

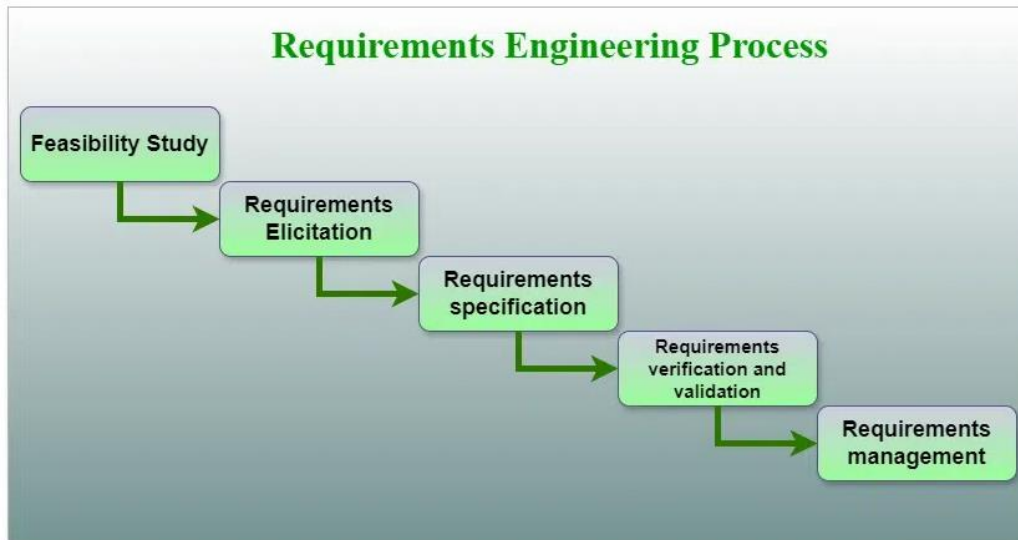
Unit 3: Software Requirements Engineering

• Requirement Engineering

What is Requirements Engineering?

A systematic and strict approach to the definition, creation, and verification of requirements for a software system is known as requirements engineering. To guarantee the effective creation of a software product, the requirements engineering process entails several tasks that help in understanding, recording, and managing the demands of stakeholders.

Requirements Engineering Process



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

Tools Involved in Requirement Engineering

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

Advantages of Requirements Engineering Process

- Helps ensure that the software being developed meets the needs and expectations of the stakeholders
- Can help identify potential issues or problems early in the development process, allowing for adjustments to be made before significant
- Helps ensure that the software is developed in a cost-effective and efficient manner
- Can improve communication and collaboration between the development team and stakeholders
- Helps to ensure that the software system meets the needs of all stakeholders.

Unit 3: Software Requirements Engineering

- Provides an unambiguous description of the requirements, which helps to reduce misunderstandings and errors.
- Helps to identify potential conflicts and contradictions in the requirements, which can be resolved before the software development process begins.
- Helps to ensure that the software system is delivered on time, within budget, and to the required quality standards.
- Provides a solid foundation for the development process, which helps to reduce the risk of failure.

Disadvantages of Requirements Engineering Process

- Can be time-consuming and costly, particularly if the requirements-gathering process is not well-managed
- Can be difficult to ensure that all stakeholders' needs and expectations are taken into account
- It Can be challenging to ensure that the requirements are clear, consistent, and complete
- Changes in requirements can lead to delays and increased costs in the development process.
- As a best practice, Requirements engineering should be flexible, adaptable, and should be aligned with the overall project goals.
- It can be time-consuming and expensive, especially if the requirements are complex.
- It can be difficult to elicit requirements from stakeholders who have different needs and priorities.
- Requirements may change over time, which can result in delays and additional costs.
- There may be conflicts between stakeholders, which can be difficult to resolve.
- It may be challenging to ensure that all stakeholders understand and agree on the requirements.

