

Table of Contents

Requirements management

Agile Requirements

Cone of Uncertainty

Requirements analysis

Taxonomy of Requirements

Requirements traceability

Disciplines & Methodologies

Tools & Technologies

Links

3

5

6

7

7

8

9

9

10

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.

Requirements management is an essential component of the Application Lifecycle Management (ALM) process. It involves gathering requirements from stakeholders, documenting the requirements, tracking changes to the requirements, and ensuring that the requirements are met during software development and testing. The benefits of requirements management in ALM include ensuring that the software application is aligned with the business needs, reducing the risk of errors and delays, improving collaboration and communication among stakeholders, and improving the overall quality of the software application.

- documenting
- analyzing
- tracing
- prioritizing
- agreeing

Examples of requirements management:

- Collecting and documenting software requirements from stakeholders, such as users, customers, and subject matter experts.
- Analyzing and prioritizing requirements based on their importance, feasibility, and impact on the system.
- Tracing requirements to ensure that they are implemented correctly and meet the project goals.
- Managing changes to requirements throughout the development process, including adding new requirements, modifying existing ones, and removing unnecessary ones.
- Defining test cases and scenarios based on the requirements to ensure that the system meets the expected behavior.
- Creating and maintaining a requirements management plan that outlines the process, tools, and responsibilities for managing requirements.
- Tracking requirements using a dedicated requirements management tool, such as JIRA, Trello, or ReqView, to ensure that they are completed on time and within budget.
- Conducting regular reviews and evaluations of the requirements management process to identify areas for improvement.
- Communicating requirements and changes to all stakeholders, including developers, testers, project managers, and customers, to ensure that everyone is aligned with the project goals.
- Ensuring that the requirements are well-defined, clear, and testable, and that they meet regulatory and compliance standards.
- Non-functional requirement (NFR)
- Epic - feature - story - task

-
- TOGAF ADM Architecture Requirements Management
 - <https://www.pmi.org/learning/library/effective-requirements-management-project-success-8181>

What is requirements management?

Requirements management is the process of defining, documenting, analyzing, prioritizing, and tracking requirements throughout the software development lifecycle.

Why is requirements management important?

Requirements management is important because it helps to ensure that software projects are delivered on time, within budget, and with the desired quality. It also helps to reduce rework, improve collaboration between stakeholders, and increase the likelihood of meeting customer needs and expectations.

What are some common techniques used in requirements management?

Common techniques used in requirements management include stakeholder interviews, surveys, use cases, user stories, prototyping, and traceability matrices.

What are some common challenges associated with requirements management?

Some common challenges associated with requirements management include changing requirements, conflicting stakeholder priorities, lack of clarity or detail in requirements, and difficulty in prioritizing requirements.

How can organizations improve their requirements management process?

Organizations can improve their requirements management process by establishing clear roles and responsibilities for stakeholders, using tools and techniques to capture and manage requirements, establishing clear and measurable requirements, and ensuring communication and collaboration between stakeholders.

What is the role of project managers in requirements management?

Project managers play a critical role in requirements management by ensuring that requirements are clearly defined, prioritized, and tracked throughout the project lifecycle. They are also responsible for managing stakeholders and ensuring that their needs and expectations are met.

How can agile methodologies impact requirements management?

Agile methodologies can impact requirements management by emphasizing continuous delivery, frequent feedback, and collaboration between stakeholders. This can lead to more iterative and flexible requirements management processes that allow for more rapid changes and adjustments.

What is the importance of traceability in requirements management?

Traceability is important in requirements management because it allows for the tracking of requirements from their origin through to their implementation and testing. This helps to ensure that requirements are properly prioritized and addressed throughout the project lifecycle.

How can requirements management impact the overall success of a software project?

Requirements management can impact the overall success of a software project by ensuring that the project meets the needs and expectations of stakeholders, is delivered on time and within budget, and meets the desired quality standards.

What are some tools and techniques used in requirements management?

Tools and techniques used in requirements management include requirement management software, use cases, user stories, personas, flowcharts, traceability matrices, and prioritization matrices.

