Module 3: Requirements Engineering

Outline

1 Basic Overview of Requirements Engineering

What is Requirements Engineering?

- The process of establishing the services that a customer requires from a system and the constraints under which it operates and is developed.
- The system requirements are the descriptions of the system services and constraints that are generated during the requirements engineering process.

Why Requirements Engineering?

- **Many projects fail:**
  - Because they start implementing the system.

    Without determining whether they are building what the customer really wants.

- Thus It is important to learn Requirements Engineering Techniques.

Goals of requirements Engineering

- Fully understand the user requirements.
- Remove inconsistencies, anomalies, etc. from requirements.
- Document requirements properly in an SRS document.

Who Carries Out Requirements Engineering?

- The person who undertakes requirements:
  - Known as Requirements Engineer. Or better as systems analyst:
  - Collects data pertaining to the product
  - Analyzes collected data:
    - To understand what exactly needs to be done.
  - Writes the **Software Requirements Specification** (SRS) document.

When is it Done?

- Final output of this phase:
  - Software Requirements Specification (SRS) Document.
- The SRS document is reviewed by the stakeholder/contractor.
  - Reviewed SRS document forms the basis of all future development activities.

Different types of Software Requirement

- User Requirements and System Requirements
- Functional Requirements, Non-functional Requirements, and Domain Constraints

User and System Requirements

User requirements definition

1. The Mentcare system shall generate monthly management reports showing the cost of drugs prescribed by each clinic during that month.

System requirements specification

1.1 On the last working day of each month, a summary of the drugs prescribed, their cost and the prescribing clinics shall be generated.
1.2 The system shall generate the report for printing after 17.30 on the last working day of the month.
1.3 A report shall be created for each clinic and shall list the individual drug names, the total number of prescriptions, the number of doses prescribed and the total cost of the prescribed drugs.
1.4 If drugs are available in different dose units (e.g. 10mg, 20mg, etc) separate reports shall be created for each dose unit.
1.5 Access to drug cost reports shall be restricted to authorized users as listed on a management access control list.

Functional, Non-functional, and Domain Requirements

- **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.
  - May state what the system should not do.
- **Non-functional requirements**
  - Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.
  - Often apply to the system as a whole rather than individual features or services.
- **Domain requirements**
  - Constraints on the system from the domain of operation

Known and Unknown Requirements

- Known Requirements
  - Those requirements that a stakeholder believes should be implemented
- Unknown Requirements
  - Forgotten by the stakeholder because they are not needed right now.
- Undreamt Requirements
  - Stakeholder may not be able to think of new requirements due to limited domain knowledge.

Feasibility Study

- A feasibility study decides whether or not the proposed system is worthwhile.

- A short focused study that checks
  - If the system contributes to organisational objectives;
  - If the system can be engineered using current technology and within budget;
  - If the system can be integrated with other systems that are used.

System stakeholders

- Any person or organization who is affected by the system in some way and so who has a legitimate interest
- Stakeholder types
  - End users
  - System managers
  - System owners
  - External stakeholders