Stages in Software Engineering Process

Requirements engineering is a critical process in software engineering that involves identifying, analyzing, documenting, and managing the requirements of a software system.

- **Elicitation:** In this stage, the requirements are gathered from various stakeholders such as customers, users, and domain experts. The aim is to identify the features and functionalities that the software system should provide.
- **Analysis:** In this stage, the requirements are analyzed to determine their feasibility, consistency, and completeness. The aim is to identify any conflicts or contradictions in the requirements and resolve them.
- **Specification:** In this stage, the requirements are documented in a clear, concise, and unambiguous manner. The aim is to provide a detailed description of the requirements that can be understood by all stakeholders.
- **Validation:** In this stage, the requirements are reviewed and validated to ensure that they meet the needs of all stakeholders. The aim is to ensure that the requirements are accurate, complete, and consistent.
- **Management:** In this stage, the requirements are managed throughout the software development lifecycle. The aim is to ensure that any changes or updates to the requirements are properly documented and communicated to all stakeholders.
- Effective requirements engineering is crucial to the success of software development projects. It helps ensure that the software system meets the needs of all stakeholders and is delivered on time, within budget, and to the required quality standards.

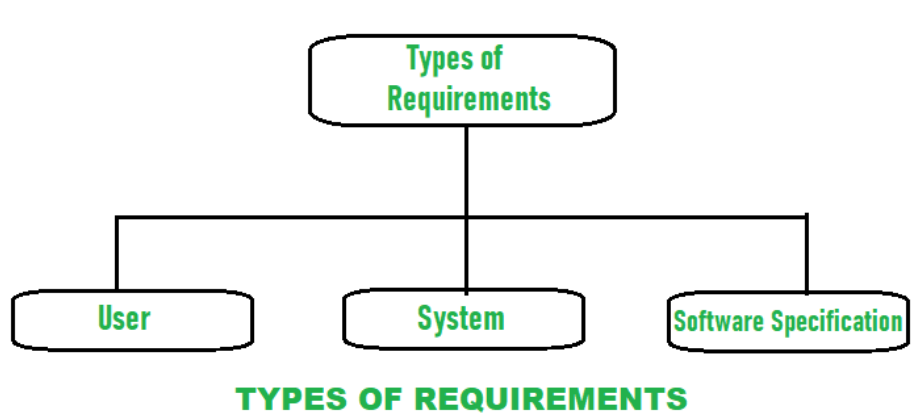
• Types of Requirements

Requirement simply means a thing that is needed or wanted. Requirement engineering is process of defining requirement, establishing, documenting it in a

Unit 3: Software Requirements Engineering

proper manner, and to keep quality of requirements of a customer from a system, and the limitations under which it operates and is developed.

Requirements are something that is needed to satisfy whether by designing, product, or process of a software system.



1. **User requirements:** User requirement simply means needs of users that should be fulfilled by software system. It is documented in a User Requirement Document (URD). User requirement is good if it is clear and short, results in increasing overall quality, increases productivity, is traceable, etc.
2. **System Requirements:** System requirement simply means needs of system to run smoothly and efficiently. It is a structured document that gives a detailed description of system functions, services, and operational constraints. It requires many hardware and software resources. If these hardware and software resources are not or less available, then it may result in system failure or causes problems during performance. Between client and contractor, it is written as a contract to define all requirements that are needed to be implemented to increase productivity.
3. **Software specification:** It is a detailed description of software system requirements with the help of which designing and implementation can be done to develop software. For software developers, software specification is usually written that makes it easier for the developer to understand overall requirement of software.

Two main types of requirements of a system:

1. **Functional requirements:** Functional requirements are mandatory which means it is compulsory and needed to be fulfilled. They generally describe and define features of end product of software system and simply focus on what the end product does. These are the requirements that a system should accomplish or do like calculations, data manipulations, etc. Functional requirement of users is high-level abstract statements. It generally describes what system should do whenever required but system functions should be described in detail by functional system requirements.