Snippet from [Wikipedia: Requirements management](#)

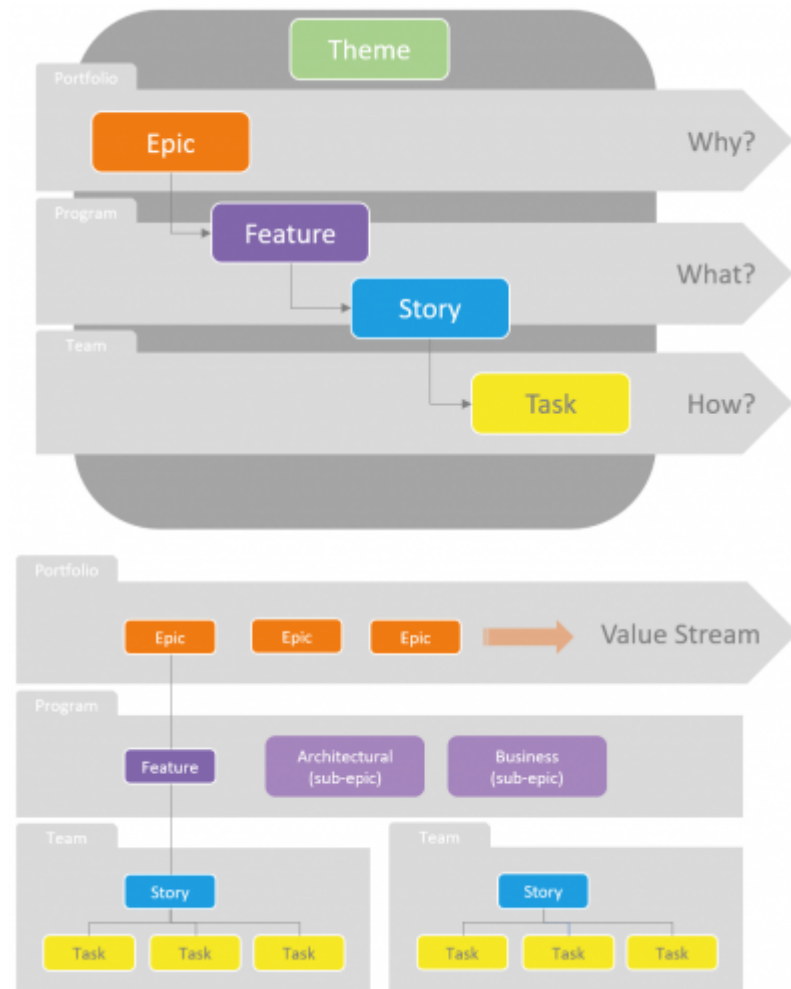
Requirements management is the process of documenting, analyzing, tracing, prioritizing and agreeing on requirements and then controlling change and communicating to relevant stakeholders. It is a continuous process throughout a project. A requirement is a capability to which a project outcome (product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

[Creative Commons Attribution-Share Alike 4.0](#)

- [DIKW pyramid](#)

