

Software Engineering

[CACS253]

BCA 4th Sem

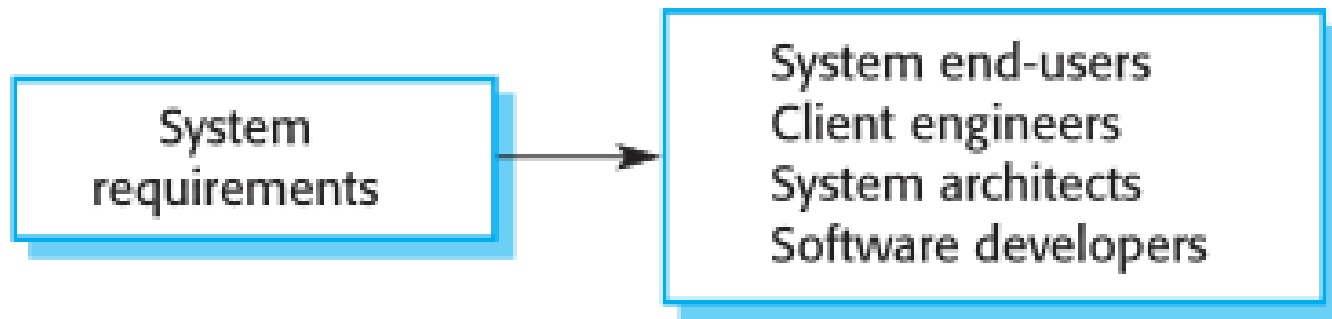
Er. Sital Prasad Mandal
Mechi Multiple Campus
Bhadrapur, Jhapa, Nepal

<https://ctal-softwareeng.blogspot.com/>

- 1. System and Software Requirements**
- 2. Type of Software Requirements:**
 - a. Functional
 - b. Non-Functional Requirements
- 3. Domain Requirements**
- 4. User Requirements**
- 5. Elicitation-and Analysis of Requirements:**
 - a. Overview of Techniques
 - b. View Points
 - c. Interviewing
 - d. Scenarios
 - e. Use-Case
 - f. Ethnography
 - g. Requirement
 - h. Validation
 - i. Requirement Specification
 - j. Feasibility

System Requirements

- System requirements are more detailed descriptions of the software system's functions, services, and operational constraints.
- The system requirements document (sometimes called a functional specification) should define exactly what is to be implemented.
- It may be part of the contract between the system buyer and the software developers.



Software Requirements

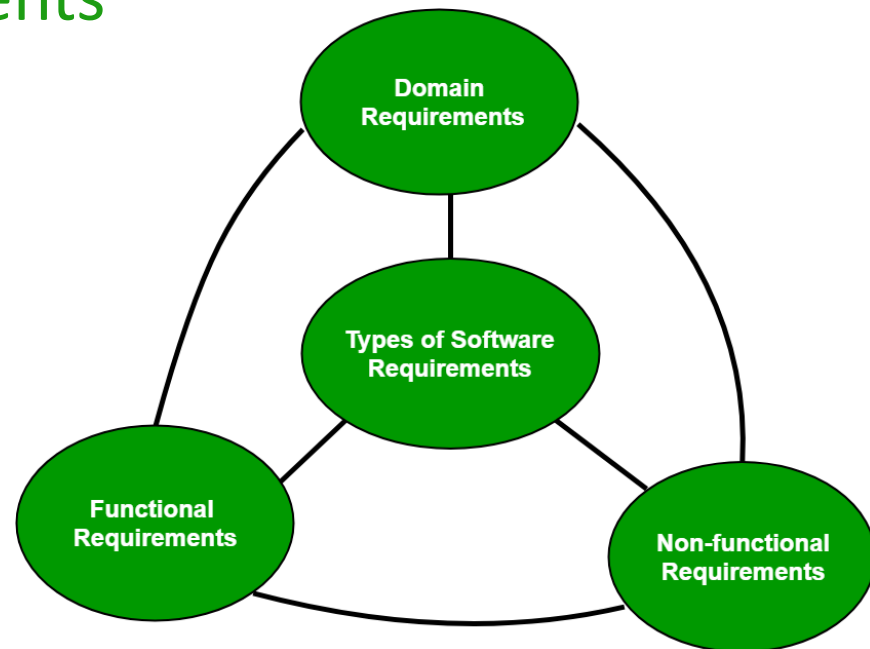
What is Software Requirement? (software specification)

- Requirement is a condition or capability possessed by the software or system component in order to solve a real world problem.
- The problems can be to automate a part of a system, to correct shortcomings of an existing system, to control a device, and so on.
- ***IEEE defines requirement as***
 - (1) A condition or capability needed by a user to solve a problem or achieve an objective.
 - (2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents.
 - (3) A documented representation of a condition or capability as in (1) or (2).

