

WIKIPEDIA

Requirements management

Requirements management is the process of documenting, analyzing, tracing, prioritizing and agreeing on requirements and then controlling change and communicating to relevant stakeholders. It is a continuous process throughout a project. A requirement is a capability to which a project outcome (product or service) should conform.

Contents

Overview

Traceability

Requirements activities

Investigation

Feasibility

Design

Construction and test

Requirements change management

Release

Tooling

See also

References

Further reading

External links

Overview

The purpose of requirements management is to ensure that an organization documents, verifies, and meets the needs and expectations of its customers and internal or external stakeholders.^[1] Requirements management begins with the analysis and elicitation of the objectives and constraints of the organization. Requirements management further includes supporting planning for requirements, integrating requirements and the organization for working with them (attributes for requirements), as well as relationships with other information delivering against requirements, and changes for these.

The traceability thus established is used in managing requirements to report back fulfilment of company and stakeholder interests in terms of compliance, completeness, coverage, and consistency. Traceabilities also support change management as part of requirements management in understanding the impacts of changes through requirements or other related elements (e.g., functional impacts through relations to functional architecture), and facilitating introducing these changes.^[2]

Requirements management involves communication between the project team members and stakeholders, and

adjustment to requirements changes throughout the course of the project.^[3] To prevent one class of requirements from overriding another, constant communication among members of the development team is critical. For example, in software development for internal applications, the business has such strong needs that it may ignore user requirements, or believe that in creating use cases, the user requirements are being taken care of.

Traceability

Requirements traceability is concerned with documenting the life of a requirement.^[4] It should be possible to trace back to the origin of each requirement and every change made to the requirement should therefore be documented in order to achieve traceability.^[5] Even the use of the requirement after the implemented features have been deployed and used should be traceable.^[5]

Requirements come from different sources, like the business person ordering the product, the marketing manager and the actual user. These people all have different requirements for the product. Using requirements traceability, an implemented feature can be traced back to the person or group that wanted it during the requirements elicitation. This can, for example, be used during the development process to prioritize the requirement,^[6] determining how valuable the requirement is to a specific user. It can also be used after the deployment when user studies show that a feature is not used, to see why it was required in the first place.

Requirements activities

At each stage in a development process, there are key requirements management activities and methods. To illustrate, consider a standard five-phase development process with Investigation, Feasibility, Design, Construction and Test, and Release stages.

Investigation

In Investigation, the first three classes of requirements are gathered from the users, from the business and from the development team. In each area, similar questions are asked; what are the goals, what are the constraints, what are the current tools or processes in place, and so on. Only when these requirements are well understood can functional requirements be developed.

In the common case, requirements cannot be fully defined at the beginning of the project. Some requirements will change, either because they simply weren't extracted, or because internal or external forces at work affect the project in mid-cycle.

The deliverable from the Investigation stage is a requirements document that has been approved by all members of the team. Later, in the thick of development, this document will be critical in preventing scope creep or unnecessary changes. As the system develops, each new feature opens a world of new possibilities, so the requirements specification anchors the team to the original vision and permits a controlled discussion of scope change.

While many organizations still use only documents to manage requirements, others manage their requirements baselines using software tools. These tools allow requirements to be managed in a database, and usually have functions to automate traceability (e.g., by allowing electronic links to be created between parent and child requirements, or between test cases and requirements), electronic baseline creation, version control, and change management. Usually such tools contain an export function that allows a specification document to be created by exporting the requirements data into a standard document application.

Feasibility

In the Feasibility stage, costs of the requirements are determined. For user requirements, the current cost of work is compared to the future projected costs once the new system is in place. Questions such as these are asked: “What are data entry errors costing us now?” Or “What is the cost of scrap due to operator error with the current interface?” Actually, the need for the new tool is often recognized as these questions come to the attention of financial people in the organization.

Business costs would include, “What department has the budget for this?” “What is the expected rate of return on the new product in the marketplace?” “What’s the internal rate of return in reducing costs of training and support if we make a new, easier-to-use system?”

Technical costs are related to software development costs and hardware costs. “Do we have the right people to create the tool?” “Do we need new equipment to support expanded software roles?” This last question is an important type. The team must inquire into whether the newest automated tools will add sufficient processing power to shift some of the burden from the user to the system in order to save people time.

