

Requirements Engineering Overview

Requirements engineering (RE) is a critical aspect of software engineering that involves the identification, documentation, and maintenance of the requirements of a system. It ensures that the final product meets the needs and expectations of stakeholders.

Problem Recognition

Problem recognition involves understanding the problem that needs to be solved. This includes:

- **Stakeholder identification:** Recognizing all the stakeholders who will be affected by the system.
- **Context understanding:** Grasping the environment and constraints in which the system will operate.
- **Goal definition:** Determining the objectives and desired outcomes of the system.

Requirements Engineering Tasks

The main tasks in requirements engineering include:

1. **Elicitation:** Gathering requirements from stakeholders through interviews, surveys, workshops, etc.
2. **Analysis:** Examining and prioritizing requirements to resolve conflicts and ambiguities.
3. **Specification:** Documenting the requirements in a clear and detailed manner.
4. **Validation:** Ensuring the requirements accurately reflect the stakeholder needs.
5. **Management:** Handling changes to the requirements throughout the project lifecycle.

Processes in Requirements Engineering

The requirements engineering process typically involves the following steps:

1. **Feasibility Study:** Assessing whether the proposed system is viable and worth pursuing.
2. **Elicitation and Analysis:** Collecting and analyzing the requirements.
3. **Specification:** Writing detailed requirements documentation.
4. **Validation:** Checking the requirements for consistency, completeness, and feasibility.
5. **Requirements Management:** Managing changes and ensuring traceability of requirements.

Requirements Specification

Requirements specification is the process of documenting the requirements in a detailed and precise manner.

This document is often referred to as the Software Requirements Specification (SRS) and includes:

- **Functional requirements:** Descriptions of the system's functions.
- **Non-functional requirements:** Performance, security, usability, and other quality attributes.
- **Constraints:** Limitations imposed by the environment or technology.

Use Cases and Functional Specification

Use cases are a way to capture functional requirements by describing interactions between users (actors) and the system to achieve specific goals. They include:

- **Actors:** Entities interacting with the system.
- **Scenarios:** Detailed sequences of steps for each interaction.
- **Preconditions and Postconditions:** Conditions that must be true before and after the scenario runs.

Functional specifications detail the functionality that the system must provide, often derived from the use cases.

Requirements Validation

Requirements validation ensures that the documented requirements:

- Reflect the real needs of stakeholders.
- Are complete and consistent.
- Are feasible within the given constraints.

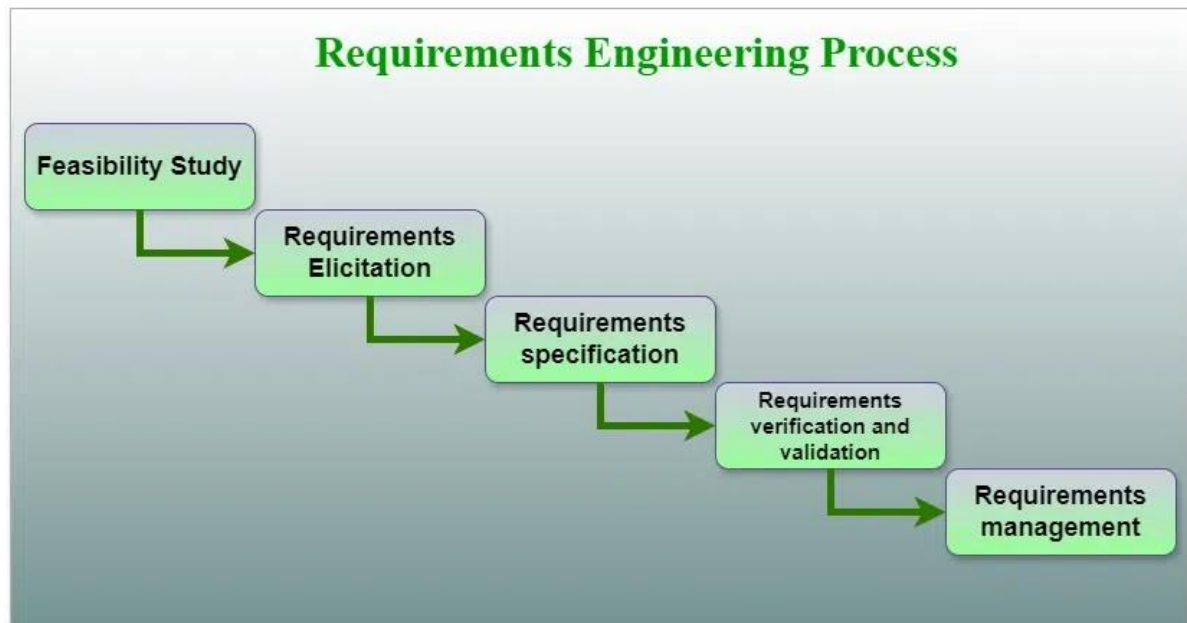
Techniques for validation include reviews, inspections, prototyping, and testing.