Types Software Requirements

A software requirement can be of 3 types:

1. Functional requirements
2. Non-functional requirements
3. Domain requirements



Types Software Requirements

Library Management System

Function Requirements

- ▶ **Add Article:** New entries must be entered in database
- ▶ **Update Article:** Any changes in articles should be updated in case of update
- ▶ **Delete Article:** Wrong entry must be removed from system
- ▶ **Inquiry Members:** Inquiry all current enrolled members to view their details
- ▶ **Inquiry Issuance:** Inquiry all database articles
- ▶ **Check out Article:** To issue any article must be checked out
- ▶ **Check In article:** After receiving any article system will reenter article by Checking
- ▶ **Inquiry waiting for approvals:** Librarian will generates all newly application which is in waiting list
- ▶ **Reserve Article:** This use case is used to reserve any book with the name of librarian, it can be pledged

Non-function Requirements

- ▶ **Safety Requirements:** The database may get crashed at any certain time due to virus or operating system failure. So, it is required to take the database backup.
- ▶ **Security Requirements:** We are going to develop a secured database for the university. There are different categories of users namely teaching staff, administrator, library staff ,students etc., Depending upon the category of user the access rights are decided.
- ▶ **Software Constraints:** The development of the system will be constrained by the availability of required software such as database and development tools. The availability of these tools will be governed by

Types Software Requirements

- Functional Requirements
 - *Are things the product must do.*
 - What the new product must do?
- Non-functional Requirements (also known as **quality requirements**)
 - *Are qualities the product must have.*
 - What qualities must the product have?
 - Does it have to be fast?
 - Or easy to use?
 - Secure from hacking?
- Constraints
 - *Are global issues that shape the requirements*
 - For example : use some existing hardware, software or business practice, or it might have to fit within a defined budget or be ready by a defined date..

Types Software Requirements

q Functional requirements

- Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.

q Non-functional requirements

- constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.

q Domain requirements

- Requirements that come from the application domain of the system and that reflect characteristics of that domain

Types Software Requirements

- **Functional Requirements**
 - **The Scope of the Product** – product boundaries
 - **Functional and Data Requirements** – things product must do and data manipulated by the functions
- **Non-functional Requirements**
 - **Look & Feel** – intended appearance
 - **Usability** – based on intended users
 - **Performance** – how fast, big, accurate, safe, reliable
 - **Operational** – intended operating environment
 - **Maintainability & Portability** – how changeable
 - **Security** – security, confidentiality, integrity
 - **Cultural & Political** – human factors

Assignment Types Software Requirements

Functional Requirements —

- Functionality or services that the system is expected to provide.
- Functional requirements may also explicitly state what the system shouldn't do.
- Functional requirements specification should be:
 - Complete: All services required by the user should be defined
 - Consistent: should not have contradictory definition (also avoid ambiguity--don't leave room for different interpretations)
- ❖ A functional requirement is a behavior that the product should do or support.
- ❖ These are basically expressed in the form of input and output.
- ❖ For example, the input might be data related to a user's online purchase, including name, address, item purchased, and payment information.
- ❖ The output in this scenario might be a notification after the transaction is performed.
- ❖ Functional requirements should be expressed with depth and specificity.
- ❖ One way to represent functional requirements is by using an information flow diagram.

Assignment Types Software Requirements

Non Functional Requirements —

- ❖ If Functional Requirements are what a product should do, Non-Functional Requirements are how well a product should do them.
- ❖ Non-Functional Requirements really address the quality factor of the product.
- ❖ In fact, they're sometimes even called quality requirements for this very reason.

Non-functional requirements address product issues including

- ✓ accuracy
- ✓ dependability
- ✓ security
- ✓ usability
- ✓ efficiency
- ✓ performance
- ✓ maintainability

For example, for an e-commerce application non-functional requirements would be that emails should be delivered to users within two hours of purchasing a product.

