CMP 519 Software Engineering

UNIT 03 - Requirements Engineering and Elicitation

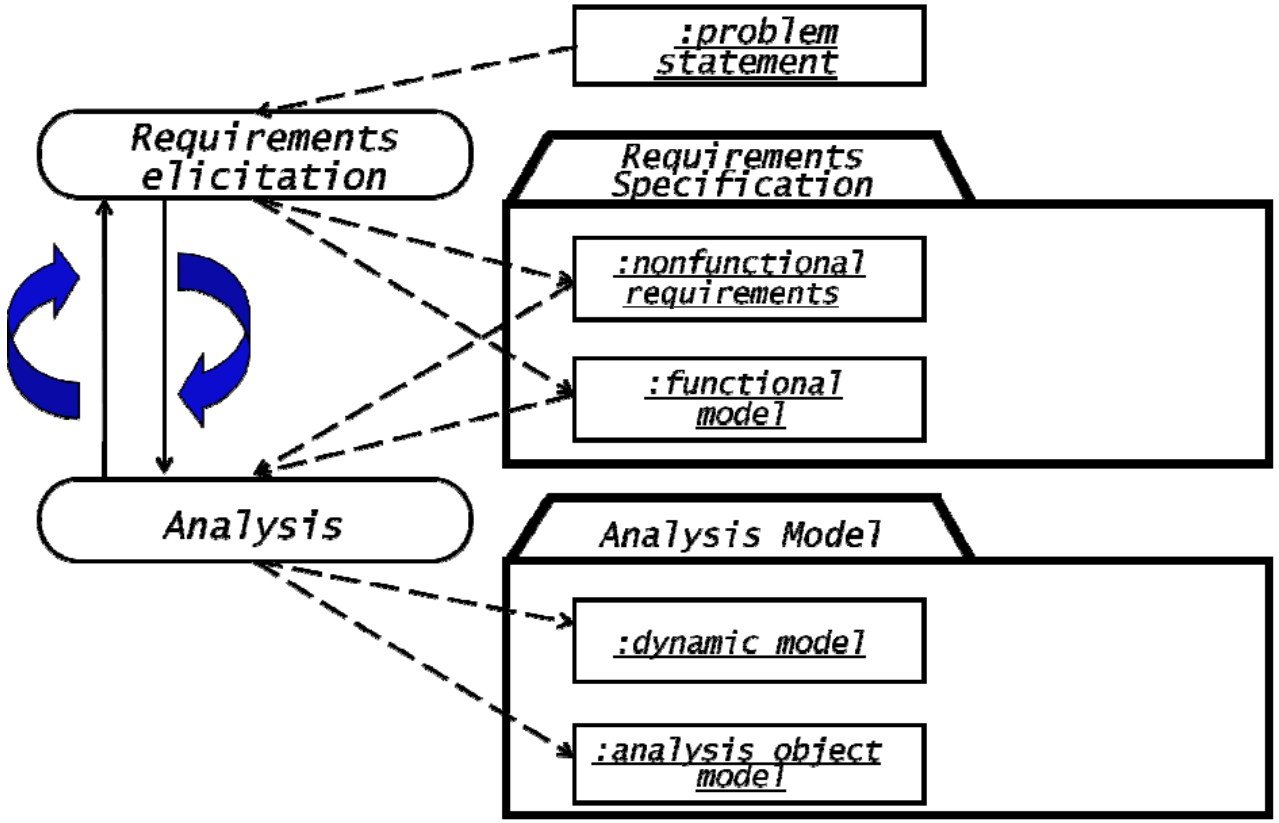
Instructor: Dr. Suresh Pokharel

DISCLAIMER

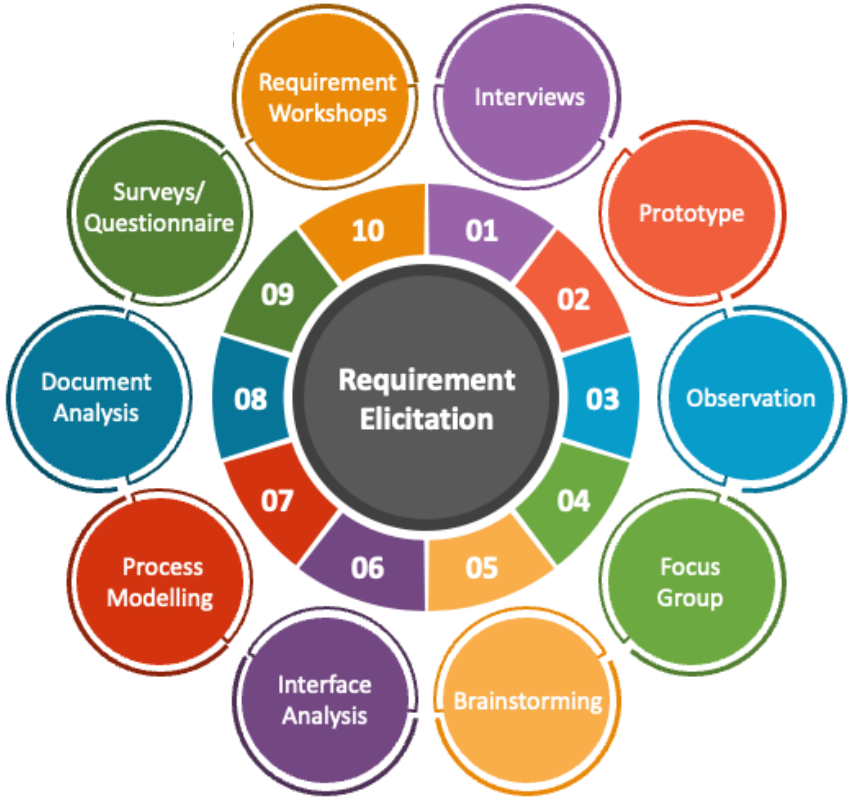
These slides are part of teaching materials for Software Engineering.

These slides do not cover all aspect of learning Software Engineering, nor are these be taken as primary source of information. As the core textbooks and reference books for learning the subject has already been specified and provided to the students, students are encouraged to learn from the original sources.

Requirement Elicitation Concepts



Requirement Elicitation Techniques



Requirement elicitation Techniques (most of them) are covered in chapter 1.

Importance of Stakeholder Involvement

- The process of discovering, documenting, and validating the requirements of a system.
- Involves understanding the needs of stakeholders and translating them into specific, measurable requirements.
- **Need**
  - Directly impacts project success: A well-defined requirements document significantly reduces the risk of project failure.
  - Cost-effective: Identifying and addressing issues early in the development cycle saves time and money.
  - Customer Satisfaction: Ensures that the final product meets the needs and expectations of users.

Importance of Stakeholder Involvement

- Stakeholders are individuals or groups who have an interest in the system's success.
- **Why Involvement is Crucial**:
  - Feel ownership and satisfaction
  - Communication and collaboration
  - Ensures system meets business needs.
  - Provides a diverse perspective to cover all use cases.
  - Helps identify potential risks early.
- **Key Stakeholders**:
  - End users.
  - Business owners.
  - System administrators.
  - Project managers.
  - Regulators and compliance officers.
  - Subject matter experts (SMEs).
  - Vendors and suppliers.

Importance of Stakeholder Involvement

- Key Stakeholders: Example
  - For an e-commerce website, stakeholders include customers (end users), the marketing team (business owners), IT staff (system administrators), and payment gateway providers (vendors).
  - In a hospital management system, stakeholders include doctors, nurses, administrators, patients, and regulatory agencies.

How do the priorities of different stakeholders (e.g., end users vs. regulators) influence requirements?

