ensure that the product or system will produce a very specific and predefined output as specified by the requirements.

The analysis is the verification of a product or system using models, calculations, and testing equipment. The analysis allows someone to make predictive statements about the typical performance of a product or system based on the confirmed test results of a sample set or by combining the outcome of individual tests to conclude something new about the product or system.

This **link** provides good examples of the four fundamentals methods.

"Software Requirements" Wiegers book Chapter 17: Validating the requirements

Reviewing requirements -

Anytime someone other than the author of a work product examines the product for problems, a peer review is taking place. Reviewing requirements is a powerful technique for identifying ambiguous or unverifiable requirements, requirements that aren't defined clearly enough for the design to begin, and other problems.

Informal review approaches include:

- A peer deskcheck, in which you ask one colleague to look over your work product.
- A passaround, in which you invite several colleagues to examine a deliverable concurrently.
- A walkthrough, during which the author describes a deliverable and solicits comments on it.

Participants -

Ensure that you have all of the necessary people in an inspection meeting before proceeding. Otherwise, you might correct issues only to find out later that someone important disagrees with the change. The participants in an inspection should represent four perspectives:

- The author of the work product and perhaps peers of the author.
- People who are the sources of information that fed into the item being inspected.
- People who will do work based on the item being inspected.
- People who are responsible for interfacing systems will be affected by the item being inspected.

Inspection roles -

All participants in an inspection, including the author, look for defects and improvement opportunities. Some of the inspection team members perform the following specific roles during the inspection. These roles are:

- Author - The author created or maintains the work product being inspected.
- Moderator - The moderator plans the inspection with the author, coordinates the activities, and facilitates the inspection meeting.
- Reader - One inspector is assigned the role of the reader.
- Recorder - The recorder uses standard forms to document the issues raised and the defects found during the meeting.

Entry criteria -
Entry criteria set some clear expectations for authors to follow while preparing for an inspection. They also, keep the inspection team from spending time on issues that should be resolved prior to the inspection. Suggested inspection entry criteria for requirements documents:

- The document conforms to the standard template and doesn't have obvious spelling, grammatical, or formatting issues.
- Line numbers or other unique identifiers are printed on the document to facilitate referring to specific locations.
- All open issues are marked as TBD (to be determined) or accessible in an issue-tracking tool.
- The moderator didn't find more than three major defects in a ten-minute examination of a representative sample of the document

Inspection stages -

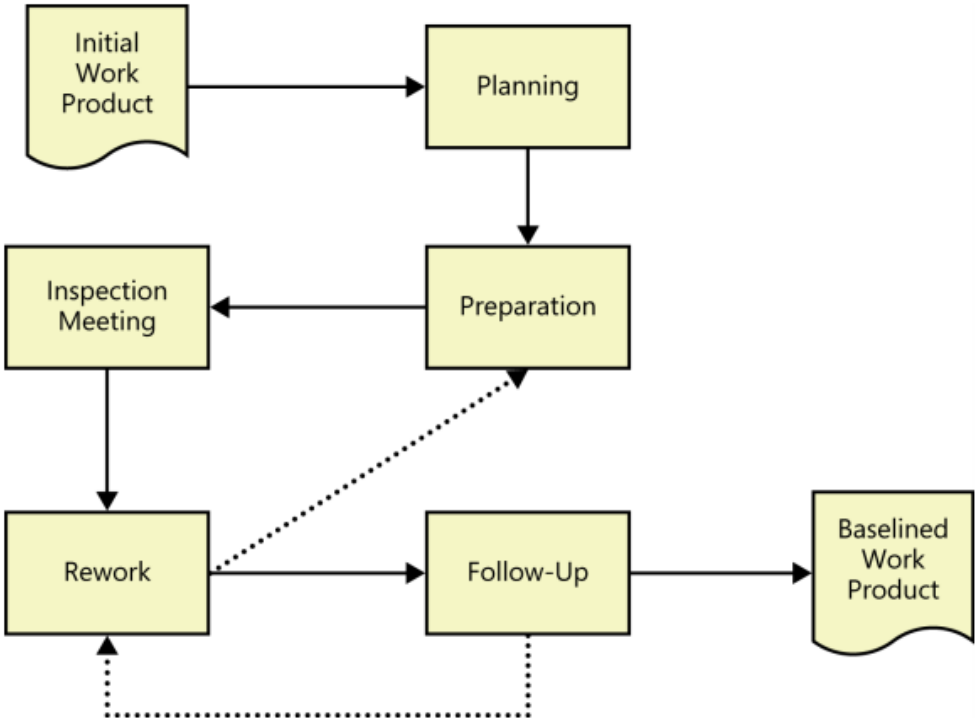

FIGURE 17-2 Inspection is a multistep process. The dotted lines indicate that portions of the inspection process might be repeated if reinspection is necessary because of extensive rework.