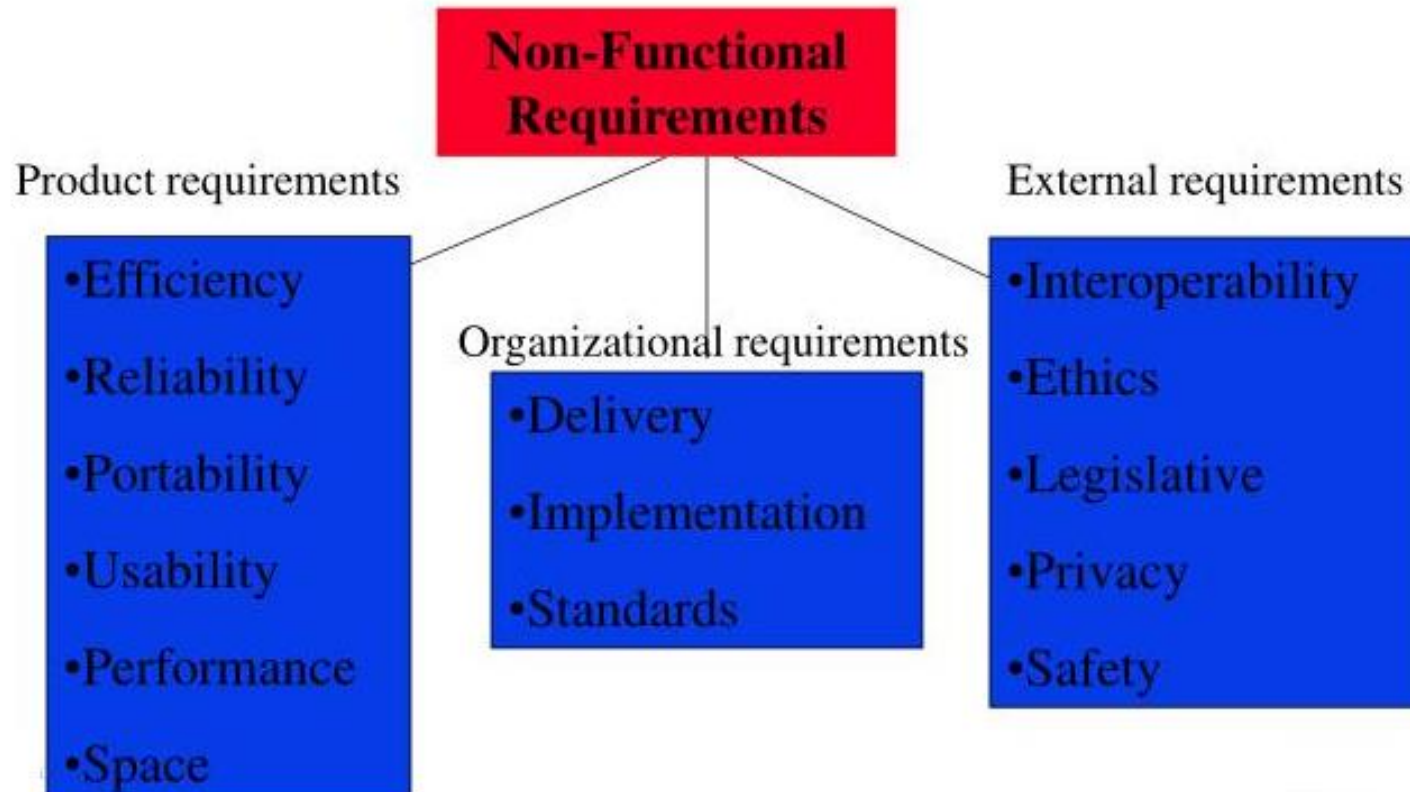
Assignment Types Software Requirements

FUNCTIONAL vs NONFUNCTIONAL REQUIREMENTS

	Functional requirements	Nonfunctional requirements
Objective	Describe what the product does	Describe how the product works
End result	Define product features	Define product properties
Focus	Focus on user requirements	Focus on user expectations
Documentation	Captured in use case	Captured as a quality attribute
Essentiality	They are mandatory	They are not mandatory, but desirable
Origin type	Usually defined by user	Usually defined by developers or other tech experts
Testing	Component, API, UI testing, etc. Tested before nonfunctional testing	Performance, usability, security testing, etc. Tested after functional testing
Types	External interface, authentication, authorization levels, business rules, etc.	Usability, reliability, scalability, performance, etc.

Assignment Types Software Requirements

Non-Functional requirements classifications:



Domain Requirements

- ❖ Requirements which are derived from the application domain of the system instead from the needs of the users are known as **domain requirements**.
- ❖ These requirements may be new functional requirements or specify a method to perform some particular computations.
 - ☐ Derived from the application domain and describe system characteristics and features that reflect the domain
 - ☐ May be new functional requirements, constraints on existing requirements or define specific computations
 - ☐ If domain requirements are not satisfied, the system may be unworkable

Domain Requirements

Library system domain requirements

- q There shall be a standard user interface to all databases which shall be based on the Z39.50 standard.
- q Because of copyright restrictions, some documents must be deleted immediately on arrival. Depending on the user's requirements, these documents will either be printed locally on the system server for manually forwarding to the user or routed to a network printer.