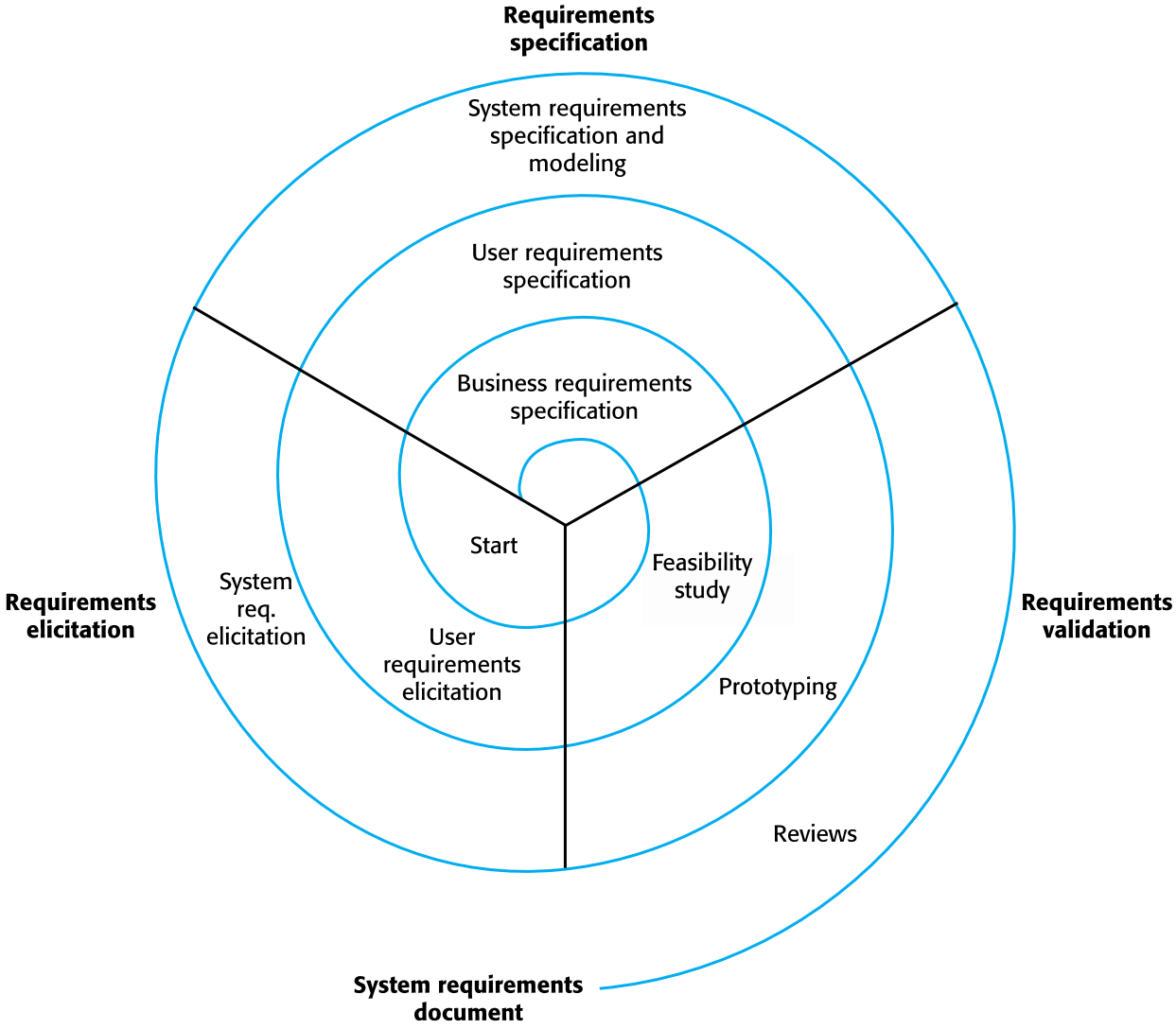
2. Requirements Engineering Processes

Requirements Engineering Process

Four Key Activities are:

- 1. Requirements Elicitation and Analysis
- 2. Requirements Specification
- 3. Requirements Validation

A spiral view of the requirements engineering process



1. Requirements Elicitation and Analysis

- Involves technical staff working with customers to find out about the application domain, the services that the system should provide and the system's operational constraints.
- May involve end-users, managers, engineers involved in maintenance, domain experts, trade unions, etc. These are called *stakeholders.*
- *Involves two activities:*
  - *1) Requirements Elicitation*
  - *2) Requirements Analysis*

Requirements Engineering Process

1.1 Requirements Elicitation/Gathering

- *Requirements Elicitation Activities*
  - 1. Studying the existing documentation (Domain Analysis)
  - 2. Interview
  - 3. Task analysis
  - 4. Scenario analysis
  - 5. Form analysis
  - 6. Brainstorming
  - 7. Use case based Requirements Elicitation
  - 8. Facilitated Application Specification Technique (FAST)

1.1 Requirements Elicitation/Gathering

Task vs Scenario

- Task
  - Software provides a set of services.
  - Services supported by a software
- Scenario
  - A task can have many scenario of operations.
- Example:
  - Task: Issue book service in a library
  - Scenario:
    - Book issued successfully.
    - Book is reserved and cannot be issued.
    - Maximum book issue limit reached.