Description of each stage found on page 335.

Defect Checklist -

Completeness

- ❑ Do the requirements address all known customer or system needs?
- ❑ Is any needed information missing? If so, is it identified as TBD?
- ❑ Have algorithms intrinsic to the functional requirements been defined?
- ❑ Are all external hardware, software, and communication interfaces defined?
- ❑ Is the expected behavior documented for all anticipated error conditions?
- ❑ Do the requirements provide an adequate basis for design and test?
- ❑ Is the implementation priority of each requirement included?
- ❑ Is each requirement in scope for the project, release, or iteration?

Correctness

- ❑ Do any requirements conflict with or duplicate other requirements?
- ❑ Is each requirement written in clear, concise, unambiguous, grammatically correct language?
- ❑ Is each requirement verifiable by testing, demonstration, review, or analysis?
- ❑ Are any specified error messages clear and meaningful?
- ❑ Are all requirements actually requirements, not solutions or constraints?
- ❑ Are the requirements technically feasible and implementable within known constraints?

Quality Attributes

- ❑ Are all usability, performance, security, and safety objectives properly specified?
- ❑ Are other quality attributes documented and quantified, with the acceptable trade-offs specified?
- ❑ Are the time-critical functions identified and timing criteria specified for them?
- ❑ Have internationalization and localization issues been adequately addressed?
- ❑ Are all of the quality requirements measurable?

Organization and Traceability

- ❑ Are the requirements organized in a logical and accessible way?
- ❑ Are all cross-references to other requirements and documents correct?
- ❑ Are all requirements written at a consistent and appropriate level of detail?
- ❑ Is each requirement uniquely and correctly labeled?
- ❑ Is each functional requirement traced back to its origin (e.g., system requirement, business rule)?

Other Issues

- ❑ Are any use cases or process flows missing?
- ❑ Are any alternative flows, exceptions, or other information missing from use cases?
- ❑ Are all of the business rules identified?
- ❑ Are there any missing visual models that would provide clarity or completeness?
- ❑ Are all necessary report specifications present and complete?

FIGURE 17-4 A defect checklist for reviewing requirements documents.

Requirements review tips-

The following tips will improve your requirements reviews:

- **Plan the examination** - When someone asks you to review a document, the temptation is to begin at the top of page one and read it straight through. The consumers of the requirements specification won't be reading it front-to-back like a book; reviewers don't have to, either.
- **Start Early** - Begin reviewing sets of requirements when they are perhaps only 10 percent complete, not when you think they're "done." Detecting major defects early and spotting systemic problems in the way the requirements are being written is a powerful way to prevent—not just find—defects.
- **Allocate sufficient time** - Give reviewers sufficient time to perform the reviews, both in terms of actual hours to review (effort) and calendar time. They have other important tasks that the review has to fit around.
- **Provide context** - Give reviewers context for the document and perhaps for the project if they are not all working on the same project. Seek out reviewers who can provide a useful perspective based on their knowledge.

- **Set review scope** - Tell reviewers what material to examine, where to focus their attention, and what issues to look for. Suggest that they use a defect checklist like the one described in the preceding section.
- **Limit re-reviews** - Don't ask anyone to review the same material more than three times. He will be tired of looking at it and won't spot major issues after a third cycle because of "reviewer fatigue."
- **Prioritize review areas** - Prioritize for review those portions of the requirements that are of high risk or have functionality that will be used frequently. Also, look for areas of the requirements that have few issues logged already.