Domain requirements problems

- q Understandability
 - Requirements are expressed in the language of the application domain
 - This is often not understood by software engineers developing the system
- q Implicitness
 - Domain specialists understand the area so well that they do not think of making the domain requirements explicit

User Requirements

- q Should describe functional and non-functional requirements so that they are understandable by system users who don't have detailed technical knowledge
- q User requirements are defined using natural language, tables and diagrams

Problems with natural language

- q Lack of clarity
 - Precision is difficult without making the document difficult to read
- q Requirements confusion
 - Functional and non-functional requirements tend to be mixed-up
- q Requirements amalgamation
 - Several different requirements may be expressed together

Elicitation-and Analysis of Requirements:

1. Elicitation-and Analysis of Requirements:

- a. Interviewing
- b. Scenarios
- c. Use-Case
- d. Ethnography
- e. Requirement
- f. Validation
- g. Requirement Specification
- h. Feasibility

Elicitation-and Analysis of Requirements:

Overview of Techniques Requirement Elicitation and Analysis

- Requirement elicitation is the practice of collecting the requirements of a system from users, customers and other stakeholders.
- This practice is also sometimes referred to as "Requirement Gathering".

Requirement Elicitation Methods:

There are a number of requirement elicitation methods:

- Interviews
- Brainstorming Session
- FAST
- Quality Function Deployment {QFD}
- Uscase Approach

Requirement Elicitation Methods:

Interviews: In this requirement gathering team Conduct an interview session with

- Entry level personal
- Middle level stakeholder
- Manager
- Users of the software(most important)

In that interview session team asks these types of questions.

Questions before successful deployment of project

- Any problem with existing system?
- Any calculation errors?
- How are the previous system works?
- Any additional functionality
- Any specific problem
- User requirements

Questions after installation or in Maintenance Phase

- Any calculation error
- Any additional functionality
- Goal of proposed product

Requirement Elicitation Methods:

Brainstorming Sessions

Brainstorming is a group creativity technique by which efforts are made to find a conclusion for a specific problem by gathering a list of ideas spontaneously contributed by its members.

It creates an atmosphere which is free from criticism and judgement and allow members to subject and explore ideas.

Mainly group of "**4-10 people(depend on project and Organization)**" attend the brainstorming session.

Brainstorming has two Phases:

Generation Phase: Offers as many ideas as possible and no Merits and demerits of the ideas as discussed.

Consolidation Phase: Ideas are discussed, revised and organised.

Requirement Elicitation Methods:

Facilitated Application Specification Technique(FAST)

This approach encourages the creation of a joint team of customers and developers who works together.

Fast Goal:

- 1- identify the problem
- 2- solution requirements
- 3- Propose elements of solutions
- 4- Negotiate different approach

- **A Meeting** is conducted at the neutral site and attended by both software engineers and customers.
- **Rules** for preparation and participation established.
- **An Agenda** is suggested that is formal enough to cover all important but informal enough to encourage the free flows of ideas.
- **A Facilitator**(can be a, customer, developer, or an outsider) controls the meeting.

Requirement Elicitation Methods:

Quality Function Deployment:

In this technique customer satisfaction is of prime concern, hence it emphasizes on the requirements which are valuable to the customer.

3 types of requirements are identified –

Normal requirements –

In this the objective and goals of the proposed software are discussed with the customer. Example – normal requirements for a result management system may be entry of marks, calculation of results, etc

Expected requirements –

These requirements are so obvious that the customer need not explicitly state them. Example – protection from unauthorized access.

Exciting requirements –

It includes features that are beyond customer's expectations and prove to be very satisfying when present. Example – when unauthorized access is detected, it should backup and shutdown all processes.

<https://ctal-softwareeng.blogspot.com/>

Requirement Elicitation Methods:

Use Case Approach:

This technique combines text and pictures to provide a better understanding of the requirements. The use cases describe the 'what', of a system and not 'how'. Hence, they only give a functional view of the system. The components of the use case design includes three major things – Actor, Use cases, use case diagram.

Actor –

It is the external agent that lies outside the system but interacts with it in some way. An actor maybe a person, machine etc. It is represented as a stick figure. Actors can be primary actors or secondary actors.

Primary actors – It requires assistance from the system to achieve a goal.

Secondary actor – It is an actor from which the system needs assistance.

Use cases –

They describe the sequence of interactions between actors and the system. They capture who(actors) do what(interaction) with the system. A complete set of use cases specifies all possible ways to use the system.

