4/23/2022

Validation Report

For FGCU Complete

Kaden Carr
FGCU COMPLETE

Contents

How to Validate Requirements

The definition of requirement validation is “confirmation that requirements (individually and as a set) define the right system as intended by stakeholders”. There are many methods of validating these requirements. The point of this validation process is to show that the requirements meet the system intended by the stakeholder and these attributes are displayed by the prototype or the product. Different methods of ensuring the correctness of the requirements would involve processes such as stakeholder approval. An approval is an important process of validating that the requirements make the system the stakeholder intends.

Why Validate Requirements

Requirement validation is an important keystone in the process of development. Agreements can be made with stakeholders before the process of development that can objectively be verified. Requirements can be displayed in the product later which will validate what the stakeholder had asked. This process ensures integrity to plans throughout the process. The process of validation is also a process of quality control. It ensures that the stakeholders' needs have been met to the standards planned to standards set before development. A faulty requirement can cause large expense to the system in which the requirements are running from.

When to Validate Requirements

Validation of requirements happens before the development process to prevent any faulty errors from occurring caused from invalid requirements. This can be done once or multiple times before creating the product.

Review and Inspections

Review and inspection are quality control practices to ensure requirements and product match. This will allow the Software Requirements created by the Business Analysis and the stakeholder to be equivalent to what was agreed before development process. Inspection is a well defined and formal peer reviewed process which is divided into several stages. The phase of inspection involves:

- planning
- overview
- individual preparation
- inspection meeting
- rework
- follow-up to verify the changes made during rework.

The following are all the phases in a typical inspection process. The inspection process involves an inspection team which consists of a moderator(plans and leads inspection), reader(presents the product to the other team members), and the recorder(logs defects found and issues raised). Other participants involved with the inspection process may involve the author(author of requirements), a second skilled requirement analyst (the author is most likely a requirement analyst), the product manager, and the users. In the inspection process the professionals and the users are equally important. The product will be judged with users testing product so inspector can validate the product helping the user's needs. A suggested practice is to keep this team to a maximum of 8 people.

Prototyping to Validate Requirements

Prototyping can be a cost-effective method used to validate the stakeholder's requirements. The prototype will be showed to the stakeholders and judged against the requirements with demonstrations and tests that will be used to validate the requirements and the prototypes match. Prototypes do not have to be functioning in order to validate requirements. Prototypes created as design concepts or a proof of concept model works to validate the requirements can be met before effort is brought to meeting these requirements through development.

Acceptance test design

An acceptance test is an import process in validating the requirements. An acceptance test is a test which signifies that the product has the feature or function specified by the requirement in the requirements phase. Acceptance test will utilize end-users that will be used to test out the prototype which is a quality control practice. If the acceptance test proves the product is not up to standards the process restarts and more development is needed. An acceptance test is an overlap between the business side of the product and the development sides. It should validate the entire project. When the system changes, the tests can easily start failing. An acceptance test validates what the system does and not how it does it. The test will be coded algorithms to check the functionality/ output check.