Unit 3: Software Requirements Engineering

2. **Non-Functional Requirements** : Non-functional requirements are not mandatory which means that they are not compulsory to be fulfilled. The non-functional requirements define system properties and system performance.

It simply focuses on how the end product works and it is not very easy and hard to find out non-functional requirements and captured as a quality attribute.

Non-functional requirements are more disapproved and if the non-functional requirements are not fulfilled then complete system is of no use.

Example - Product requirement, organizational requirement.

• Feasibility Study

A feasibility study is the first item on a project manager's agenda before approving the start of a project. Although they are not needed to carry out a feasibility study themselves, project managers utilize it as a roadmap to steer the project and gain a complete grasp of its characteristics, business objectives, and risk considerations.

What is a Feasibility Study in Project Management?

A feasibility study is a thorough analysis of a project proposal that considers all elements essential to success in order to determine the project's chances of success.

A feasibility analysis, as its name suggests, is used to assess an idea's viability, such as confirming that a project is both legally and technically possible and also economically reasonable. There are several potential causes for this, including the need for excessive resources, which not only hinders them from executing other jobs but can also cause more expense than a business would recover by taking on an unsuccessful project.

Types of Feasibility Studies

Feasibility analysis assesses the project's likelihood of success, perceived impartiality is a crucial element in the legitimacy of the assessment for interested investors and financing institutions.

1. **Technical Feasibility**

The technological resources that the organization has access to are the main focus of this examination. It aids companies in determining if the technological resources are enough and if the tech team has the skills necessary to turn concepts into functional systems.

2. **Financial Feasibility**

Before allocating financial resources, this evaluation often includes a cost or benefit assessment of a project to help businesses examine the feasibility, costs, and advantages related to a project.

Unit 3: Software Requirements Engineering

3. Legal Feasibility

This analysis looks at any potential conflicts between the planned project and applicable laws, such as zoning regulations, data protection laws, and social media usage regulations.

A feasibility study can find that the preferred site of the organization is not permitted for that particular kind of enterprise.

4. Operational Feasibility

This evaluation entails doing research to determine whether and how effectively the needs of the organization can be satisfied by the project's completion.

Additionally, an operational feasibility study evaluates how the planning of a project fulfills the needs found during the system development process' requirements analysis phase.

5. Scheduling Feasibility

The most crucial evaluation for a project's success is its scheduling feasibility. After taking into account the organization's capabilities, it predicts the amount of time required to execute the project and assesses if that timeframe is accessible.

Importance of Feasibility Study

A feasibility study is valuable because it allows a company to be spot?on before devoting time, resources, or money. A feasibility study could find innovative suggestions that change the project's scope.

- Increases the focus of project teams
- Explores new opportunities
- Offers essential information that aids in deciding whether to move forward or not
- Decreases the number of available business possibilities
- Finds a strong justification for continuing the project
- Raises the success rate by looking at several factors
- Aids in making project decisions
- Identifies the reasons why proceeding is not a good idea

Aim of Feasibility Study

- The overall objective of the organization are covered and contributed by the system or not.
- The implementation of the system be done using current technology or not.
- Can the system be integrated with the other system which are already exist

Feasibility Study Process

The below steps are carried out during entire feasibility analysis.

Unit 3: Software Requirements Engineering

1. **Information assessment:** It assesses the original project concept and establishes the main aims and objectives.
2. **Information collection:** It collects the necessary information and data required to evaluate the project's many components.
3. **Report writing:** It produces an in-depth feasibility report that details the analysis and results.
4. **General information:** It gives a summary of the main points discussed in the report on the feasibility study.

- **Requirement Documentation**

Requirement documentation in software engineering is a formal record that outlines what a software system must do, serving as a blueprint for development, testing, and maintenance. It captures stakeholder needs, system functionality, and constraints to ensure clarity and alignment across the project team. Below is a detailed explanation of its purpose, components, types, best practices, and tools, tailored for clarity and practicality.