Use case diagram –

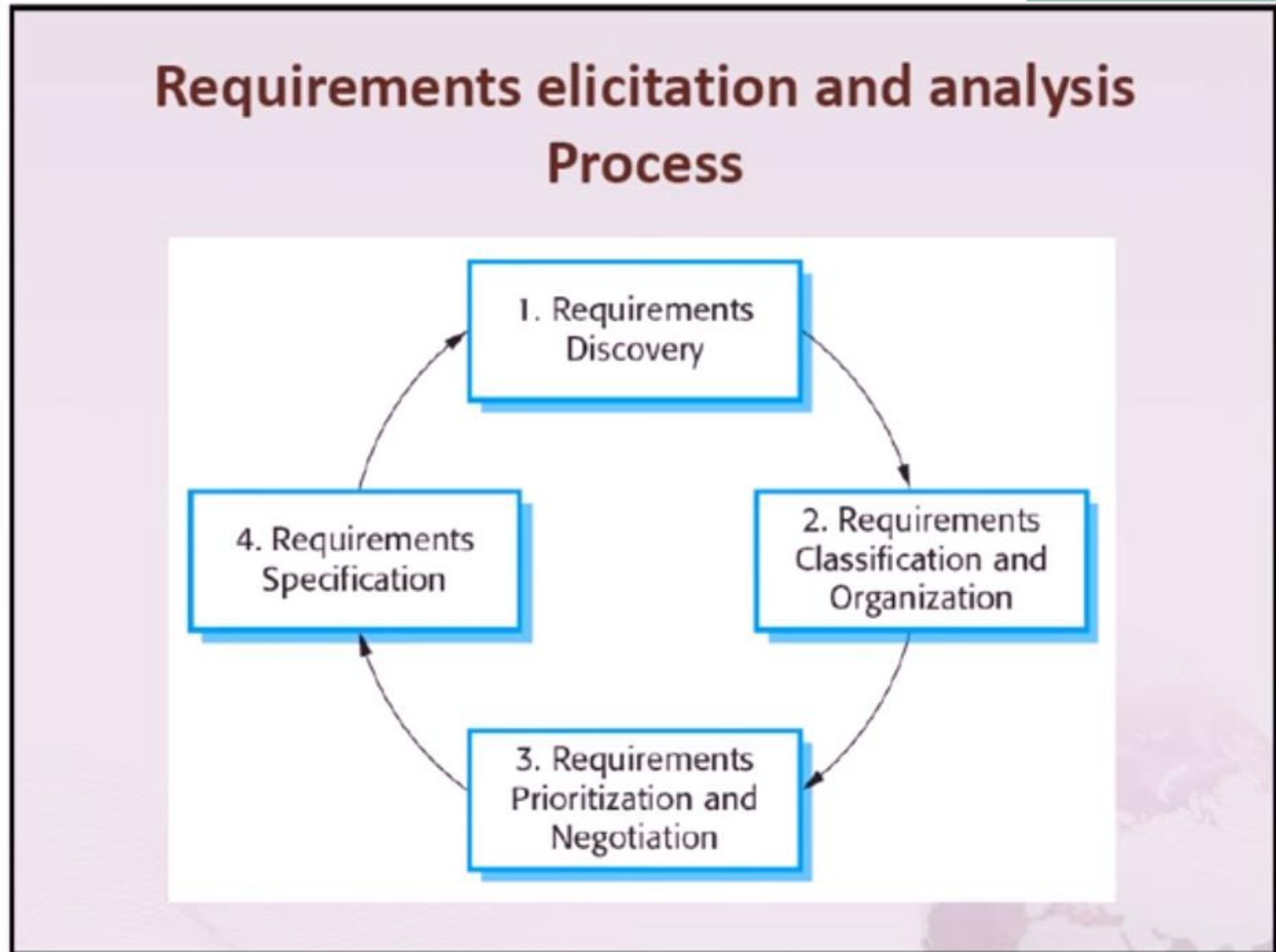
A use case diagram graphically represents what happens when an actor interacts with a system. It captures the functional aspect of the system.

A stick figure is used to represent an actor.

An oval is used to represent a use case.

A line is used to represent a relationship between an actor and a use case.

Elicitation-and Analysis of Requirements:



Elicitation-and Analysis of Requirements:

Process activities

- Requirements discovery
 - Interacting with stakeholders to discover their requirements. Domain requirements are also discovered at this stage.
- Requirements classification and organisation
 - Groups related requirements and organises them into coherent clusters.
- Prioritisation and negotiation
 - Prioritising requirements and resolving requirements conflicts.
- Requirements specification
 - Requirements are documented and input into the next round of the spiral.

Elicitation-and Analysis of Requirements:

Problems of requirements elicitation

- Stakeholders don't know what they really want.
- Stakeholders express requirements in their own terms.
- Different stakeholders may have conflicting requirements.
- Organisational and political factors may influence the system requirements.
- The requirements change during the analysis process. New stakeholders may emerge and the business environment change.