The question also points out a fundamental point about requirements management. A human and a tool form a system, and this realization is especially important if the tool is a computer or a new application on a computer. The human mind excels in parallel processing and interpretation of trends with insufficient data. The CPU excels in serial processing and accurate mathematical computation. The overarching goal of the requirements management effort for a software project would thus be to make sure the work being automated gets assigned to the proper processor. For instance, “Don’t make the human remember where she is in the interface. Make the interface report the human’s location in the system at all times.” Or “Don’t make the human enter the same data in two screens. Make the system store the data and fill in the second screen as needed.”

The deliverable from the Feasibility stage is the budget and schedule for the project.

Design

Assuming that costs are accurately determined and benefits to be gained are sufficiently large, the project can proceed to the Design stage. In Design, the main requirements management activity is comparing the results of the design against the requirements document to make sure that work is staying in scope.

Again, flexibility is paramount to success. Here’s a classic story of scope change in mid-stream that actually worked well. Ford auto designers in the early ‘80s were expecting gasoline prices to hit \$3.18 per gallon by the end of the decade. Midway through the design of the Ford Taurus, prices had centered to around \$1.50 a gallon. The design team decided they could build a larger, more comfortable, and more powerful car if the gas prices stayed low, so they redesigned the car. The Taurus launch set nationwide sales records when the new car came out, primarily because it was so roomy and comfortable to drive.

In most cases, however, departing from the original requirements to that degree does not work. So the requirements document becomes a critical tool that helps the team make decisions about design changes.^[7]

Construction and test

In the construction and testing stage, the main activity of requirements management is to make sure that work and cost stay within schedule and budget, and that the emerging tool does in fact meet requirements. A main tool used in this stage is prototype construction and iterative testing. For a software application, the user interface can

be created on paper and tested with potential users while the framework of the software is being built. Results of these tests are recorded in a user interface design guide and handed off to the design team when they are ready to develop the interface. This saves their time and makes their jobs much easier.

Verification: This effort verifies that the requirement has been implemented correctly. There are 4 methods of verification: analysis, inspection, testing, and demonstration. Numerical software execution results or throughput on a network test, for example, provides analytical evidence that the requirement has been met. Inspection of vendor documentation or spec sheets also verifies requirements. Actually testing or demonstrating the software in a lab environment also verifies the requirements: a test type of verification will occur when test equipment not normally part of the lab (or system under test) is used. Comprehensive test procedures which outline the steps and their expected results clearly identify what is to be seen as a result of performing the step. After the step or set of steps is completed the last step's expected result will call out what has been seen and then identify what requirement or requirements have been verified (identified by number). The requirement number, title and verbiage are tied together in another location in the test document.

Requirements change management

Hardly would any software development project be completed without some changes being asked of the project. The changes can stem from changes in the environment in which the finished product is envisaged to be used, business changes, regulation changes, errors in the original definition of requirements, limitations in technology, changes in the security environment and so on. The activities of requirements change management include receiving the change requests from the stakeholders, recording the received change requests, analyzing and determining the desirability and process of implementation, implementation of the change request, quality assurance for the implementation and closing the change request. Then the data of change requests be compiled, analyzed and appropriate metrics are derived and dovetailed into the organizational knowledge repository.^[8]

Release

Requirements management does not end with product release. From that point on, the data coming in about the application's acceptability is gathered and fed into the Investigation phase of the next generation or release. Thus the process begins again.

Tooling

Acquiring a tool to support requirements management is no trivial matter and it needs to be undertaken as part of a broader process improvement initiative. It has long been a perception that a tool, once acquired and installed on a project, can address all of its requirements management-related needs. However, the purchase or development of a tool to support requirements management can be a costly decision. Organizations may get burdened with expensive support contracts, disproportionate effort can get misdirected towards learning to use the tool and configuring it to address particular needs, and inappropriate use that can lead to erroneous decisions. Organizations should follow an incremental process to make decisions about tools to support their particular needs from within the wider context of their development process and tooling.^[9] The tools are presented in Requirements traceability.

See also

- Requirement

- [Requirements engineering](#)
- [Requirements analysis](#)
- [Requirements traceability](#)
- [Requirements Engineering Specialist Group](#)
- Process area (CMMI):
 - [Requirements Development \(RD\)](#)
 - [Requirements Management \(REQM\)](#)
- [Product requirements document](#)