Purpose of Requirement Documentation

- **Clarity and Agreement:** Ensures all stakeholders (clients, developers, testers) have a shared understanding of the system's purpose and functionality.
- **Guidance for Development:** Provides a reference for developers to build the system and for testers to verify it meets expectations.
- **Scope Management:** Defines project boundaries to prevent scope creep.
- **Traceability:** Links requirements to design, code, and test cases for accountability.
- **Risk Reduction:** Minimizes miscommunication, rework, and project delays.

Tools for Requirement Documentation

1. Documentation Tools:

- Microsoft Word/Google Docs: For traditional SRS documents.
- Confluence: Collaborative platform for team documentation.

2. Requirement Management Tools:

- Jira: Tracks user stories and backlogs in Agile projects.
- IBM DOORS: For complex, large-scale projects.
- Trello: For lightweight requirement tracking.

3. Diagramming Tools: Lucidchart, Visio, or Draw.io: For use case diagrams and flowcharts.

4. Prototyping Tools:

Figma, Balsamiq: For creating wireframes to visualize requirements.

Unit 3: Software Requirements Engineering

Who Uses the Requirements Document and Why?

Requirement documentation in software engineering guides various stakeholders throughout the software development lifecycle.

- **Development Project Leader:** Utilizes the requirement documentation to establish project direction, allocate resources effectively, and ensure alignment with organizational objectives. This ensures smooth project execution and timely delivery.
- **Requirements Analyst:** Analyzes and interprets the documented requirements to identify potential gaps, inconsistencies, or conflicts. This critical role ensures the requirements are comprehensive, clear, and feasible.
- **Development Team Member:** Relies on the requirement documentation to understand the scope of work, prioritize tasks, and develop software solutions that meet specified requirements. This promotes collaboration and streamlines the development process.
- **QA Specialist:** Uses the documented requirements to design test cases, verify software functionality, and ensure the final product meets quality standards. This helps in detecting and resolving defects early in the development cycle.
- **User Documentation Specialist:** Leverages the requirement documentation to create user manuals, tutorials, and other documentation that assists end-users in understanding and utilizing the software effectively. This enhances user experience and reduces support inquiries.
- **Legacy Support Specialist:** References the requirement documentation to understand the original design and functionality of the software, aiding in troubleshooting and maintaining legacy systems.
- **Maintenance Team Member:** Relies on the documented requirements to implement changes, updates, and enhancements to the software over its lifecycle. This ensures the software remains relevant and effective in meeting evolving user needs.
- **Development Manager:** Oversees the entire development process, ensuring that the documented requirements are met within budget and schedule constraints. This role is crucial in driving project success and achieving organizational objectives.
- **Project Sponsor:** Reviews the requirement documentation to ensure the project aligns with business goals, regulatory requirements, and stakeholder expectations. This helps in securing funding and support for the project.

Unit 3: Software Requirements Engineering

- **Business Expert:** Provides domain-specific knowledge and insights during the requirement elicitation process, ensuring the documented requirements accurately reflect business needs and priorities. This alignment between technical and business requirements is essential for delivering value to the organization.

- **Requirement Validation**

What is Requirements Validation?

Requirements validation is the process of checking that requirements defined for development, define the system that the customer wants. To check issues related to requirements, we perform requirements validation. We typically use requirements validation to check errors at the initial phase of development as the error may increase excessive rework when detected later in the development process.

During the requirement validation process. Different types of checks should be carried out according to the requirement in the requirement document:-

1. **Validity check:-** A user may think that a system is needed to perform certain functions. However, further thorough analysis may identify additional or different functions that are required.
2. **Consistency check:-** requirements in the document should not conflict. That is, there should not be contradictory constraints or different descriptions of the same system function.
3. **Completeness check:-** these requirements documents should include requirements that define all functions and the constraints intended by the system user.
4. **Realism check:-** Using knowledge of existing technology, the requirements should be checked to ensure that they can be implemented. These checks should also take account of the budget and schedule for the system development.
5. **Verifiability:-** to reduce the potential for dispute between customer and contractor, system requirements should always be written so that they are verifiable. This means that you should be able to write a set of tests that can demonstrate that the delivered system meets the specified requirement.

