

SOFTWARE ENGINEERING

UNIT – 2

TOPIC – 3

REQUIREMENTS ENGINEERING AND FEASIBILITY STUDY

1. Requirements Engineering (RE)

Definition:

Requirements Engineering (RE) is a structured process that involves understanding what a software system must do to meet the needs of its users and stakeholders. It involves gathering, analyzing, documenting, validating, and managing the requirements that specify what a system should do.

Key Goals of Requirements Engineering:

- **Define System Requirements:** Clearly describe what the software must accomplish to solve a specific problem or fulfil user needs.
- **Minimize Errors:** Reduce the chance of building a product that doesn't meet user expectations.
- **Guide Development:** Provide a clear blueprint that developers and project managers can follow throughout the software development lifecycle.

Why is Requirements Engineering Important?

- **Improves Communication:** It serves as a communication bridge between stakeholders and developers, making sure everyone understands the project goals.
- **Cost and Time Efficiency:** Catching errors or misunderstandings in the requirements phase is much cheaper and quicker than fixing them during later stages of development.
- **Reduces Risks:** Identifying issues early reduces project risks, such as time overruns, budget issues, and the development of features that are not needed.

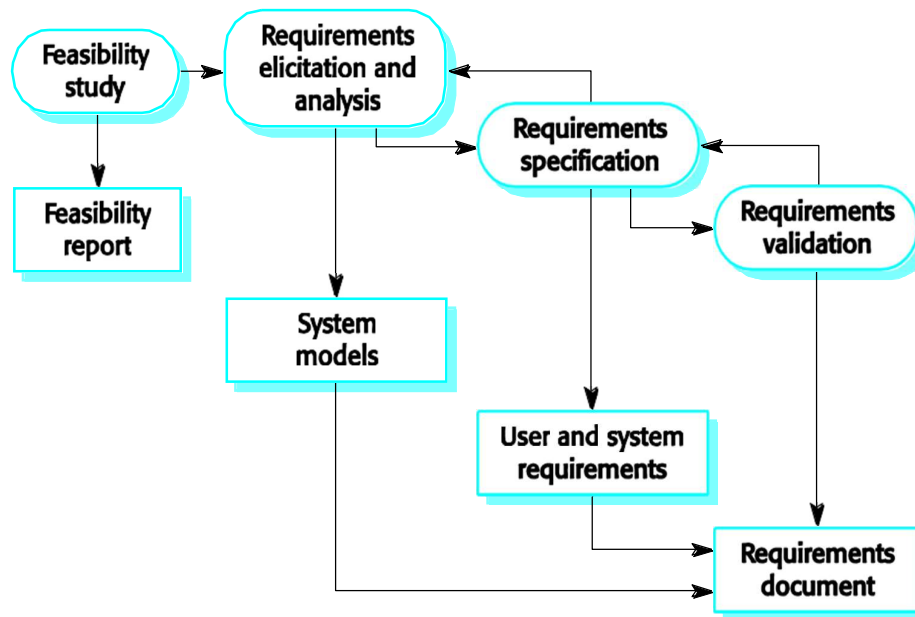
Common Problems if Requirements Engineering is Neglected:

- **Misunderstandings:** Without clear requirements, developers may misunderstand what needs to be built.

- **Scope Creep:** The project may grow out of control because requirements were not clearly defined from the beginning.
- **Project Failure:** Many software projects fail due to incomplete, unclear, or misunderstood requirements.

2. Steps in the Requirements Engineering Process

The Requirements Engineering process is divided into several steps, each contributing to the development of a clear set of software requirements:



1. **Feasibility Study:** A preliminary analysis that determines if the project is viable. Focuses on whether the project can be developed with available resources, technology, and time constraints.
2. **Requirements Elicitation:** Involves collecting the needs of stakeholders, including customers, end-users, and other parties. Techniques used for elicitation include interviews, questionnaires, workshops, and observations.

Example: For a hospital management system, requirements can be gathered from doctors, nurses, administrative staff, and patients to understand their specific needs.

3. **Requirements Analysis:** In this phase, the gathered requirements are analyzed to ensure they are clear, complete, consistent, and feasible. Conflicts between requirements are identified and resolved to ensure that they do not contradict each other.

Example: If a shopping app requires both "fast delivery" and "low shipping cost," analysis is needed to find a balance between these conflicting needs.

4. **Requirements Documentation:** All requirements are documented in a structured format, usually in a Software Requirements Specification (SRS) document. The SRS includes detailed descriptions of both functional requirements (what the system should do) and non-functional requirements (performance, security, usability).

