Requirements Engineering: Summary and Scenarios

1 Overview of Requirements Engineering

Requirements engineering (RE) is the process of defining the services a customer needs from a system and the constraints under which it operates. The RE process involves finding out, analyzing, documenting, and validating these requirements.

2 Types of Requirements

- **Functional Requirements**: Describe what the system should do, including services, reactions to inputs, and behavior in specific situations.
- **Non-Functional Requirements**: Constraints on the system such as performance, security, and usability, often applying to the system as a whole.
- **Domain Requirements**: Constraints from the system's operational domain.

3 User and System Requirements

- **User Requirements**: High-level statements in natural language with diagrams for customers.
- **System Requirements**: Detailed descriptions of system functions, services, and constraints, often forming part of a contract.

4 Agile Methods and Requirements

Agile methods prioritize flexibility and incremental development, often using user stories instead of detailed system requirements. This approach is effective for business systems but less so for critical systems requiring thorough pre-delivery analysis.

5 Requirements Engineering Processes

- **Requirements Elicitation**: Gathering requirements through stakeholder interaction, discovery, classification, prioritization, and specification.
- **Requirements Analysis**: Understanding and refining the gathered requirements.
- **Requirements Specification**: Documenting the requirements in a structured format.
- **Requirements Validation**: Ensuring the requirements accurately reflect the customer's needs.
- **Requirements Management**: Handling changes in requirements throughout the development process.

6 Requirements Elicitation Techniques

- **Interviews**: Engaging stakeholders to gather information through structured or open-ended questions.
- **Ethnography**: Observing how people work to uncover implicit requirements.
- **Scenarios and User Stories**: Real-life examples of system use to relate requirements to practical situations.

7 Requirements Specification Methods

- **Natural Language**: Using descriptive sentences, possibly supplemented by diagrams.
- **Structured Specifications**: Limiting the writer's freedom by using templates or forms to ensure consistency.
- **Graphical Notations**: Using diagrams like UML to represent requirements.
- **Mathematical Specifications**: Formal methods to precisely define requirements, though not always understandable by customers.

8 Requirements Validation Techniques

- **Reviews**: Systematic analysis of requirements by stakeholders.
- **Prototyping**: Creating executable models to check requirements.
- **Test-Case Generation**: Developing tests to ensure requirements are verifiable.

9 Scenarios for Better Understanding

9.1 Scenario 1: Hospital Appointment System

- **User Requirement**: A patient should be able to book an appointment online.
- **Functional Requirement**: The system shall allow patients to search for available appointment slots and book them.
- **Non-Functional Requirement**: The system should confirm the appointment within 5 seconds of booking.

9.2 Scenario 2: Online Shopping Platform

- **User Requirement**: Users should be able to add items to their cart and checkout.
- **Functional Requirement**: The system shall allow users to view product details, add items to their cart, and proceed to checkout.
- **Non-Functional Requirement**: The system should handle up to 10,000 concurrent users during peak hours.

9.3 Scenario 3: Airline Reservation System

- **User Requirement**: A customer should be able to search for and book flights.
- **Functional Requirement**: The system shall display available flights based on search criteria and allow customers to book tickets.
- **Non-Functional Requirement**: The system should ensure transaction security using industry-standard encryption methods.

10 Conclusion

Requirements engineering is critical for developing systems that meet customer needs and function within specified constraints. Understanding the types of requirements, the processes involved, and effective elicitation and validation techniques is essential for successful system development.