Challenges in Requirements Gathering

- *Communication Barriers*
  - Stakeholders might lack technical knowledge.
  - Developers might not understand domain-specific details.
- *Changing Requirements*
  - Requirements evolve due to market or organizational changes.
- *Conflicting Requirements and Priorities*
  - Different stakeholders may have opposing priorities and might have conflicting interest
- *Unstated or Implicit Requirements*
  - Hidden needs that stakeholders assume are obvious.
- *Incomplete or Ambiguous Requirements*
  - Vague or unclear requirements

Detailed Requirements Analysis

Completeness and Consistency

- Completeness:
  - All necessary requirements are identified.
  - No critical functionality is missing.
- Consistency:
  - Requirements do not contradict each other.
  - Terminology and concepts are uniform.

Clarity and Correctness

- Clarity:
  - Requirements are unambiguous and easily understood.
  - Avoid vague terms like "user-friendly."
- Correctness:
  - Requirements accurately reflect stakeholder needs.
  - Validated against original goals.

Importance of Stakeholder Involvement

- *Examples: Consistency and Completeness*
  - In a school management system, consistency ensures the term "student ID" is used uniformly instead of mixing terms like "ID" or "roll number."
  - For an online food delivery app, completeness ensures features like order placement, tracking, and payment are covered without gaps.

- *Examples: Clarity and Correctness*
  - A requirement stating "The system should allow users to reset their password via email" is clear, whereas "The system should improve account management" is vague.
  - A warehouse management system accurately reflecting stakeholder needs ensures inventory levels are tracked in real-time and errors are minimized.

1. How would you ensure completeness when gathering requirements for a new system?
2. Discuss the role of stakeholders to validate the correctness of requirements?

Detailed Requirements Analysis

Realism and Verifiability

- Realism:
  - Requirements must be achievable within technical, financial, and time constraints.
  - Consider system performance and scalability.

- Verifiability:
  - Requirements must be testable.
  - Define clear acceptance criteria.

Traceability of Requirements

- Ability to trace requirements from origin to implementation
- Benefits
  - Ensures all requirements are addressed.
  - Facilitates impact analysis for changes.
- Techniques
  - Use traceability matrices (see next slide)
  - Linking requirements to test cases

Traceability Matrices: Examples

Traceability Matrix

S no.	Req. ID	Req Desc	Scenario	TC Id	TC Desc	Test Result	Defect Id	Defect Status
1	BR 1.0	1. INBOX	1.Mails	TC_gmail_maill_01	Check the mail in inbox	Pass		
				TC_gmail_maill_02	Verify the open mail	Fail	Def_01	Open
				TC_gmail_maill_03	Verify the unread mail	Pass		
				TC_gmail_maill_04	Verify the Downloads and attachments	Fail	Def_02	Open
				TC_gmail_maill_05	Verify to navigate to different pages	Pass		
			2. Reply	TC_gmail_Rply_01	Verify to reply a mail	Pass		
				TC_gmail_Rply_02	Verify to reply all mails	Fail	Def_03	Open
			3. FWD	.....	.....	.....		
				.....	.....	.....		
				.....	.....	.....		

REQUIREMENT TRACEABILITY MATRIX

Project Name :	
Created By :	
Date :	

Test case IDs → / Req. IDs ↓	Req. desc.	Req. type	Test case no. 1	Test case no. 2	Test case no. 3	Test case no. 4	Test case no. 5	Test case no. 6	Test case no. 7	Test case no. 8	Test case no. 9	Test case no. 10
Requirement no. 1			✓						✓			
Requirement no. 2				✓								✓
Requirement no. 3										✓		
Requirement no. 4					✓			✓				✓
Requirement no. 5											✓	
Requirement no. 6						✓						
Requirement no. 7									✓			