1.1. Requirements Elicitation

- **Challenges in Eliciting requirements**
  - Home exercise for you!

1.2 Requirements Analysis

- Main purpose of requirements analysis:
  - Clearly understand the user requirements,
  - Detect inconsistencies, ambiguities, and incompleteness.
- Incompleteness and inconsistencies:
  - Resolved through further discussions with the end-users and the customers.

1.2 Requirements Analysis

- Requirements analysis involves:
  - Obtaining a clear, in-depth understanding of the product to be developed,
  - Remove all ambiguities and inconsistencies from the initial customer perception of the problem.

- Involves two activities:
  1. Requirements classification and organisation
  2. Prioritisation and negotiation

1.2 Requirements Analysis

- It is quite difficult to obtain:
  - A clear, in-depth understanding of the problem:
    - Especially if there is no working model of the problem.
- Some anomalies and inconsistencies can be very subtle:
  - Escape even most experienced eyes.
  - If a formal model of the system is constructed,
    - Many of the subtle anomalies and inconsistencies get detected.

2. Requirements Specification

- Main aim of requirements specification:
  - Systematically organize the requirements arrived during requirements analysis.
  - Document requirements properly.

- The requirements may be part of a contract for the system development
  - It is therefore important that these are as complete as possible.

2. Requirements Specification

Software Requirements Document (Software Requirements Specification, SRS)

- SRS is the official statement of what is required of the system developers.
- Should include both a definition of user requirements and a specification of the system requirements.
- It is NOT a design document. As far as possible, it should set of WHAT the system should do rather than HOW it should do it.

2. Requirements Specification - SRS

SRS: A Contract Document

- SRS document is a contract between the development team and the customer.
- Once the SRS document is approved by the customer,
  - Any subsequent controversies are settled by referring the SRS document.
- The final product will be acceptable to the customer:
  - As long as it satisfies all the requirements recorded in the SRS document.

2. Requirements Specification - SRS

- The SRS document is known as **black-box specification**:
  - The system is considered as a black box whose internal details are not known.
  - Only its visible external (i.e. input/output) behavior is documented.

- SRS document concentrates on:
  - What needs to be done
  - Carefully avoids the solution ("how to do") aspects.

2. Requirements Specification - SRS

Properties/Characteristics of a Good SRS Document

1. It should be Correct.
2. It should be Concise.
3. It should specify what the system must do and not say how to do it.
4. Easy to change., i.e. it should be well-structured.
5. It should be Consistent.
6. It should be Complete.
7. It should be Traceable.
8. It should be Verifiable.

2. Requirements Specification

- SRS document, normally contains three important parts:
  - Functional requirements,
  - Non-functional (Performance) requirements,
  - Design Constraints.

2. Requirements Specification

- **Functional requirements describe:**
  - A set of high-level requirements
  - Each high-level requirement:
    - takes in some data from the user
    - outputs some data to the user
  - Each high-level requirement:
    - might consist of a set of identifiable functions

2. Requirements Specification

- **Functional requirements**
- For each high-level requirement:
  - Every function is described in terms of:
    - Input data set
    - Output data set
    - Processing required to obtain the output data set from the input data set.

2. Requirements Specification

Non-functional Requirements

- Characteristics of the system which can not be expressed as functions:
  - Maintainability,
  - Portability,
  - Usability, etc.