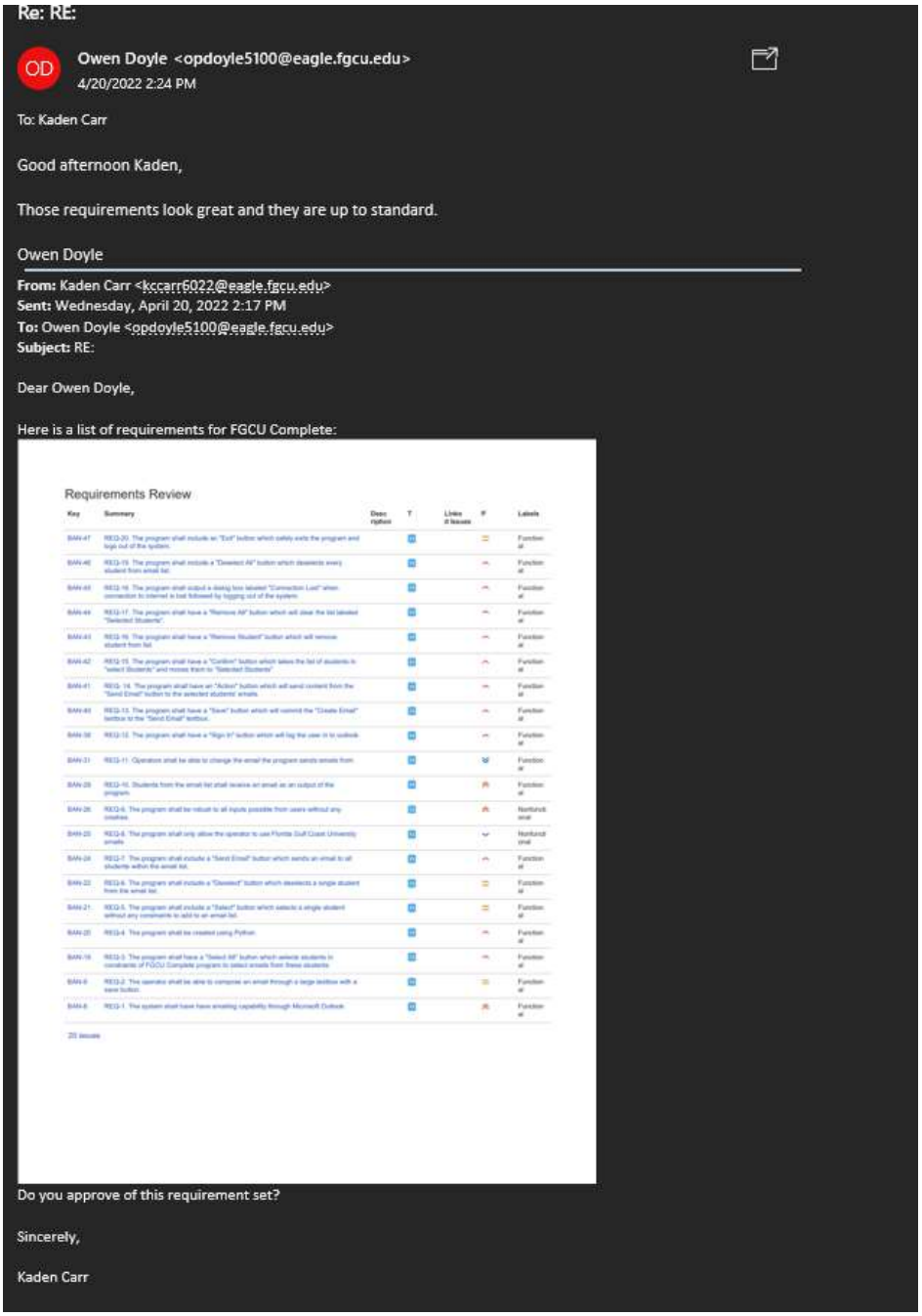
Requirements Engineering Process and Validation of Behavioral Requirements

Characteristics for the future requirement or feature the stakeholder wants to have is provided before development and ideally agreed upon with developers and stakeholders. These requirements will have behavioral features that involve necessary outcomes that are unambiguous, feasible, singular, complete, conforming, and verifiable. These supposed features have to meet the standards and can be verified afterwards. Validation of a behavioral requirement is ensuring that everyone interprets the requirement the same way ad the acceptance criteria that was agreed upon is met. The behavior must be specifically met and defined in a way

which is specifically objective. This objectively defined behavior can now be validated to this standard later down the line.

Validation of FGCU Complete

The process of verifying requirements for FGCU Complete involved writing the set of requirements that describe the system under the standards for a correct requirement. After this requirement set was composed along with a consistent prototype that follows the set of requirements it was signed off by an expert which validated the requirement set. The following email was the email of the requirement sign off. The requirement sign off was performed by Owen Doyle, a secondary analyst.



Re: RE:

OD Owen Doyle <opdoyle5100@eagle.fgcu.edu>
4/20/2022 2:24 PM

To: Kaden Carr

Good afternoon Kaden,

Those requirements look great and they are up to standard.

Owen Doyle

From: Kaden Carr <kccarr6022@eagle.fgcu.edu>
Sent: Wednesday, April 20, 2022 2:17 PM
To: Owen Doyle <opdoyle5100@eagle.fgcu.edu>
Subject: RE:

Dear Owen Doyle,

Here is a list of requirements for FGCU Complete:

