(11224) INTRODUCTION TO SOFTWARE ENGINEERING

Lecture 5: Systems Requirements

Shereen Fouad

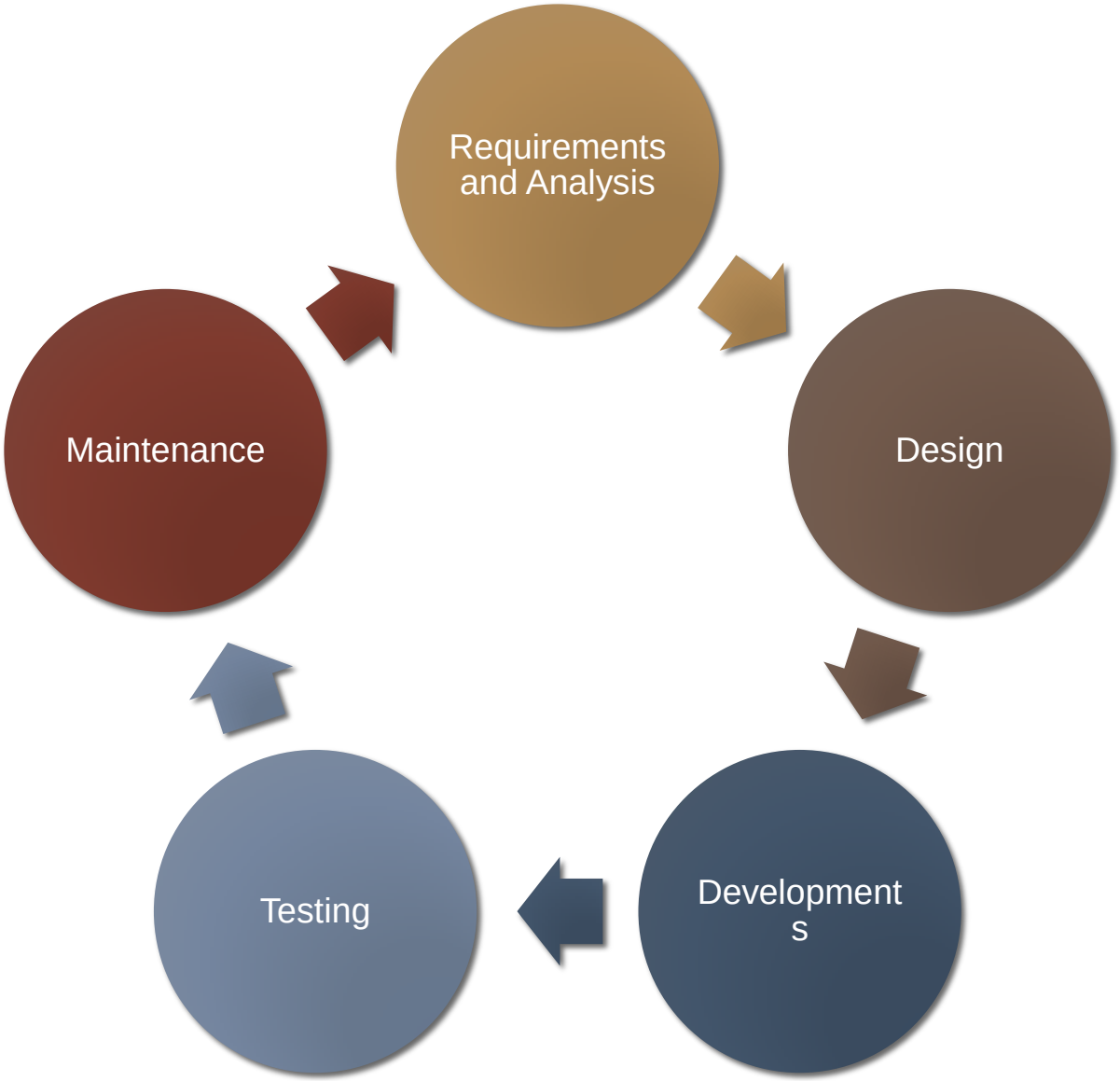
Overview

- Motivation: Software Lifecycle
- What is Requirements?
- Requirements Engineering Components
- Types of requirements
  - Functional, non-functional and domain Requirements
- Requirements specification
- Real life Example

Software Processes

Which activities should I start with?