Elicitation-and Analysis of Requirements:

Interviewing

- Formal or informal interviews with stakeholders are part of most RE processes.
- Types of interview
 - Closed interviews based on pre-determined list of questions
 - Open interviews where various issues are explored with stakeholders.
- Effective interviewing
 - Be open-minded, avoid pre-conceived ideas about the requirements and are willing to listen to stakeholders.
 - Prompt the interviewers to get discussions going using a springboard question, a requirements proposal, or by working together on a prototype system.

Elicitation-and Analysis of Requirements:

Interviews in practice

- Normally a mix of closed and open-ended interviewing.
- Interviews are good for getting an overall understanding of what stakeholders do and how they might interact with the system.
- Interviews are not good for understanding domain requirements
 - Requirements engineers cannot understand specific domain terminology;
 - Some domain knowledge is so familiar that people find it hard to articulate or think that it isn't worth articulating.

Elicitation-and Analysis of Requirements:

Scenarios

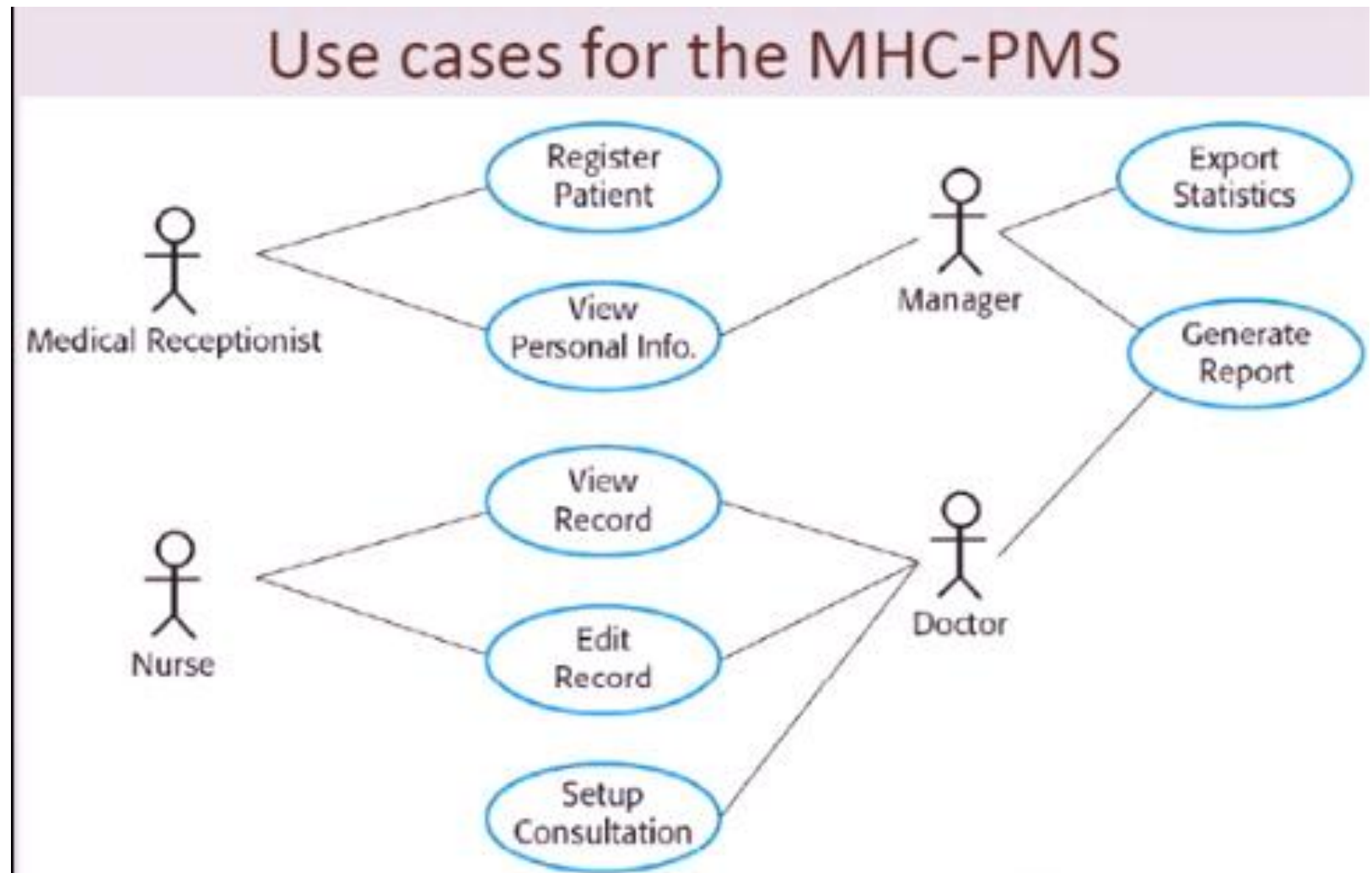
- Scenarios are real-life examples of how a system can be used.
- They should include
 - A description of the starting situation;
 - A description of the normal flow of events;
 - A description of what can go wrong;
 - Information about other concurrent activities;
 - A description of the state when the scenario finishes.

