SOFTWARE ENGINEERING

UNIT – 2

TOPIC – 4

REQUIREMENTS ELICITATION AND ANALYSIS

1. Requirements Elicitation

Definition:

Requirements elicitation is the process of gathering and discovering the needs, expectations, and constraints from users, stakeholders, and anyone who will be affected by the system. This stage focuses on collecting as much relevant information as possible to understand what the software should do.

Goal of Requirements Elicitation:

The primary goal is to capture a complete set of requirements that accurately reflects what stakeholders need from the system. This helps to ensure that the development team has a clear understanding of the desired features and functionalities.

2. Who Are the Stakeholders?

Stakeholders are anyone with an interest in the development and use of the software system. Understanding who these stakeholders are is critical to gathering all relevant requirements.

1. **End-Users:**
    - These are the people who will directly use the software. They have firsthand knowledge of the tasks they need to accomplish using the system.
    - **Example:** In a hospital management system, end-users include doctors, nurses, and administrative staff who interact with the software daily.

2. **Managers:**
    - Managers ensure that the system aligns with the overall business strategy and objectives. They focus on how the software will impact the organization's operations.
    - **Example:** For a retail store, the manager will be interested in how the inventory management system supports sales and improves stock control.
3. **Developers:**
    - Developers are the technical team responsible for building the software according to the requirements gathered.
    - **Example:** Developers need detailed requirements to understand the specific technical tasks they must perform to create an e-commerce website.
4. **Regulators:**
    - Regulators ensure that the system meets industry standards, compliance regulations, and legal requirements.
    - **Example:** In financial software, regulators might include legal experts who ensure compliance with financial reporting standards and data protection laws.

3. Methods of Requirements Elicitation

The process of elicitation involves various methods to gather information from stakeholders, each suited for different situations. These methods aim to cover all aspects of what the software must achieve.

1. **Interviews:** This involves one-on-one or group discussions with stakeholders to understand their needs. It is a direct way to gather specific details.
    - **Types of Interviews:**
        - **Structured Interviews:** Have a predefined set of questions to ensure that all topics are covered systematically.
        - **Unstructured Interviews:** More flexible, allowing stakeholders to speak freely about their thoughts, which can uncover new insights.
    - **Example:** In developing an online banking system, a structured interview with customers might ask specific questions about their needs for security features and ease of transactions.

2. **Brainstorming Sessions:** This technique gathers stakeholders in a creative discussion to generate a wide range of ideas about the system.
    - **Phases of Brainstorming:**
        - **Idea Generation:** All participants share their ideas without evaluating them.
        - **Evaluation Phase:** The ideas are discussed, and the most relevant ones are selected for further development.
    - **Example:** During the design of a new mobile game, a brainstorming session might involve designers, developers, and marketers generating ideas for game mechanics and features.
3. **Facilitated Application Specification Technique (FAST):** FAST is a structured group meeting led by a facilitator, aimed at discussing specific requirements in detail. It encourages collaboration between stakeholders.
    - **Role of the Facilitator:** Ensures that the meeting stays focused, that everyone participates, and that misunderstandings are clarified.
    - **Example:** For a corporate HR management system, the facilitator might lead a session where HR managers, IT staff, and end-users discuss the key features required for tracking employee performance.
4. **Quality Function Deployment (QFD):** QFD is used to transform customer needs into engineering specifications. It involves detailed matrices (often called "House of Quality") to relate customer desires to system capabilities.
    - **Steps in QFD:**
        - Collecting Customer Needs: Understanding what features or services customers value the most.
        - Relating these needs to technical requirements: Defining how each need will be addressed by specific system functionalities.
    - **Example:** A car manufacturer uses QFD to convert customer demands for fuel efficiency and safety into specific design changes and technological innovations.
5. **Use Case Approach:** Use cases describe how different types of users (actors) interact with the system to achieve specific goals. It helps in visualizing the functional requirements of the system.
    - **Components of a Use Case:**
        - **Actors:** The roles that users play when interacting with the system.