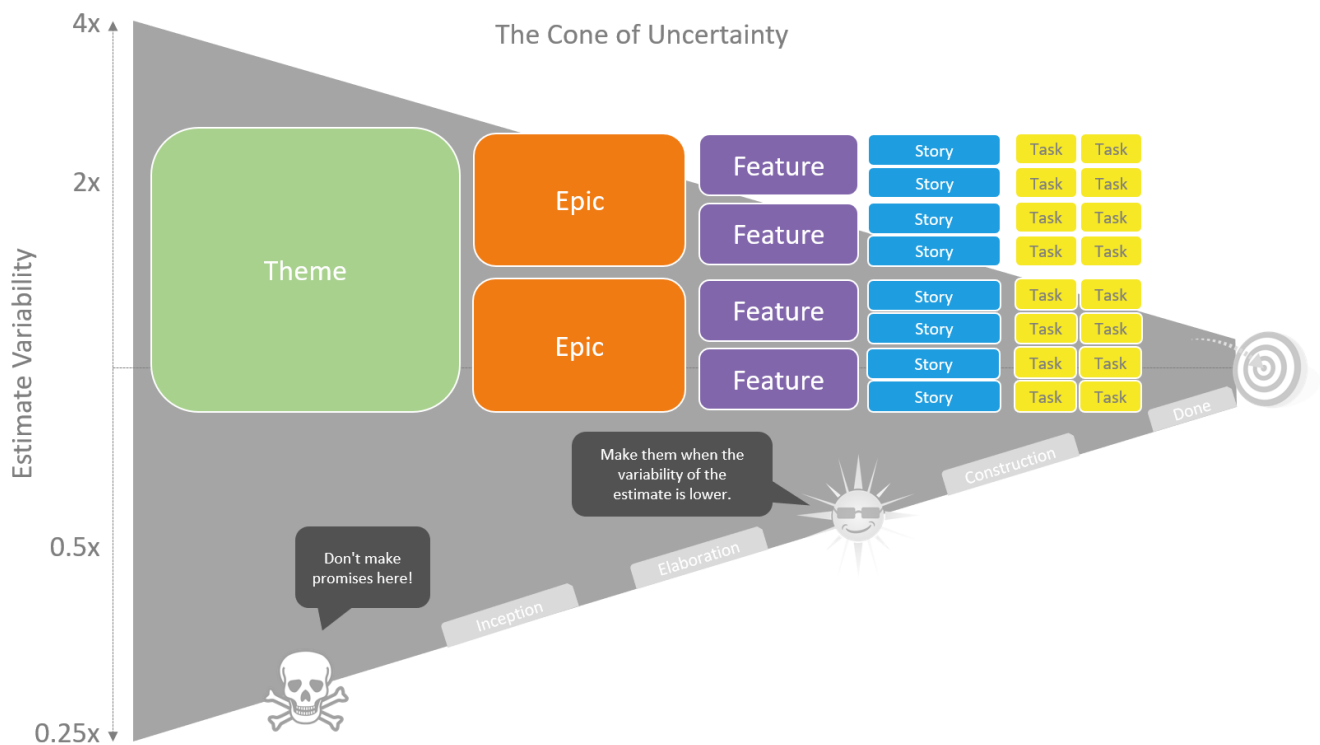
Agile Requirements

- [SAFe Requirements Model](#)
- [Agile Requirements Modeling](#)



Cone of Uncertainty

- [Cone of Uncertainty](#)



- http://www.agilenutshell.com/cone_of_uncertainty
- Dunning-Kruger Effect

Requirements analysis

Snippet from [Wikipedia: Requirements analysis](#)

In systems engineering and software engineering, **requirements analysis** focuses on the tasks that determine the needs or conditions to meet the new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, *analyzing, documenting, validating and managing* software or system requirements.

Requirements analysis is critical to the success or failure of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

[Creative Commons Attribution-Share Alike 4.0](#)

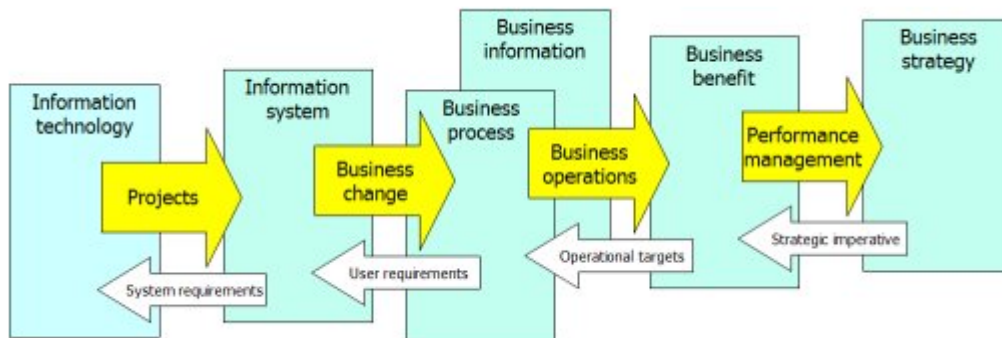
Taxonomy of Requirements

- [Taxonomy of Requirements](#)

Functional vs non-functional requirements

- <https://www.altexsoft.com/blog/business/functional-and-non-functional-requirements-specification-and-types>

Information Management Body of Knowledge (IMBOK)



The IMBOK process areas are:

- **Projects:** Adding new capacity, software, and hardware to information systems
- **Business Change:** Evaluating information to drive improvements in processes
- **Business Operations:** The day-to-day of a business. These will guide improvements based on updates to processes, and will hopefully increase benefits.
- **Performance Management:** Trying to ensure operations are running at peak capacity

[IMBOK.org](https://www.almbok.org)

Requirements traceability

Snippet from [Wikipedia: Requirements traceability](#)

Requirements traceability is a sub-discipline of requirements management within software development and systems engineering. Traceability as a general term is defined by the IEEE Systems and Software Engineering Vocabulary as (1) the degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor-successor or primary-subordinate relationship to one another; (2) the identification and documentation of derivation paths (upward) and allocation or flowdown paths (downward) of work products in the work product hierarchy; (3) the degree to which each element in a software development product establishes its reason for existing; and (4) discernible association among two or more logical entities, such as requirements, system elements, verifications, or tasks.