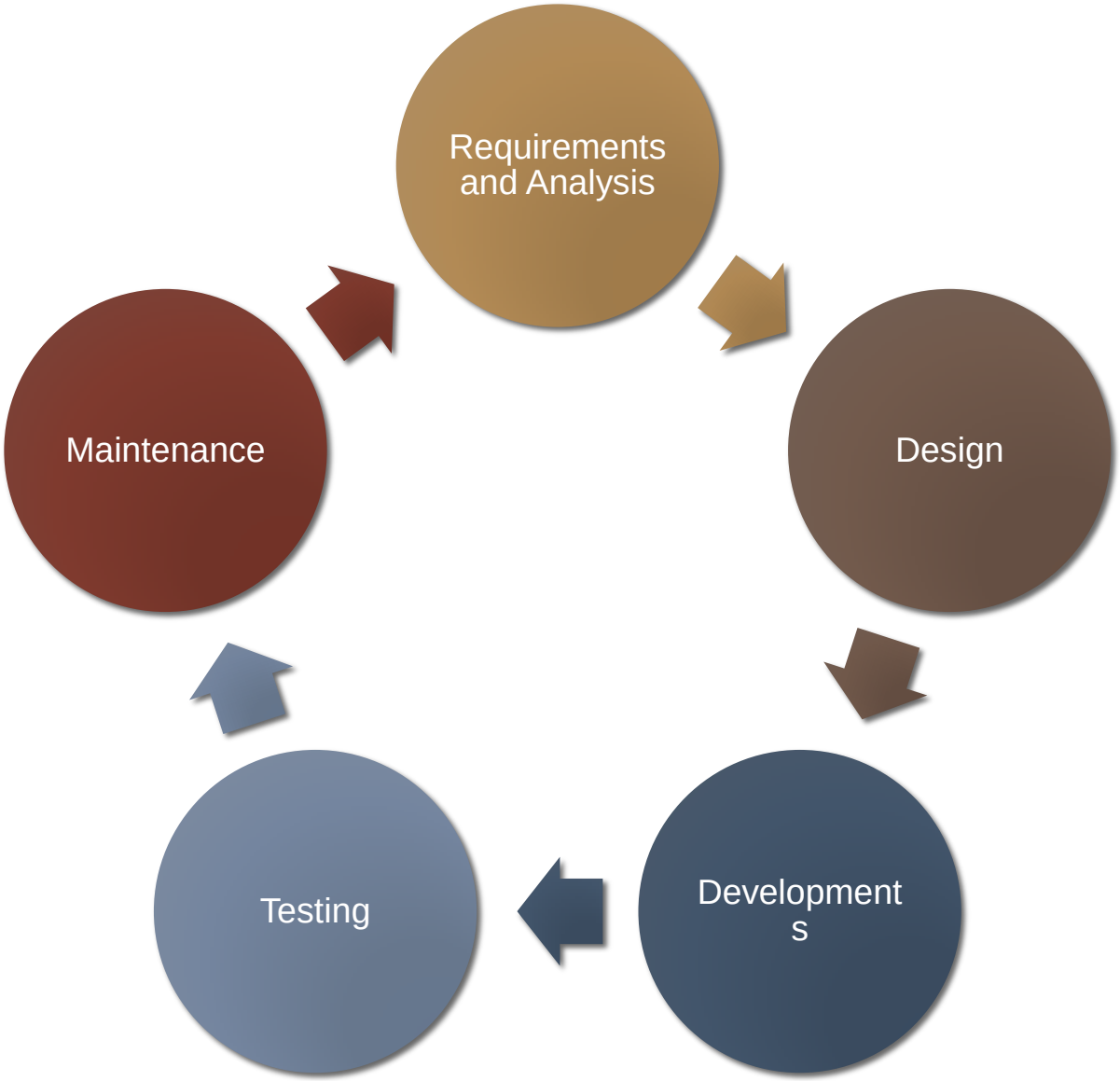
Why?