Requirements Traceability Matrix

	Testing								Defects			
WBS Deliverables	Test Case ID	Test Description	TEST	UAT	QA	PROD	PRE-PROD	NON-PROD	Defective	Defect ID	Defect Description	Req. Status
WBS-001	TC001	Verify user can successfully register an account	Pass	Pass	Pass	Pass	Fail		No			Complete
WBS-001	TC002	Verify error message is displayed for invalid inputs	N/A	Fail	Pass	Fail	N/A		Yes	DEF001	Invalid email format	Complete
WBS-001	TC003	Verify user receives a confirmation email	Pass	N/A	Fail	N/A	Pass		Yes	DEF002	Email not sent	In Progress
WBS-002	TC004	Verify user can log in with valid credentials	Fail	Pass	N/A	N/A	N/A		NO			In Progress
WBS-002	TC005	Verify error message is displayed for incorrect login	Fail	Fail	N/A	Pass	Fail		Yes	DEF003	Incorrect username	Complete
WBS-002	TC006	Verify "Forgot Password" link redirects correctly	Fail	Fail	N/A	Pass	Pass		No			In Progress
WBS-003	TC007	Verify user can create a new post				N/A	N/A		No			Not Started
WBS-003	TC008	Verify error message is displayed for empty content	Pass	Pass	Pass	Fail	Fail		Yes	DEF004	Empty post content	Complete

Importance of Stakeholder Involvement

- *Examples: Realism and Verifiability*
  - A requirement like "The system must handle 10,000 concurrent users" is realistic if the infrastructure supports it and verifiable through load testing.
  - For a flight booking system, verifying "Users should receive confirmation emails within 5 seconds of booking" is achievable through email server performance testing.

1. What are the risks of including unrealistic requirements in a project?
2. How do you define verifiable acceptance criteria for non-functional requirements?
3. What challenges might arise in maintaining traceability throughout the project lifecycle?

Elicitation Activities

1. Identifying Actors and Stakeholders
2. Developing Scenarios and Use Cases
3. Refining Use Cases and User Stories
4. Identifying Relationships among Actors and Use Cases

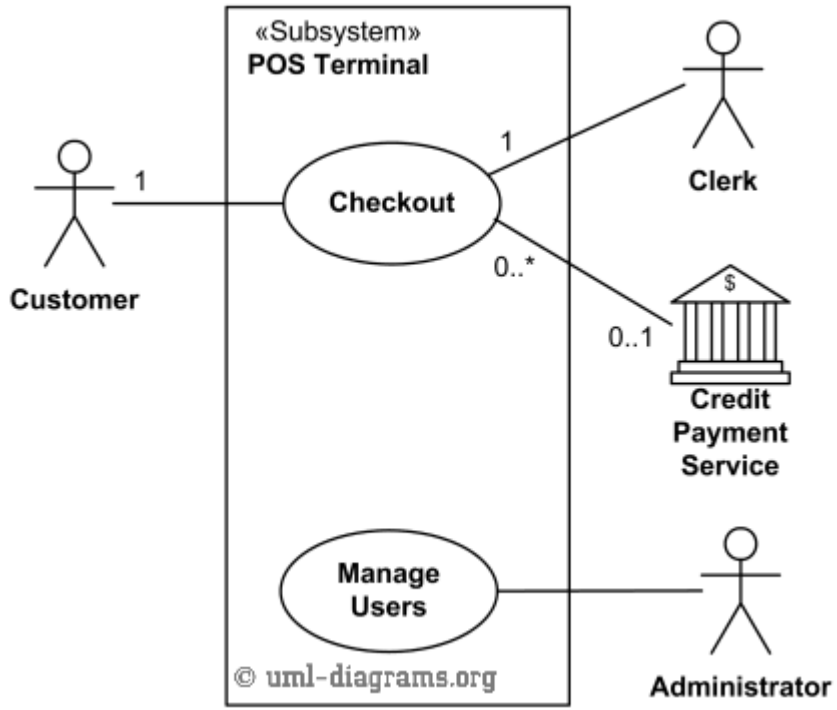
Elicitation Activities: Identifying Actors and Stakeholders

- **Actors**
  - Represent entities interacting with the system.
  - Examples: Users, external systems.
- **Stakeholder**
  - The people or groups affected by a software development project
- **Stakeholder Identification Methods**
  - Brainstorming sessions.
  - Stakeholder analysis.
- **Examples**
  - For a ride-sharing app, actors include riders, drivers, and payment systems.
  - For an online education platform, stakeholders include students, teachers, content creators, and platform administrators.