Requirements review challenges -

A peer review is both a technical activity and a social activity. Asking some colleagues to tell you what's wrong with your work is a learned—not instinctive—behavior. It takes time for a software organization to instill peer reviews into its culture. Following are some challenges that organizations face regarding requirements reviews:

- **Large requirements documents** - The prospect of thoroughly examining a several-hundred-page requirements document is daunting. You might be tempted to skip the review entirely and just proceed with construction—not a wise choice. To avoid overwhelming the review team, perform incremental reviews throughout requirements development. Identify high-risk areas that need a careful look through inspection, and use informal reviews for less risky material.
- **Large inspection teams** - Many project participants and customers hold a stake in the requirements, so you might have a long list of potential participants for requirements inspections. However, large review teams increase the cost of the review, make it hard to schedule meetings, and have difficulty reaching an agreement on issues. To deal with large inspection teams:
    - Make sure every participant is there to find defects, not to be educated or to protect a position.
    - Understand which perspective (such as customer, developer, or tester) each inspector represents. Several people who represent the same community can pool their input and send just one representative to the inspection meeting.
    - Establish several small teams to inspect the requirements in parallel and combine their defect lists, removing any duplicates. The results of parallel inspections are primarily additive rather than redundant.
- **Geographically separated reviewers** - Organizations often build products through the collaboration of geographically dispersed teams. This makes reviews more challenging. Teleconferencing doesn't reveal the body language and expressions of other reviewers like a face-to-face meeting does, but video conferencing can be an effective solution.
- **Unprepared reviewers** - One of the prerequisites to a formal review meeting is that the participants have examined the material being reviewed ahead of time, individually identifying their initial sets of issues. Without this preparation, you risk people spending the meeting time doing all of their thinking on the spot and likely missing many important issues.

Prototyping requirements -

All kinds of prototypes allow you to find missing requirements before more expensive activities like development and testing take place. Something as simple as a paper mock-up can be used to walk through use cases, processes, or functions to detect any omitted or erroneous requirements. Prototypes also help confirm that stakeholders have a shared understanding of the requirements. Proof-of-concept prototypes can demonstrate that the requirements are feasible.

Acceptance Criteria -

Working with customers to develop acceptance criteria provides a way to validate both the requirements and the solution itself. If a customer can' t express how she would evaluate the system's satisfaction of a particular requirement, that requirement is not clear enough.

Encourage users to use the SMART mnemonic - Specific, Measurable, Attainable, Relevant, and Time-sensitive-when defining acceptance criteria.

Acceptance tests -

Acceptance tests constitute the largest portion of the acceptance criteria. Creators of acceptance tests should consider the most commonly performed and most important usage scenarios when deciding how to evaluate the software's acceptability. Focus on testing the normal flows of the use cases and their corresponding exceptions, devoting less attention to the less frequently used alternative flows.

Verification of requirements found here: Verification

In case the table does not load properly -

Key	Summary	Verification Approach
DB-65	When the legal info database is updated, the BLRB shall mirror the update to the Boomi Legal info backup database in case of errors /corruption.	Analysis
DB-64		Inspection

	If a legal counsel clicks the edit question button, then BLRB shall display the edit mode page within 30 seconds.	
DB-63	The BLRB shall share the account information database with the Dell hub for single sign-on capabilities.	Analysis
DB-62	If the legal counsel submits a change to the list of legal questions without any issues, then the BLRB shall prompt the legal counsel to wait for other's approval.	Demonstration
DB-61	If the legal counsel submits a change to the list of legal questions, then the BLRB shall prompt the legal counsel if a submission issue occurs.	Demonstration
DB-60	The BLRB shall allow users to access the Dell Hub through the BLRB home page.	Inspection
DB-59	If the user does not provide the correct login information, then the BLRB shall prompt the user to enter valid login information.	Demonstration
DB-58	Every three months, the BLRB shall prompt the user to update/change their password to maintain user security.	Analysis
DB-57	The BLRB shall allow users to search for third-party products involved with Dell Boomi to depict correct legal information.	Demonstration
DB-56	The BLRB shall provide a disclaimer about the legal information.	Inspection
DB-50	While the user is logged in, the BLRB shall respond to any number of question entries.	Test
DB-49	When the user is logged in, the BLRB shall allow users to provide feedback to the BLRB to improve bot interaction.	Inspection
DB-48	The BLRB shall prevent unauthorized access.	Analysis
DB-47	BLRB shall permit legal counsels to remove legal answers to abide by current Boomi standards.	Test
DB-46	If the user clicks the view all questions button, then the BLRB shall display all current legal questions/answers.	Test
DB-45	When a legal question is not found, the BLRB shall display legal department contact information within 10 seconds.	Inspection
DB-44	When a legal question is entered, the BLRB shall display a correct legal response within 10 seconds.	Inspection
DB-43	BLRB shall permit legal counsels to update legal answers to avoid inaccurate responses.	Test
DB-42	When the user is not logged in, the BLRB shall display a login screen for username and password.	Demonstration
DB-41	Once the user is logged in, the BLRB shall prompt the user within 6 seconds to input a question.	Inspection