Software Processes

To start, two questions need to be answered:

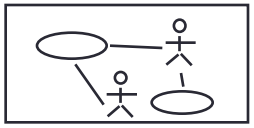
- How can we identify the purpose of a system?
- What is inside, what is outside the system?



Software Lifecycle Activities

Software Lifecycle Activities

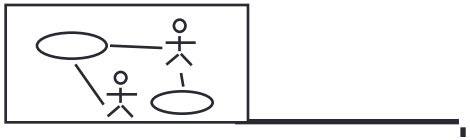
Requirements Elicitation	Requirements Analysis	System Design	Detailed Design	Implemen-tation	Testing



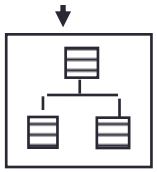
Use Case Model

Software Lifecycle Activities

Requirements Elicitation	Requirements Analysis	System Design	Detailed Design	Implemen-tation	Testing



Expressed in terms of

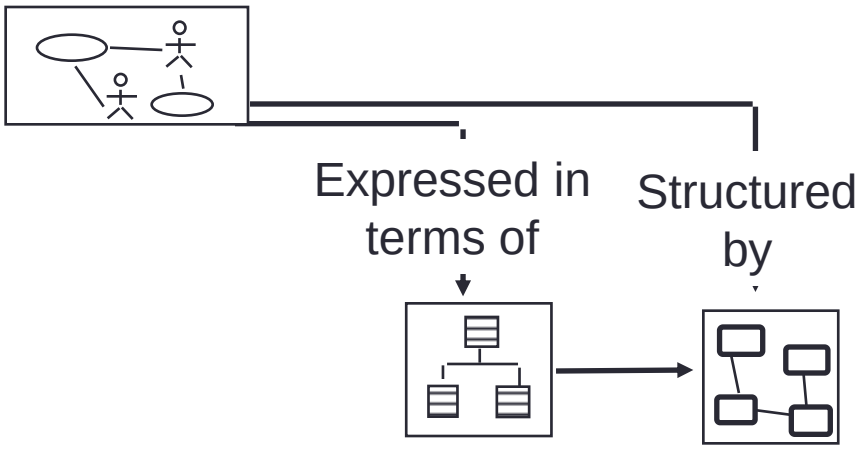


Use Case Model

Application Domain Objects

Software Lifecycle Activities

Requirements Elicitation	Requirements Analysis	System Design	Detailed Design	Implemen-tation	Testing



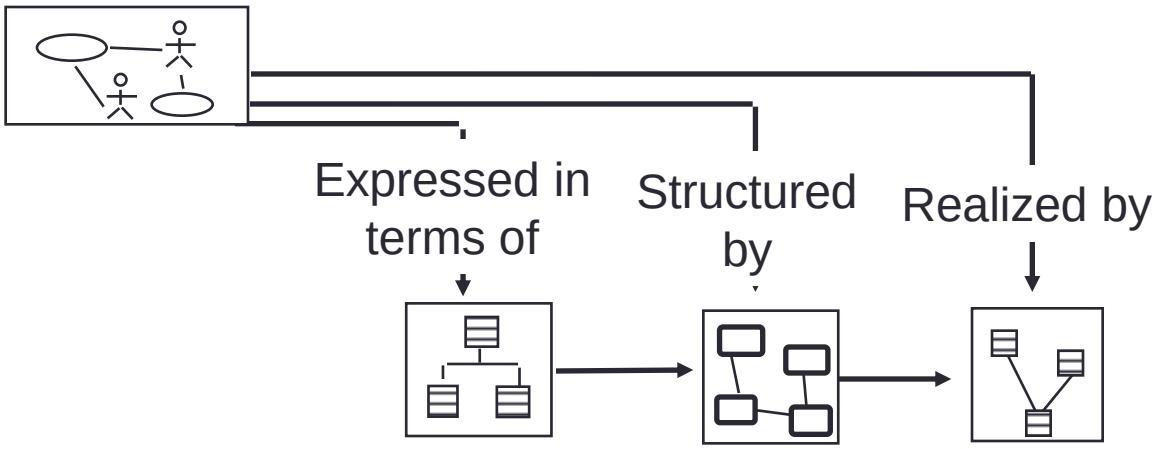
Use Case Model

Application Domain Objects

Sub-systems

Software Lifecycle Activities

Requirements Elicitation	Requirements Analysis	System Design	Detailed Design	Implemen-tation	Testing