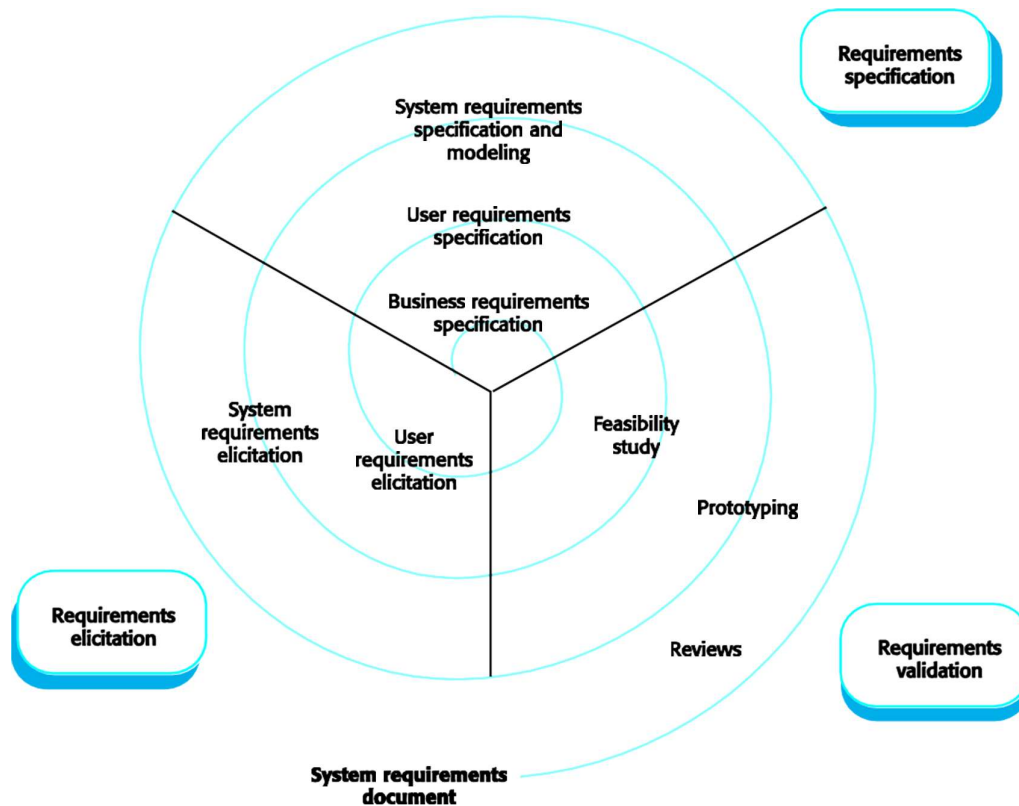
Example: The SRS for a banking software might include functional requirements like "the system shall support account transfers" and non-functional requirements like "transactions should process within two seconds."

5. **Requirements Validation:** Validation ensures that the documented requirements align with what stakeholders actually expect from the system. It involves reviewing the requirements with stakeholders to confirm that they accurately represent their needs.

Example: In a meeting, stakeholders might review and validate that the software's interface design matches their user experience expectations.

6. **Requirements Management:** This step involves handling changes to the requirements as the project evolves. Requirements management ensures that any updates to requirements are systematically controlled and communicated to all team members.

Example: If a new regulation requires added security features, requirements management tracks and integrates these changes into the project plan.



3. Feasibility Study

What is a Feasibility Study?

A feasibility study is a critical assessment conducted to determine if a proposed project or system is practical and achievable. It evaluates various factors like technology, budget, timeline, and overall value to decide whether the project should proceed.

Importance of a Feasibility Study:

- **Risk Mitigation:** Identifies potential risks in technology, costs, and operations before they become problems.
- **Efficient Resource Use:** Ensures that the company's time, money, and effort are invested in projects that are likely to succeed.
- **Informed Decision-Making:** Helps stakeholders decide whether to approve, reject, or modify the project based on concrete data.

4. Types of Feasibility Studies

Feasibility studies can be categorized into several types, each focusing on different aspects of a project's potential success:

1. **Technical Feasibility:** Evaluates whether the technology and tools required for the project are available and suitable.

Considerations: Does the development team have the technical expertise? Is the technology stable and scalable?

Example: Before developing an AI-based customer support system, the company assesses if their infrastructure supports machine learning algorithms and if the team has the skills to manage AI development.

2. **Operational Feasibility:** Assesses whether the project aligns with the organization's existing processes and satisfies user needs.

Considerations: Will the system integrate with current workflows? Will users be able to adapt to the new system?

Example: For a retail company implementing a new inventory management system, operational feasibility would check if employees can transition smoothly to the new system without disrupting sales.

3. **Economic Feasibility:** Analyzes the financial aspects to determine if the project is cost-effective.

Considerations: Do the benefits outweigh the costs? What is the expected return on investment (ROI)?

Example: Developing an e-commerce platform involves significant costs; the company must evaluate if the expected increase in online sales justifies these expenses.

4. **Legal Feasibility:** Ensures that the project complies with legal standards and regulations.

Considerations: Does the system adhere to data protection laws, copyright issues, or industry-specific regulations?

Example: A healthcare application must comply with laws like the Health Insurance Portability and Accountability Act (HIPAA) to protect patient data.

5. **Schedule Feasibility:** Evaluates whether the project can be completed within the desired time frame.

Considerations: Are there adequate resources to meet the deadlines? Is the timeline realistic?

Example: Developing a mobile app to be released in sync with a marketing campaign requires careful scheduling to ensure timely delivery.

5. Example of a Feasibility Study

Let's consider the example of a bank planning to develop a mobile banking app:

- **Technical Feasibility:** Assesses if the bank's IT team can develop the app using the latest mobile technology. Do they have the necessary skills and hardware?
- **Economic Feasibility:** Analyzes whether the app's development and maintenance costs are within the bank's budget and if it will attract enough users to justify the investment.
- **Operational Feasibility:** Evaluates if the mobile app will integrate seamlessly with the bank's existing systems and enhance customer satisfaction.
- **Legal Feasibility:** Ensures that the app will comply with financial industry regulations like secure data handling and user privacy.
- **Schedule Feasibility:** Checks if the development can be completed on time, especially if it's planned to launch during a specific financial quarter.

6. Importance of Feasibility Studies in Software Projects

- **Strategic Alignment:** Helps ensure that the project aligns with the strategic goals of the organization.
- **Avoids Wasted Investment:** Saves resources by identifying non-viable projects early.

- **Supports Business Growth:** Facilitates innovation and growth by investing in feasible projects with a high chance of success.

Requirements Engineering and Feasibility Study are essential components in software development. Requirements Engineering helps define what the software should achieve, ensuring that developers build the right product, while Feasibility Studies ensure that the project is practical, achievable, and beneficial. Together, these processes reduce risks, save costs, and guide the project to success by laying a strong foundation for all software development activities.