Requirements Review

Key	Summary	Desc ription	T	Links d Issues	P	Labels
BAN-47	REQ-20: The program shall include an "Exit" button which safely exits the program and logs out of the system.					Function al
BAN-46	REQ-19: The program shall include a "Deselect All" button which deselects every student from email list.					Function al
BAN-45	REQ-18: The program shall output a dialog box labeled "Connection Lost" when connection to internet is lost followed by logging out of the system.					Function al
BAN-44	REQ-17: The program shall have a "Remove All" button which will clear the list labeled "Selected Students".					Function al
BAN-43	REQ-16: The program shall have a "Remove Student" button which will remove student from list.					Function al
BAN-42	REQ-15: The program shall have a "Confirm" button which takes the list of students in "select Students" and moves them to "Selected Students"					Function al
BAN-41	REQ- 14: The program shall have an "Action" button which will send content from the "Send Email" button to the selected students' emails.					Function al
BAN-40	REQ-13: The program shall have a "Save" button which will commit the "Create Email" textbox to the "Send Email" textbox.					Function al
BAN-39	REQ-12: The program shall have a "Sign In" button which will log the user in to outlook					Function al
BAN-31	REQ-11: Operators shall be able to change the email the program sends emails from				M	Function al
BAN-29	REQ-10: Students from the email list shall receive an email as an output of the program.				R	Function al
BAN-28	REQ-9: The program shall be robust to all inputs possible from users without any crashes				R	Nonfunct ional
BAN-25	REQ-8: The program shall only allow the operator to use Florida Gulf Coast University emails					Nonfunct ional
BAN-24	REQ-7: The program shall include a "Send Email" button which sends an email to all students within the email list.					Function al
BAN-22	REQ-6: The program shall include a "Deselect" button which deselects a single student from the email list.					Function al
BAN-21	REQ-5: The program shall include a "Select" button which selects a single student without any constraints to add to an email list.					Function al
BAN-20	REQ-4: The program shall be created using Python.					Function al
BAN-19	REQ-3: The program shall have a "Select All" button which selects students in constraints of FGCU Complete program to select emails from these students					Function al
BAN-8	REQ-2: The operator shall be able to compose an email through a large textbox with a save button.					Function al
BAN-6	REQ-1: The system shall have have emailing capability through Microsoft Outlook				R	Function al

20 issues

Do you approve of this requirement set?

Sincerely,

Kaden Carr

Additionally, a prototype for the FGCU Complete Emailing System is readably available with the set of requirements which consistently displays every functional requirement listed in the requirement set. What can also be used to verify the requirements was a requirement checklist displayed below.

Key	**Summary**	**Labels**	**P**

BAN-47	REQ-20. The program shall include an "Exit" button which safely exits the program and logs out of the system.	Functional	=
BAN-46	REQ-19. The program shall include a "Deselect All" button which deselects every student from email list.	Functional	^
BAN-45	REQ-18. The program shall output a dialog box labeled "Connection Lost" when connection to internet is lost followed by logging out of the system.	Functional	^
BAN-44	REQ-17. The program shall have a "Remove All" button which will clear the list labeled "Selected Students".	Functional	^
BAN-43	REQ-16. The program shall have a "Remove Student" button which will remove student from list.	Functional	^
BAN-42	REQ-15. The program shall have a "Confirm" button which takes the list of students in "select Students" and moves them to "Selected Students"	Functional	^
BAN-41	REQ- 14. The program shall have an "Action" button which will send content from the "Send Email" button to the selected students' emails.	Functional	^
BAN-40	REQ-13. The program shall have a "Save" button which will commit the "Create Email" textbox to the "Send Email" textbox.	Functional	^
BAN-39	REQ-12. The program shall have a "Sign In" button which will log the user in to outlook.	Functional	^
BAN-31	REQ-11. Operators shall be able to change the email the program sends emails from	Functional	⌄⌄
BAN-29	REQ-10. Students from the email list shall receive an email as an output of the program.	Functional	^^
BAN-26	REQ-9, The program shall be robust to all inputs possible from users without any crashes.	Nonfunctional	^^
BAN-25	REQ-8. The program shall only allow the operator to use Florida Gulf Coast University emails	Nonfunctional	⌄
BAN-24	REQ-7. The program shall include a "Send Email" button which sends an email to all students within the email list.	Functional	^
BAN-22	REQ-6. The program shall include a "Deselect" button which deselects a single student from the email list.	Functional	=
BAN-21	REQ-5. The program shall include a "Select" button which selects a single student without any constraints to add to an email list.	Functional	=

BAN-20	REQ-4. The program shall be created using Python	Functional	^
BAN-19	REQ-3. The program shall have a "Select All" button which selects students in constraints of FGCU Complete program to select emails from these students.	Functional	^
BAN-9	REQ-2. The operator shall be able to compose an email through a large textbox with a save button.	Functional	=
BAN-8	REQ-1. The system shall have have emailing capability through Microsoft Outlook.	Functional	^^