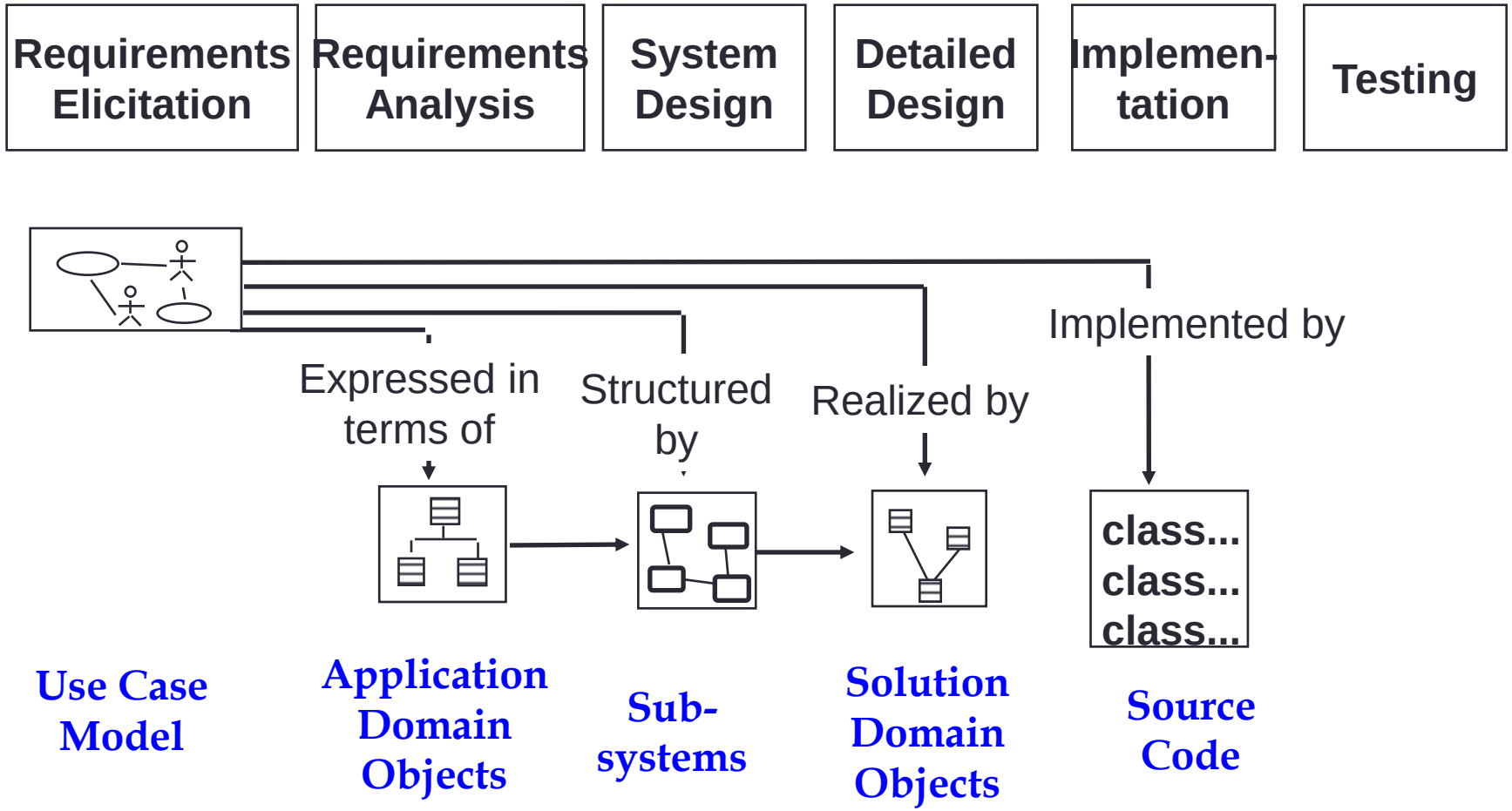
Use Case Model

Application Domain Objects