Advantages of Requirements Validation Techniques

1. **Improved quality of the final product:** By identifying and addressing requirements early on in the development process, using validation techniques can improve the overall quality of the final product.
2. **Reduced development time and cost:** By identifying and addressing requirements early on in the development process, using validation techniques can reduce the likelihood of costly rework later on.

Unit 3: Software Requirements Engineering

3. **Increased user involvement:** Involving users in the validation process can lead to increased user buy-in and engagement in the project.
4. **Improved communication:** Using validation techniques can improve communication between stakeholders and developers, by providing a clear and visual representation of the software requirements.
5. **Easy testing and validation:** A prototype can be easily tested and validated, allowing stakeholders to see how the final product will work and identify any issues early on in the development process.
6. **Increased alignment with business goals:** Using validation techniques can help to ensure that the requirements align with the overall business goals and objectives of the organization.
7. **Traceability:** This technique can help to ensure that the requirements are being met and that any changes are tracked and managed.
8. **Agile methodologies:** Agile methodologies provide an iterative approach to validate requirements by delivering small chunks of functionality and getting feedback from the customer.

Disadvantages of Requirements Validation Techniques

1. **Increased time and cost:** Using validation techniques can be time-consuming and costly, especially when involving multiple stakeholders.
2. **Risk of conflicting requirements:** Using validation techniques can lead to conflicting requirements, which can make it difficult to prioritize and implement the requirements.
3. **Risk of changing requirements:** Requirements may change over time and it can be difficult to keep up with the changes and ensure that the project is aligned with the updated requirements.
4. **Misinterpretation and miscommunication:** Misinterpretation and miscommunication can occur when trying to understand the requirements.
5. **Dependence on the tool:** The team should be well-trained on the tool and its features to avoid dependency on the tool and not on the requirement.
6. **Limited validation:** The validation techniques can only check the requirement that is captured and may not identify the requirement that is missed.
7. **Limited to functional requirements:** Some validation techniques are limited to functional requirements and may not validate non-functional requirements.

How Do We Perform Requirement Validation?

Here is a step-by-step guide on how to perform requirement validation:

Step 1: Gather Requirements

- Identify all relevant stakeholders, including users, project managers, and subject matter experts.
- Utilize various techniques to gather requirements, such as interviews, workshops, and surveys.
- Consolidate and document all gathered requirements, ensuring clarity and organization.

Unit 3: Software Requirements Engineering

Step 2: Analyze Requirements

- Scrutinize each requirement for clarity, ensuring it is well-defined and easy to understand.
- Assess the completeness of each requirement, ensuring it captures all necessary details and considerations.
- Check for consistency among requirements, identifying any conflicts or contradictions.
- Evaluate the alignment of each requirement with the project's overall objectives.

Step 3: Validate Requirements

- Employ inspections, where a team of reviewers examines the requirements for accuracy and consistency.
- Conduct reviews, where stakeholders provide feedback on the requirements, identifying potential issues and areas for improvement.
- Utilize prototyping to create mockups or simulations of the software to visualize and test the requirements.
- Consider using automated tools to check for syntax errors, format inconsistencies, and potential compliance issues.

Step 4: Document Findings

- Document the findings from the validation process, including any identified issues, their severity, and proposed solutions.
- Categorize the findings based on their type, such as ambiguity, incompleteness, inconsistency, or misalignment with objectives.
- Assign ownership and timelines for addressing the identified issues.

Step 5: Refine Requirements

- Collaborate with stakeholders to refine the requirements based on the validation findings.
- Incorporate the proposed solutions into the requirements documentation.
- Review the refined requirements to accurately reflect the customer's needs and align with the project's objectives.