Requirements traceability in particular, is defined as "the ability to describe and follow the life of a requirement in both a forwards and backwards direction (i.e., from its origins, through its development and specification, to its subsequent deployment and use, and through periods of ongoing refinement and iteration in any of these phases)". In the requirements engineering field, traceability is about understanding how high-level requirements – objectives, goals, aims, aspirations, expectations, business needs – are

transformed into development ready, low-level requirements. It is therefore primarily concerned with satisfying relationships between layers of information (aka artifacts). However, traceability may document relationships between many kinds of development artifacts, such as requirements, specification statements, designs, tests, models and developed components. For example, it is common practice to capture verification relationships to demonstrate that a requirement is verified by a certain test artifact.

Traceability is especially relevant when developing safety-critical systems and therefore prescribed by safety guidelines, such as DO178C, ISO 26262, and IEC61508. A common requirement of these guidelines is that critical requirements must be verified and that this verification must be demonstrated through traceability.

[Creative Commons Attribution-Share Alike 4.0](#)

Disciplines & Methodologies

- [Business Process Management \(BPM\)](#)
- [Customer relationship management](#)
- [Enterprise Search](#)
- [Functional specification](#)
- [Fusion development](#)
- [GitOps](#)
- [Information Management](#)
- [Information security](#)
- [Knowledge Management](#)
- [MoSCoW](#)
- [Presentation Techniques](#)
- [Product breakdown structure \(PBS\)](#)
- [Pugh Matrix](#)
- [Requirements analysis](#)
- [Software design](#)
- [Software development process](#)
- [Software documentation](#)
- [Static program analysis](#)
- [User experience design](#)
- [Visual Collaboration](#)
- [Work breakdown structure \(WBS\)](#)

Tools & Technologies

- [Appian](#)
- [AppSheet](#)
- [Axure](#)
- [Azure DevOps](#)
- [Confluence](#)
- [Figma](#)
- [GitLab](#)
- [HP ALM](#)
- [iRise](#)

- [Jama](#)
- [Mendix](#)
- [Microfocus](#)
- [Microsoft Feedback Client](#)
- [Microsoft Office](#)
- [Miro](#)
- [Orcanos](#)
- [Rally Software](#)
- [Rational solution for CLM](#)
- [Rational Team Concert](#)
- [ReQtest](#)
- [Salesforce Platform](#)
- [SAP Solution Manager](#)
- [SharePoint](#)
- [Team Foundation Server](#)
- [TeamForge](#)
- [Helix ALM \(TestTrack\)](#)
- [Visual Studio](#)
- [VSALM](#)
- [Visual Studio Team Services](#)
- [Webflow](#)

Links

- [12 Best Software Development Methodologies with Pros and Con](#)
- [Code.org](#)
- [Continuous Improvement Toolkit](#)
- [Developer Roadmaps \(roadmap.sh\)](#)
- [Developerexperience.io](#)
- [Devicon](#)
- [Envatotuts+](#)
- [Knowledge Management Tools](#)
- [Modern Requirements](#)
- [Modern Work Blog](#)
- [Total number of Websites](#)
- [unFIX](#)
- [Vectorly](#)
- [w3Schools](#)
- [Wicked Problem Solving](#)
- [API](#)
- [... as a service](#)
- [CIO Wiki](#)
- [Dunning-Kruger Effect](#)
- [EABOK](#)
- [Enterprise Value](#)
- [GDPR](#)
- [IEEE software life cycle](#)
- [ISO/IEC 24744](#)
- [Low-code development](#)

- PMBOK
- PMI
- RSS
- SEBoK
- Shadow IT
- ISO/IEC 15504
- SWEBOK
- UML
- User experience (UX)
- V-Model

External links:

- <https://www.pmi.org/learning/library/effective-requirements-management-project-success-8181>
- <https://www.projectmanagement.com/blogs/285716/Requirements>
- <https://www.pmi.org/pmbok-guide-standards/practice-guides/requirements-management>
- <https://www.requirementsmanagementtools.com/>

[governance](#), [skill](#), [dx](#), [devx](#)

ToDo

-  - [Support Us...](#) →

- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Verification and Validation
- Requirements Traceability
- Requirements Prioritization
- Requirements Change Management
- Requirements Documentation
- Requirements Repository
- Requirements Metrics and Reporting
- Requirements Engineering
- Requirements Management Tools
- Requirements Collaboration
- User Stories
- Use Cases
- Business Process Modeling
- Stakeholder Management
- Requirements Negotiation
- Agile Requirements Management
- Requirements Management Process
- Requirements Management Plan
- Requirements Management Best Practices

From:

<https://www.almbok.com/> - **ALMBoK.com**

Permanent link:

<https://www.almbok.com/requirements/requirements>

Last update: **2023/05/01 18:13**