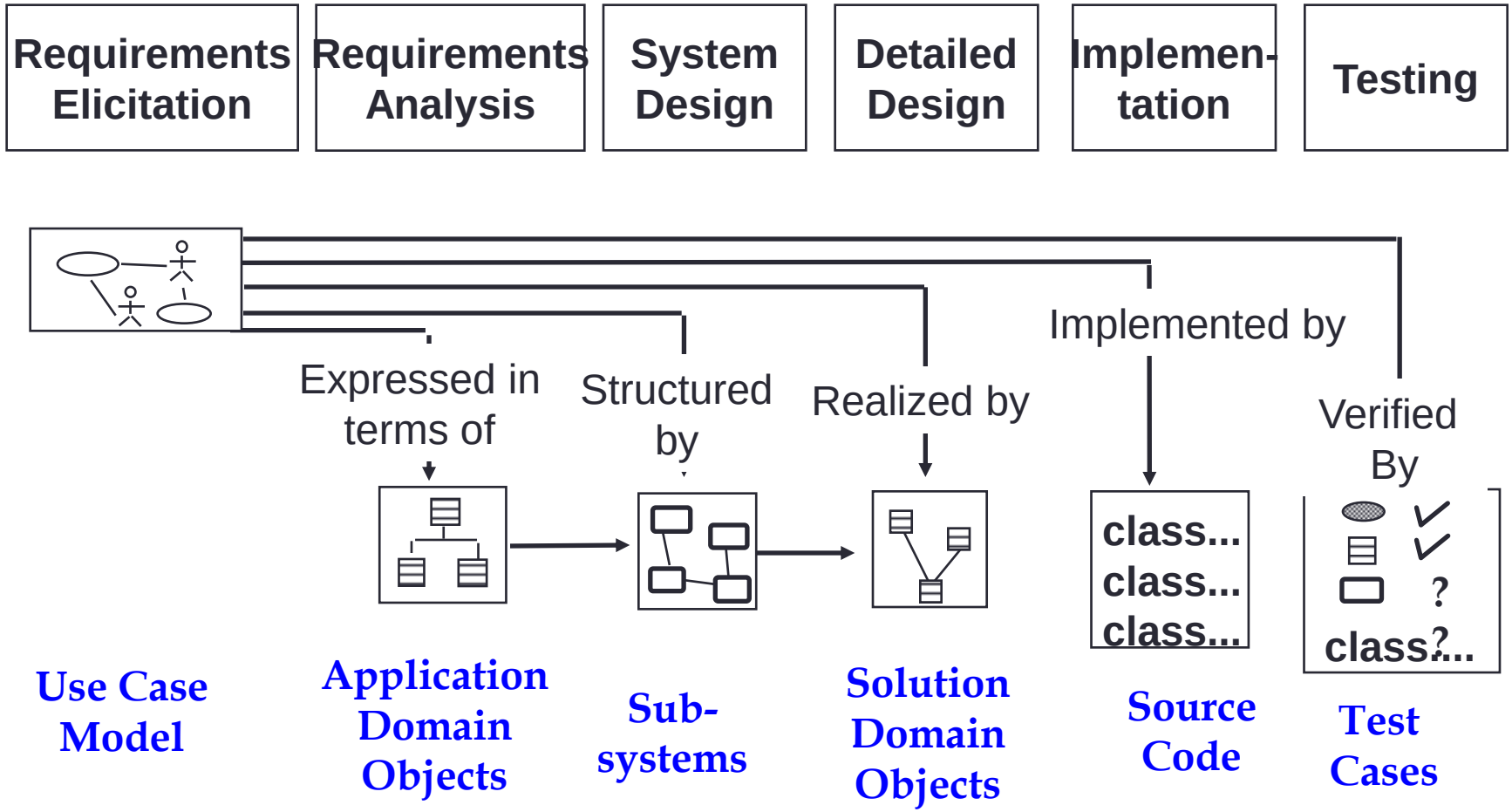
Sub-systems

Solution Domain Objects