Requirements Analysis

Requirements analysis involves examining the gathered requirements to:

- Detect and resolve conflicts.
- Prioritize requirements based on stakeholder needs.
- Model the requirements using tools like data flow diagrams or entity-relationship diagrams.

Requirements Engineering Process



Requirements Engineering Process

1. **Feasibility Study**
2. **Requirements elicitation**
3. **Requirements specification**
4. **Requirements for verification and validation**
5. **Requirements management**

1. Feasibility Study

The feasibility study mainly concentrates on below five mentioned areas below. Among these Economic Feasibility Study is the most important part of the feasibility analysis and the Legal Feasibility Study is less considered feasibility analysis.

1. **Technical Feasibility:** In Technical Feasibility current resources both hardware software along required technology are analyzed/assessed to develop the project. This technical feasibility study reports whether there are correct required resources and technologies that will be used for project development. Along with this, the feasibility study also analyzes the technical skills and capabilities of the technical team, whether existing technology can be used or not, whether maintenance and up-gradation are easy or not for the chosen technology, etc.
2. **Operational Feasibility:** In Operational Feasibility degree of providing service to requirements is analyzed along with how easy the product will be to operate and maintain after deployment. Along with this other operational scopes are determining the usability of the product, Determining suggested solution by the software development team is acceptable or not, etc.
3. **Economic Feasibility:** In the Economic Feasibility study cost and benefit of the project are analyzed. This means under this feasibility study a detailed analysis is carried out will be cost of the project for development which includes all required costs for final development hardware and software resources required, design and development costs operational costs, and so on. After that, it is analyzed whether the project will be beneficial in terms of finance for the organization or not.

2. Requirements Elicitation

It is related to the various ways used to gain knowledge about the project domain and requirements. The various sources of domain knowledge include customers, business manuals, the existing software of the same type, standards, and other stakeholders of the project. The techniques used for requirements elicitation include interviews, brainstorming, task analysis, Delphi technique, prototyping, etc. Some of these are discussed [here](#). Elicitation does not produce formal models of the requirements understood. Instead, it widens the domain knowledge of the analyst and thus helps in providing input to the next stage.

Requirements elicitation is the process of gathering information about the needs and expectations of stakeholders for a software system. This is the first step in the requirements engineering process and it is critical to the success of the software development project. The goal of this step is to understand the problem that the software system is intended to solve and the needs and expectations of the stakeholders who will use the system. Several techniques can be used to elicit requirements, including:

- **Interviews:** These are one-on-one conversations with stakeholders to gather information about their needs and expectations.

- **Surveys:** These are questionnaires that are distributed to stakeholders to gather information about their needs and expectations.
- **Focus Groups:** These are small groups of stakeholders who are brought together to discuss their needs and expectations for the software system.
- **Observation:** This technique involves observing the stakeholders in their work environment to gather information about their needs and expectations.
- **Prototyping:** This technique involves creating a working model of the software system, which can be used to gather feedback from stakeholders and to validate requirements.

It's important to document, organize, and prioritize the requirements obtained from all these techniques to ensure that they are complete, consistent, and accurate.

3. Requirements Specification

This activity is used to produce formal software requirement models. All the requirements including the functional as well as the non-functional requirements and the constraints are specified by these models in totality. During specification, more knowledge about the problem may be required which can again trigger the elicitation process. The models used at this stage include ER diagrams, data flow diagrams(DFDs), function decomposition diagrams(FDDs), data dictionaries, etc.

Requirements specification is the process of documenting the requirements identified in the analysis step in a clear, consistent, and unambiguous manner. This step also involves prioritizing and grouping the requirements into manageable chunks.

The goal of this step is to create a clear and comprehensive document that describes the requirements for the software system. This document should be understandable by both the development team and the stakeholders.