2. Requirements Specification

Non-functional Requirements

- Non-functional requirements include:
  - Reliability issues,
  - Performance issues:
    - Example: How fast the system can produce results
      - so that it does not overload another system to which it supplies data, etc.
  - Human-computer interface issues,
  - Interface with other external systems,
  - Security, maintainability, etc.

2. Requirements Specification

Design Constraints

- Describe things that must be followed while designing and developing the system.

2. Requirements Specification

Organization of SRS

1. Introduction
   - 1.1 Purpose
   - 1.2 Product Scope
   - 1.3 Definition, Abbreviation, Glossary
   - 1.4 References
   - 1.5 Overview
2. Overall General Description
   - 2.1 Product Perspective
   - 2.2 Product Functions
   - 2.3 User Characteristics
   - 2.4 Constraints
   - 2.5 Assumptions and Dependencies
   - 2.6 Apportioning of Requirements
3. Specific Requirements
   - 3.1 External Interfaces
   - 3.2 Functional Requirements
   - 3.3 Nonfunctional Requirements
   - 3.4 Design Constraints

3. 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.

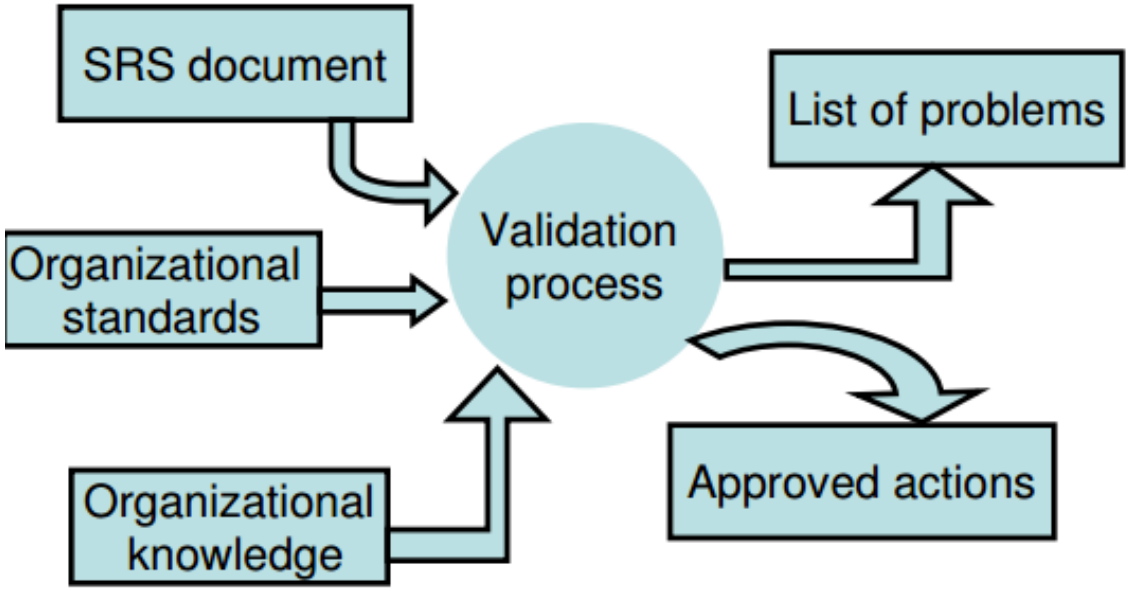
3. Requirements Validation

Requirements Checking

- Validity
- Consistency
- Completeness
- Realism
- Verifiability
- Conformance to standards