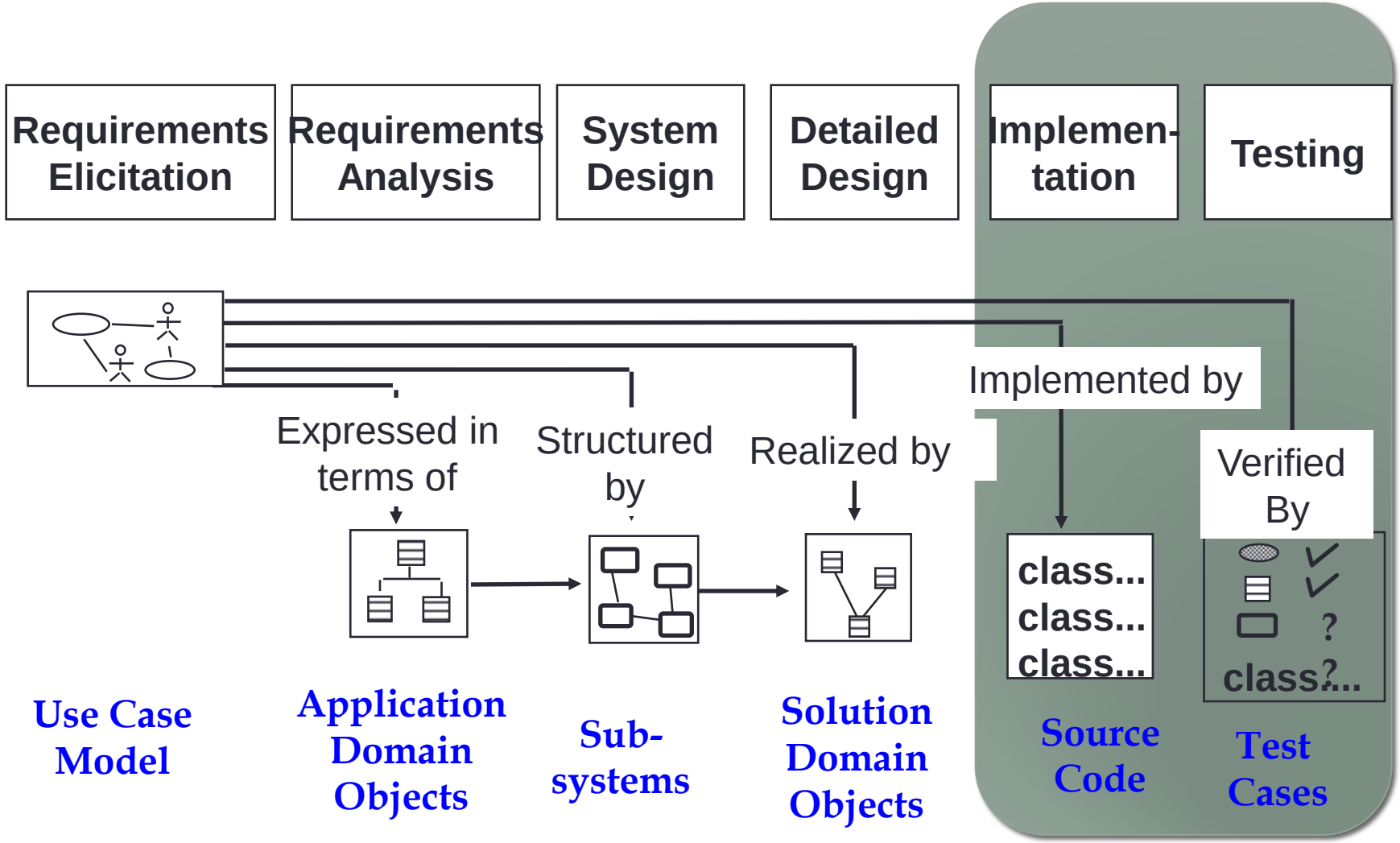
Software Lifecycle Activities