Elicitation-and Analysis of Requirements:

Use cases

- Use-cases are a scenario based technique in the UML which identify the actors in an interaction and which describe the interaction itself.
- A set of use cases should describe all possible interactions with the system.
- Sequence diagrams may be used to add detail to use-cases by showing the sequence of event processing in the system.

Elicitation-and Analysis of Requirements:



Elicitation-and Analysis of Requirements:

Requirements validation

- Concerned with demonstrating that the requirements define the system that the customer really wants.
- Requirements error costs are high so validation is very important
 - Fixing a requirements error after delivery may cost up to 100 times the cost of fixing an implementation error.

Elicitation-and Analysis of Requirements:

Requirements checking

- **Validity**. Does the system provide the functions which best support the customer's needs?
- **Consistency**. Are there any requirements conflicts?
- **Completeness**. Are all functions required by the customer included?
- **Realism**. Can the requirements be implemented given available budget and technology
- **Verifiability**. Can the requirements be checked?

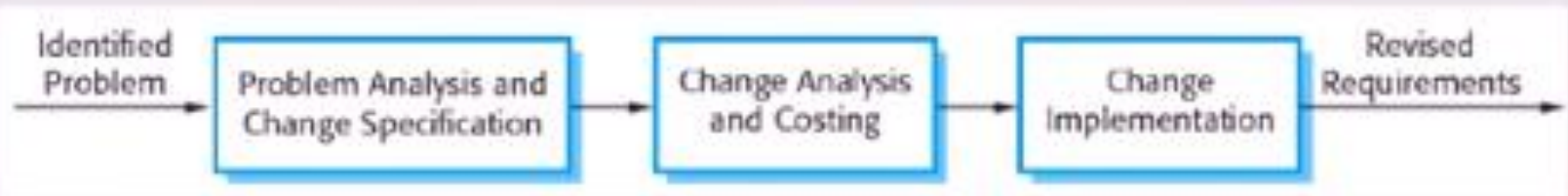
Elicitation-and Analysis of Requirements:

Requirements validation techniques

- Requirements reviews
 - Systematic manual analysis of the requirements.
- Prototyping
 - Using an executable model of the system to check requirements.
- Test-case generation
 - Developing tests for requirements to check testability.

Elicitation-and Analysis of Requirements:

Requirements change management

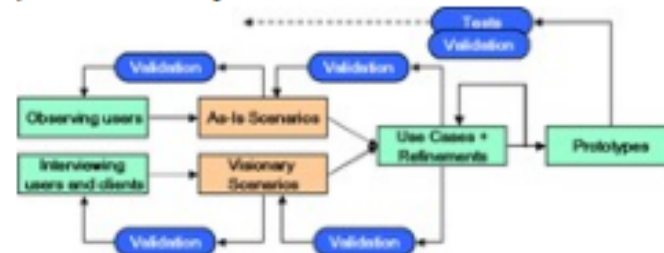


Last topics & Outlook

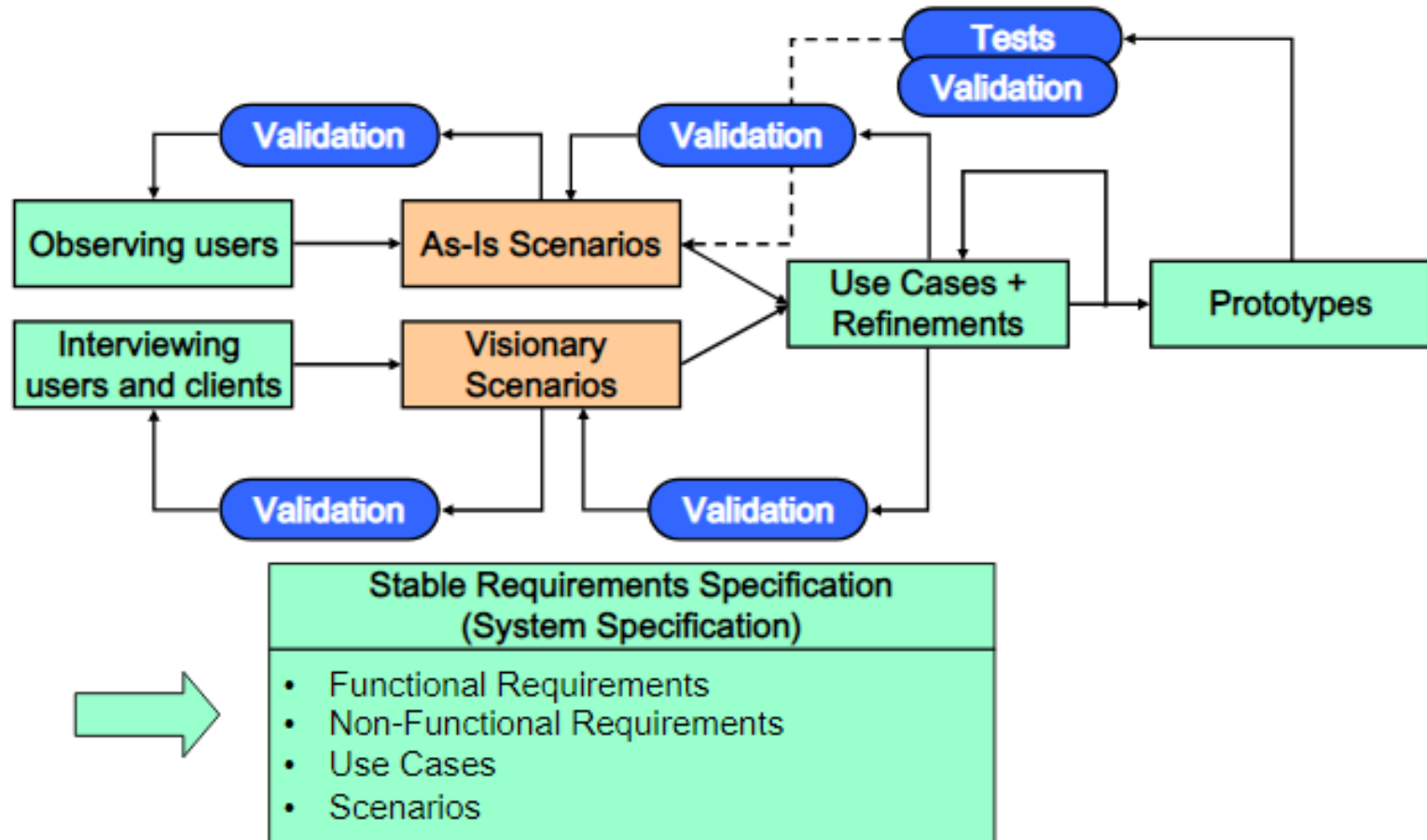
- ♦ What are requirements?
 - ♦ Functional vs. non-functional requirements
- ♦ Process of requirements elicitation
- ♦ 1. Step: How to elicit requirements
 - ♦ Analyzing existing systems (Tests, Manuals)
 - ♦ Interviews

Today:

- ♦ Elicitation
 - ♦ Observations
 - ♦ Write Down Requirements (→ Scenarios)
 - ♦ Aggregate Scenarios (→ Use Cases) and Refine Use Cases
 - ♦ Requirement Validation
- ♦ Requirements Analysis
 - ♦ Map Use Cases to the Analysis Object Model



Last topics & Outlook



Process of Requirements Elicitation: The Requirements Elicitation Cycle

Question Practices

1. Discuss the significance and use of requirement engineering. What are the problems in the formulation of requirements?
2. Requirements analysis is unquestionably the most communication intensive step in the software engineering process. Why does the communication path frequently break down ?
3. What are crucial process steps of requirement engineering ? Discuss with the help of a diagram.
4. Discuss the present state of practices in requirement engineering. Suggest few steps to improve the present state of practice.
5. Explain the importance of requirements. How many types of requirements are possible and why ?
6. What do you understand with the term “requirements elicitation”? Discuss any two techniques in detail.

Question Practices

7. List out requirements elicitation techniques. Which one is most popular and why ?
8. Describe facilitated application specification technique (FAST) and compare this with brainstorming sessions.
9. Discuss quality function deployment technique of requirements elicitation. Why an importance or value factor is associated with every requirement
10. Explain the use case approach of requirements elicitation. What are use-case guidelines ?
11. What are components of a use case diagram. Explain their usage with the help of an example.
12. Consider the problem of library management system and design the following:
 - (i) Problem statement
 - (ii) Use case diagram
 - (iii) Use cases

Question Practices

14. What is software requirements specification (SRS) ? List out the advantages of SRS standards. Why is SRS known as the black box specification of a system ?
15. What is the role of SRS in the development process of a software? Explain the characteristics of good SRS in detail.
16. Explain the requirement validation techniques.
17. Discuss the difference between the following: (a) Functional & nonfunctional requirements (b) User & system requirements
17. Consider the problem of railway reservation system and design the following.

(i) Problem statement (ii) Use case diagram (iii) Use cases