3. Requirements Validation

Requirements validation Process



3. Requirements Validation

Requirements validation techniques

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

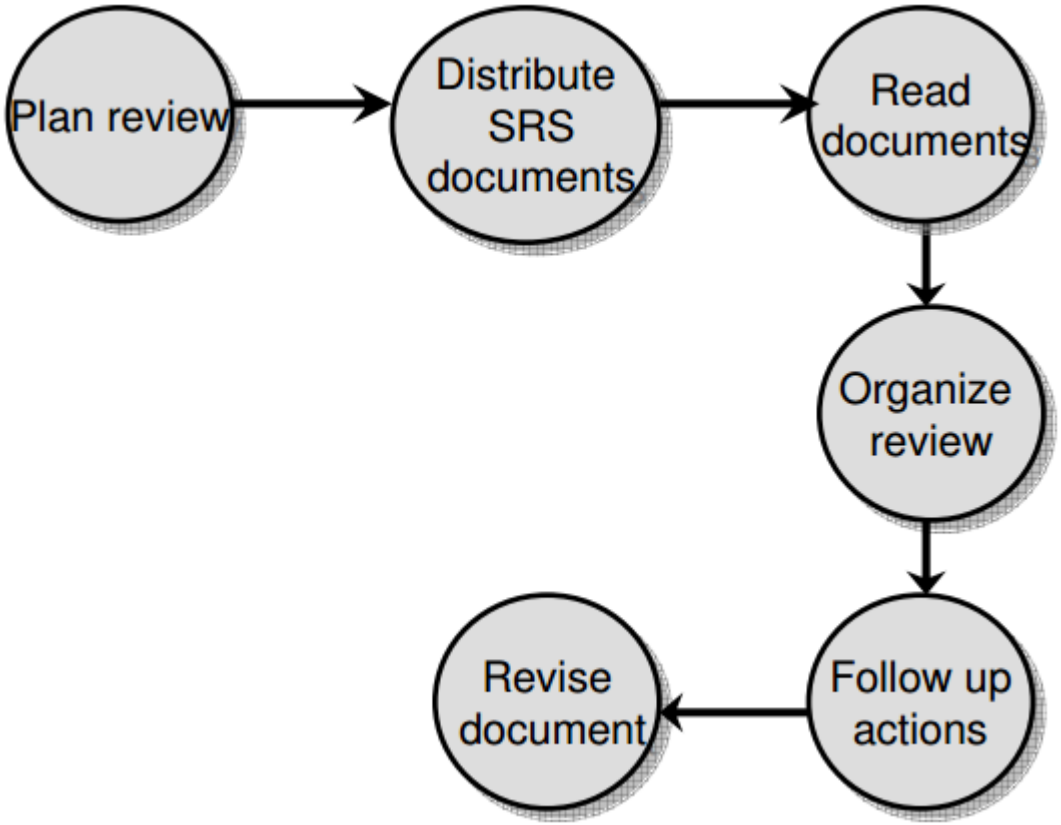
3. Requirements Validation

Requirements reviews

- Regular reviews should be held while the requirements definition is being formulated.
- Both client and contractor staff should be involved in reviews.
- Reviews may be formal (with completed documents) or informal. Good communications between developers, customers and users can resolve problems at an early stage.

3. Requirements Validation

Requirements review Process



3. Requirements Validation

Requirements reviews

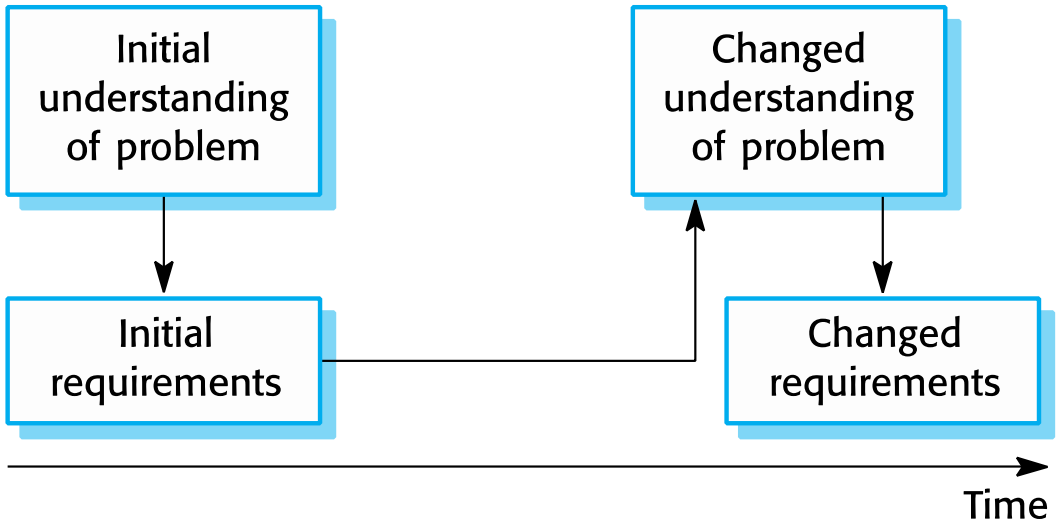
- **Reviews Check**
  - Verifiability: Is the requirement realistically testable?
  - Comprehensibility: Is the requirement properly understood?
  - Traceability: Is the origin of the requirement clearly stated?
  - Adaptability: Can the requirement be changed without a large impact on other requirements?