        - **Scenarios:** Step-by-step description of the interactions between the actor and the system.
    - **Example:** For an online shopping site, a use case might detail the steps a customer takes to log in, search for products, add items to the cart, and make a purchase.
6. **Ethnography:** Ethnography involves observing users in their natural work environment to understand how they perform tasks and interact with current systems. It focuses on actual behaviour rather than just stated preferences.
    - **Benefits:** Provides insights into workflow inefficiencies and the real-world challenges users face.
    - **Example:** A researcher observing factory workers might notice that the existing software interface is causing delays in processing orders, leading to design improvements.

4. Challenges in Requirements Elicitation

Requirements elicitation is often complex due to various challenges, such as:

- **Communication Barriers:** Different stakeholders might have varying terminologies and perspectives, making it difficult to understand their needs.
- **Changing Requirements:** Stakeholders' needs can evolve over time, resulting in a moving target for requirements.
- **Unstated Requirements:** Users might not be aware of all their needs or assume that certain requirements are obvious.

5. Requirements Analysis

Definition:

Requirements analysis is the process of refining, validating, and prioritizing the gathered requirements. It aims to ensure that the requirements are complete, consistent, and feasible before the development process begins.

Key Objectives of Requirements Analysis:

- **Clarity:** Ensures that every requirement is clear and understandable to all stakeholders, reducing the risk of misunderstandings.
- **Conflict Resolution:** Identifies and resolves any conflicts between different stakeholders' needs and expectations.
- **Prioritization:** Determines the importance of each requirement to focus on the most critical features that deliver maximum value to users.

6. Steps in Requirements Analysis

1. **Gap Analysis:** To identify missing requirements or areas where the system might fail to meet business goals.
    - **Method:** Compare the current state of the system with the desired state to pinpoint gaps.
    - **Example:** For a customer support system, gap analysis might reveal that there is no feature for tracking customer complaints, which is essential for improving service.
2. **Conflict Resolution:** To handle conflicting requirements that arise from different stakeholder groups.
    - **Approach:** Facilitating discussions to find a compromise or prioritizing one requirement over another based-on project goals.
    - **Example:** Marketing wants to add more visual elements to an app, but the technical team is concerned about performance issues. Conflict resolution involves finding a balance between aesthetics and speed.
3. **Feasibility Analysis:** To check whether the requirements can be realistically implemented within the constraints of technology, budget, and time.
    - **Considerations:** Evaluates if the requirements align with the project's scope and the resources available.
    - **Example:** For a startup developing a machine learning product, feasibility analysis might determine that they lack the technical expertise or budget to implement certain advanced features.

7. Importance of Requirements Analysis

- **Avoids Scope Creep:** Ensures that the project remains focused on the agreed-upon requirements and does not expand unnecessarily.
- **Ensures System Quality:** By validating and prioritizing requirements, it helps create a well-defined product that meets user needs effectively.
- **Reduces Costs:** Catching issues early in the analysis phase can prevent expensive rework during later stages of development.

8. Practical Example of Requirements Analysis

Scenario: A transportation company is developing a fleet management system.

- **Stakeholders:** Fleet managers, drivers, IT support, logistics planners, and finance officers.
- **Gap Analysis:** Identified that the current system does not support real-time vehicle tracking, which is crucial for efficient fleet management.
- **Conflict Resolution:** Drivers want an easy-to-use interface, while fleet managers need detailed analytics. The solution involves creating different user views tailored to each group's needs.
- **Prioritization:** The top priority is to implement the vehicle tracking feature due to its impact on operational efficiency.

Requirements Elicitation and Requirements Analysis are critical phases in software development that ensure the final product aligns with user needs and business goals. These processes involve actively engaging stakeholders, employing various techniques to gather information, and rigorously analysing the requirements to ensure clarity, feasibility, and completeness. Properly executed, these activities set the foundation for successful software development, reducing risks, avoiding costly changes, and delivering value to both users and the organization.