References

1. Stellman, Andrew; Greene, Jennifer (2005). *Applied Software Project Management* (<https://web.archive.org/web/20150209011617/http://www.stellman-greene.com/aspm/>). O'Reilly Media. ISBN 978-0-596-00948-9. Archived from the original (<http://www.stellman-greene.com/aspm/>) on 2015-02-09.
2. "Requirements management" (http://www.ogc.gov.uk/delivery_lifecycle_requirements_management.asp). UK Office of Government Commerce. Retrieved 2009-11-10.
3. *A Guide to the Project Management Body of Knowledge* (<http://www.pmi.org/>) (4th ed.). Project Management Institute. 2008. ISBN 978-1-933890-51-7.
4. Gotel, O., Finkelstein, A. [An Analysis of the Requirements Traceability Problem](http://ollygotel.com/papers/an-analysis-of-the-requirements-traceability-problem/) (<http://ollygotel.com/papers/an-analysis-of-the-requirements-traceability-problem/>) Proc. of First International Conference on Requirements Engineering, 1994, pages 94-101
5. Gotel, Orlena; Cleland-Huang, Jane; Hayes, Jane Huffman; Zisman, Andrea; Egyed, Alexander; Grünbacher, Paul; Dekhtyar, Alex; Antoniol, Giuliano; Maletic, Jonathan (2012-01-01). Cleland-Huang, Jane; Gotel, Orlena; Zisman, Andrea (eds.). *Software and Systems Traceability* (<https://archive.org/details/software-systemst00clel>). Springer London. pp. 3 (<https://archive.org/details/software-systemst00clel/page/n21>)–22. doi:10.1007/978-1-4471-2239-5_1 (https://doi.org/10.1007%2F978-1-4471-2239-5_1). ISBN 9781447122388.
6. Rempel, Patrick; Mäder, Patrick (2015-03-23). Fricker, Samuel A.; Schneider, Kurt (eds.). *Requirements Engineering: Foundation for Software Quality*. Lecture Notes in Computer Science. Springer International Publishing. pp. 81–97. doi:10.1007/978-3-319-16101-3_6 (https://doi.org/10.1007%2F978-3-319-16101-3_6). ISBN 9783319161006.
7. Ralph, P., and Wand, Y. A Proposal for a Formal Definition of the Design Concept. In, Lyytinen, K., Loucopoulos, P., Mylopoulos, J., and Robinson, W., (eds.), *Design Requirements Engineering: A Ten-Year Perspective*: Springer-Verlag, 2009, pp. 103-136
8. Chemuturi, M. (2013). *Requirements Engineering and Management for Software Development Projects*. doi:10.1007/978-1-4614-5377-2 (<https://doi.org/10.1007%2F978-1-4614-5377-2>). ISBN 978-1-4614-5376-5.
9. Gotel, Orlena; Mäder, Patrick (2012-01-01). Cleland-Huang, Jane; Gotel, Orlena; Zisman, Andrea (eds.). *Software and Systems Traceability* (<https://archive.org/details/software-systemst00clel>). Springer London. pp. 43 (<https://archive.org/details/software-systemst00clel/page/n61>)–68. doi:10.1007/978-1-4471-2239-5_3 (https://doi.org/10.1007%2F978-1-4471-2239-5_3). ISBN 9781447122388.

Further reading

- CMMI Product Team (August 2006). "CMMI for Development, Version 1.2" (<http://www.sei.cmu.edu/library/abstracts/reports/06tr008.cfm>) (PDF). Technical Report CMU/SEI-2006-TR-008. Software Engineering Institute. Retrieved 2008-01-22.
- Colin Hood, Simon Wiedemann, Stefan Fichtinger, Urte Pautz *Requirements Management: Interface Between Requirements Development and All Other Engineering Processes* Springer, Berlin 2007, ISBN 3-540-47689-X
- Requirements Management - A Practice Guide (<https://www.pmi.org/pmbok-guide-standards/practice-guides/requirements-management>), PMI

External links

- U.K. Office of Government Commerce (OGC) - Requirements management (http://web.archive.nationalarchives.gov.uk/20110822131357/http://www.ogc.gov.uk/delivery_life_cycle_requirements_management.asp) (archive; OGC website ceased activity on 1 October 2011)
- CDC Unified Process Practices Guide - Requirements Management (http://www2.cdc.gov/cdcup/library/practices_guides/CDC_UP_Requirements_Management_Practices_Guide.pdf)
- International Requirements Engineering Board (IREB) (<https://www.ireb.org>)
- TopTeam Analyst - Requirements Management Tool (<http://www.technosolutions.com/>)
- RTM for Jira - Requirements and Test Management extension (<https://hexygen.com/requirements-test-management-jira/>)

Retrieved from "https://en.wikipedia.org/w/index.php?title=Requirements_management&oldid=973647810"

This page was last edited on 18 August 2020, at 11:54 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.