20 issues

Checklist Item	Status
The system-level technical requirements are traceable to the user requirements	Correct
Each system requirement describes something relevant: a function the system must perform, performance a function must provide, a constraint on the design, or a reference such as to an interface definition	Correct
The level of detail that the requirements provide about system functionality is appropriate. The requirements are sufficient to describe what the overall system must do, what its performance must be, and what constraints an engineer should consider. There are few requirements that specifically affect the design of only one component of the system. The major requirements drivers (e.g., those stressing the design) and associated risks should be identified	Correct
The requirements include any legal or regulatory constraints within which the system must perform. Example: There may be restrictions on the use or quantity of certain hazardous materials in a system	Correct
The requirements include enterprise architecture constraints within which the system must integrate (or toward which the system is desired to migrate). Requirements include appropriate open systems and modularity standards. Examples: DoD Net-Ready requirements, modular open system architecture concepts, Electronic Systems Center strategic technical plan goals	Correct
Environmental design requirements are specified. Example: A control unit may be in a controlled office environment and the other major components may be outdoors, thus two environments must be defined and associated with the functionality operating in each environment	Correct

All external interfaces for the system are included. Major internal interfaces may also be included if they are important to system modularity, or future growth in capability. These may include physical (mechanical fastening, electrical wiring, connectors), functional (mechanical stress transfer points, cooling, power sources, antennas, wire message formats, data exchanges), and software (software interface specifications, library calls, data formats, etc.). Remember that an internal interface between two subsystems that use a transport mechanism that is not part of the system is a hidden external interface. For example, two subsystems that communicate internally with each other over a sensitive but unclassified network as the internal interface (the data exchanged between them) and an external interface (the wiring and internet protocols to enable the data exchanges with the network).	Correct
Requirement statements use the word "shall" or "should." The word "shall" has meaning in contractual language and is enforceable legally. Other words like "will," "may," "should," and "must" may show intent but are not legally binding in contracts. In some situations, it may be desirable to use "should" to show the government's intent and preference while at the same time allowing flexibility and latitude. Example: "The system shall have a mean time between failures of greater than 500 hours."	Correct
Requirements statements are unambiguous. Terminology is clear without the use of informal jargon. Statements are short and concise.	Correct
Performance requirements statements (including logistics/sustainment/support) are quantifiable, testable, and/or verifiable. Avoid the phrase "shall not." It is very difficult to prove a negative. Avoid qualitative words like "maximize" or "minimize." They force an engineer to judge when the design is good enough. The user may think that the engineer did not "minimize enough" and get into a legal argument with the contractor. Note: Every user requirements document includes: "the system shall be easy to use" requirement. Talk to other MITRE staff for examples from other projects and seek out a human factors specialist for requirements wording that is suitable both for specifying these requirements and methodologies for verifying them. Avoid specific, one-point values when defining requirements. Use ranges (minimum of, more than, less than, maximum of, within, etc.) to accommodate appropriate interpretation. Using a single point value may cause arguments if the system is tested at that exact value only, or if a test appears to be successful from an intent perspective, but does not meet the exact value stated in the system requirement. Example: The system shall process a minimum of 100 transactions/sec. Example: The system shall be operable up to and including 30,000 ft. Example: The system shall operate in temperatures between 5 and 35 degrees Celsius.	Correct
If objective performance values are included as goals, ensure they are clearly identified and distinguished from firm requirements. User requirement documents refer to threshold requirements (those that must be provided), and objective requirements (better performance has value to the user, but not above the objective requirement). Example: The system shall detect and display up to 100 targets within the surveillance volume with a goal of detecting and displaying up to 125 targets.	Correct

The operational and support environment is described and defined. Example: The system shall be maintainable by an Air Force level 5 technician. Example: The system shall be reparable while in flight	Correct
The requirements include appropriate use of Government and industry specifications, standards, and guides. Only include them if they are relevant and ensure that the correct version is listed in a list of reference documents.	Correct
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 onetime 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.	Correct

The following checklist is a checklist to verify a credible requirement set. Additionally, the set of requirements is further verified by a secondary analyst approving the FGCU Complete requirement set listed above. The following two methods with a verifiable prototype validate the consistency, completeness, and correctness.