1. Can you identify potential actors and stakeholders for a library management system?
2. Are Actors and Stakeholders same thing?

Elicitation Activities: Developing Scenarios and Use Cases

- **Scenarios**:
  - Describe real-world interactions with the system.
  - Provide context for requirements.
- **Use Cases**:
  - Focus on a specific goal or task.
  - Include actors, preconditions, steps, and outcomes.



Use Case Diagram

Elicitation Activities: Refining Use Cases and User Stories

- **Refinement Process**:
  - Add more details to use cases and user stories.
  - Address edge cases and exceptions.
  - Example: Expanding a use case "Book a Flight" to handle scenarios like cancellations or rescheduling
- **Techniques**:
  - Walkthroughs with stakeholders.
  - Iterative feedback cycles.

1. Develop a scenario for POS in supermarket

User Stories

- User stories are commonly used in adaptive or agile methodologies
- Short, high-level descriptions of required functionality expressed in customer terms.
- A typical user story has the form: “As a <role>, I want <goal/desire> so that <benefit>.”
- Just enough information so that the developers can produce a reasonable estimate of the effort to implement it.
- The aim is to avoid some of the waste that often happens in projects where detailed requirements are gathered early but become invalid before the work begins.

Elicitation Activities: Identifying Relationships among Actors and Use Cases

- **Relationships**:
  - Dependency: One use case depends on another.
  - Inclusion: A use case includes another.
  - Extension: Variations of a base use case.
- **Visualization**:
  - Use case diagrams to represent relationships.

Elicitation Activities: Identifying Relationships among Actors and Use Cases

For POS system in supermarket, perform the following tasks.

- Identifying Actors and Stakeholders
- Developing Scenarios (one scenario and steps is fine) and Use Cases (one use case diagram is fine)
- Refining Use Cases
- Identifying Relationships among Actors and Use Cases

Managing Requirements

- Maintaining Traceability
- Negotiating Specifications with Clients
- Documenting Requirements

Maintaining Traceability

- **Why It's Important**
  - Helps track changes and their impacts.
  - Ensures alignment with business objectives.
- **Tools**
  - Requirement management tools (e.g., Jira, DOORS).
- **Examples**
  - A traceability matrix links the requirement "Generate monthly reports" to its design document and test cases.
  - For a smart home system, traceability ensures that "Control lights remotely" is implemented and tested across all device platforms.

Negotiating Specification with Clients

- Negotiation Process:
  - Prioritize requirements with stakeholders.
  - Resolve conflicts and balance trade-offs.
- Best Practices:
  - Use prototypes to clarify expectations.
  - Document agreed-upon specifications.
- Examples:
  - For a mobile app, negotiating whether to prioritize offline functionality or faster development.
  - In a CRM system, deciding between detailed customer analytics or simpler usability features.

What Factors should be considered when balancing trade-off during negotiations?

Documenting Requirements

- Purpose
  - Serves as a reference throughout the project.
  - Provides clarity to the development team.
- Formats
  - Text-based documents.
  - Models and diagrams.
- Key Sections
  - Functional requirements.
  - Non-functional requirements.
  - Assumptions and constraints.
  - Ensure End-to-end Journey of business is reflected
- Examples
  - A requirement document for a weather app includes sections like "Display current temperature" and "Support notifications for severe weather alerts."
  - A document for an inventory system includes "Track stock levels" and "Generate restocking alerts."

END OF UNIT 03