2.3 Requirements Management

Changing Requirements

- The business and technical environment of the system always changes after installation.
- The people who pay for a system and the users of that system are rarely the same people.
- Large systems usually have a diverse user community, with many users having different requirements and priorities that may be conflicting or contradictory.

Requirements evolution



Requirements Management

- Requirements management is the process of managing changing requirements during the requirements engineering process and system development.
- New requirements emerge as a system is being developed and after it has gone into use.
- Enduring requirements and Volatile requirements.

Requirements Management

- You need to keep track of individual requirements and maintain links between dependent requirements
  - Why?
    - To assess the impact of requirements changes.

Requirements Management Planning

- Requirements management decisions:
  - Requirements identification
    - Uniquely identify each requirements.
  - A change management process
    - To access the impact/cost of change
  - Traceability policies
    - Policies to define inter-requirements relationships
  - Tool support
    - Tool support for requirements storage, change management, traceability management.

Requirements Management Planning

- Requirements management decisions:
  - **Traceability**
    - Traceability is concerned with the relationships between requirements, their sources and the system design
      - Source traceability: Links from requirements to stakeholders who proposed these requirements
      - Requirements traceability: Links between dependent requirements
      - Design traceability: Links from the requirements to the design

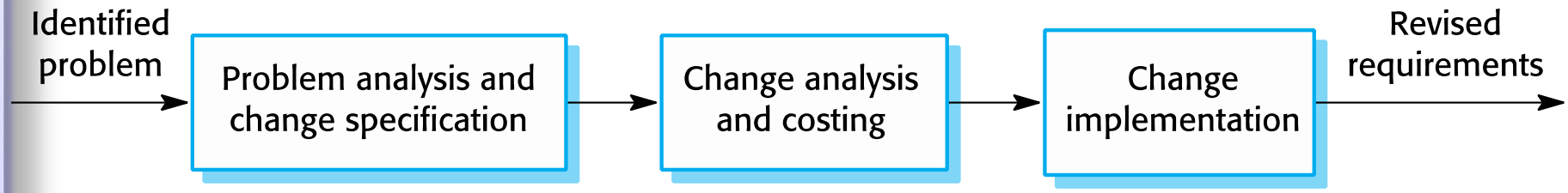
Requirements Management Planning

- Requirements management decisions:
  - **Traceability**

Req id	1.1	1.2	1.3	2.1	2.2	2.3	3.1	3.2
1.1		D	R					
1.2			D			D		D
1.3	R			R				
2.1			R		D			D
2.2								D
2.3		R		D				
3.1								R
3.2							R	

A Traceability matrix

Requirements Change Management



Suggested Readings

- Sommerville I, Software Engineering, Pearson Education, 10th Edition, 2016, Chap 4 Introduction, Sec 4.1, 4.2, 4.3, 4.5
- Mall R., “Fundamentals of Software Engineering”, Fourth Edition, PHI, 2016. Chapter 4 Introduction, Sec 4.1, 4.2