Several types of requirements are commonly specified in this step, including

1. **Functional Requirements:** These describe what the software system should do. They specify the functionality that the system must provide, such as input validation, data storage, and user interface.
2. **Non-Functional Requirements:** These describe how well the software system should do it. They specify the quality attributes of the system, such as performance, reliability, usability, and security.
3. **Constraints:** These describe any limitations or restrictions that must be considered when developing the software system.
4. **Acceptance Criteria:** These describe the conditions that must be met for the software system to be considered complete and ready for release.

To make the requirements specification clear, the requirements should be written in a natural language and use simple terms, avoiding technical jargon, and using a consistent format throughout the document. It is also important to use diagrams, models, and other visual aids to help communicate the requirements effectively.

Once the requirements are specified, they must be reviewed and validated by the stakeholders and development team to ensure that they are complete, consistent, and accurate.

4. Requirements Verification and Validation

Verification: It refers to the set of tasks that ensures that the software correctly implements a specific function.

Validation: It refers to a different set of tasks that ensures that the software that has been built is traceable to customer requirements. If requirements are not validated, errors in the requirement definitions would propagate to the successive stages resulting in a lot of modification and rework. The main steps for this process include:

1. The requirements should be consistent with all the other requirements i.e. no two requirements should conflict with each other.
2. The requirements should be complete in every sense.
3. The requirements should be practically achievable.

Reviews, buddy checks, making test cases, etc. are some of the methods used for this.

Requirements verification and validation (V&V) is the process of checking that the requirements for a software system are complete, consistent, and accurate and that they meet the needs and expectations of the stakeholders.

The goal of V&V is to ensure that the software system being developed meets the requirements and that it is developed on time, within budget, and to the required quality.

1. Verification is checking that the requirements are complete, consistent, and accurate. It involves reviewing the requirements to ensure that they are clear, testable, and free of errors and inconsistencies. This can include reviewing the requirements document, models, and diagrams, and holding meetings and walkthroughs with stakeholders.
2. Validation is the process of checking that the requirements meet the needs and expectations of the stakeholders. It involves testing the requirements to ensure that they are valid and that the software system being developed will meet the needs of the stakeholders. This can include testing the software system through simulation, testing with prototypes, and testing with the final version of the software.
3. Verification and Validation is an iterative process that occurs throughout the software development life cycle. It is important to involve stakeholders and the development team in the V&V process to ensure that the requirements are thoroughly reviewed and tested.

It's important to note that V&V is not a one-time process, but it should be integrated and continue throughout the software development process and even in the maintenance stage.

5. Requirements Management

Requirement management is the process of analyzing, documenting, tracking, prioritizing, and agreeing on the requirement and controlling the communication with relevant stakeholders. This stage takes care of the changing nature of requirements. It should be ensured that the SRS is as modifiable as possible to incorporate changes in requirements specified by the end users at later stages too. Modifying the software as per requirements in a systematic and controlled manner is an extremely important part of the requirements engineering process. Requirements management is the process of managing the requirements throughout the software development life cycle, including tracking and controlling changes, and ensuring that the requirements are still valid and relevant. The goal of requirements management is to ensure that the software system being developed meets the needs and expectations of the stakeholders and that it is developed on time, within budget, and to the required quality.

Several key activities are involved in requirements management, including:

1. **Tracking and controlling changes:** This involves monitoring and controlling changes to the requirements throughout the development process, including identifying the source of the change, assessing the impact of the change, and approving or rejecting the change.
2. **Version control:** This involves keeping track of different versions of the requirements document and other related artifacts.
3. **Traceability:** This involves linking the requirements to other elements of the development process, such as design, testing, and validation.
4. **Communication:** This involves ensuring that the requirements are communicated effectively to all stakeholders and that any changes or issues are addressed promptly.
5. **Monitoring and reporting:** This involves monitoring the progress of the development process and reporting on the status of the requirements.

Requirements management is a critical step in the software development life cycle as it helps to ensure that the software system being developed meets the needs and expectations of stakeholders and that it is developed on time, within budget, and to the required quality. It also helps to prevent scope creep and to ensure that the requirements are aligned with the project goals.

Tools Involved in Requirement Engineering

- Observation report
- Questionnaire (survey, poll)
- Use cases
- User stories
- Requirement workshop
- Mind mapping
- Roleplaying
- Prototyping