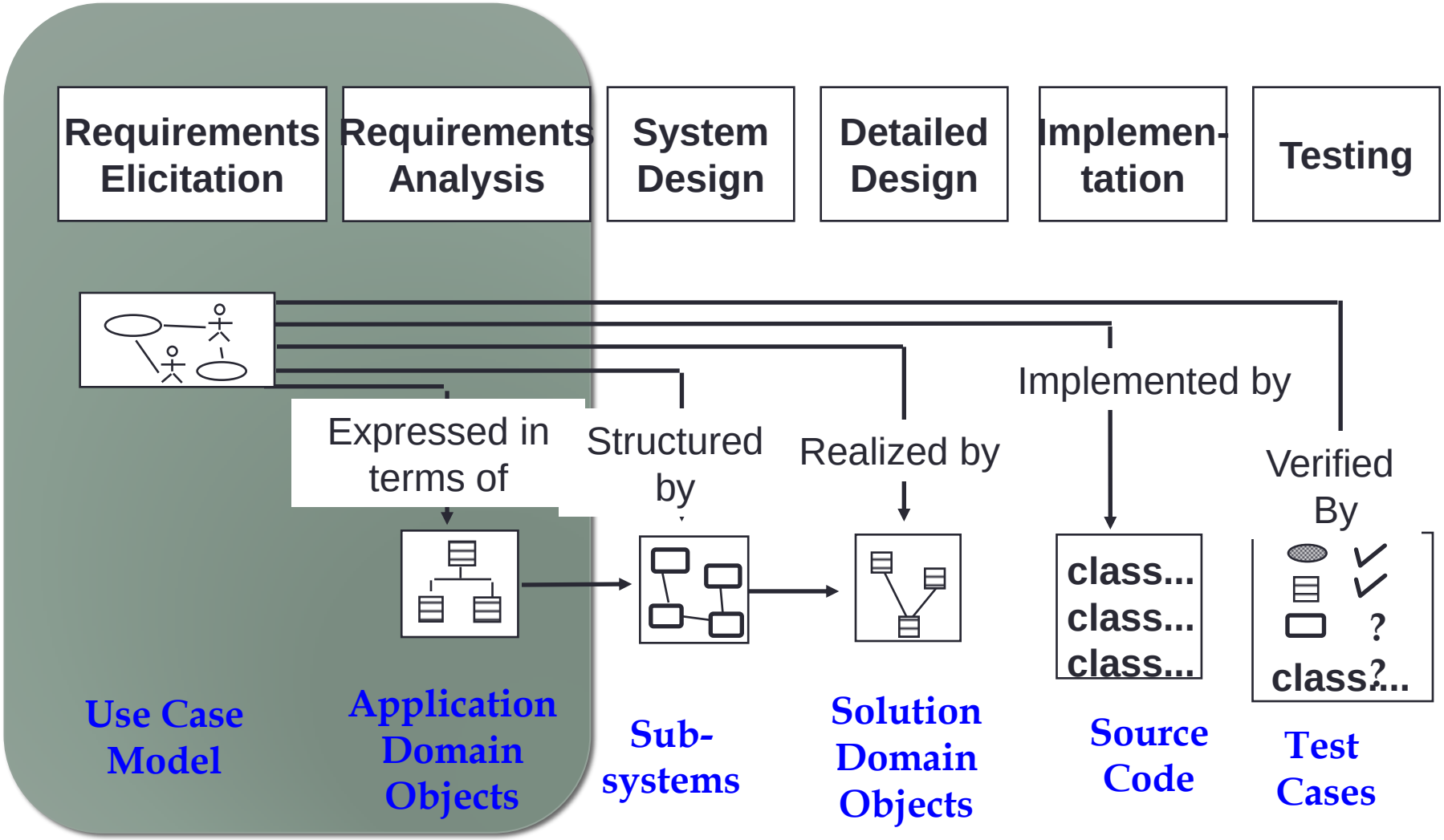
Software Lifecycle Activities



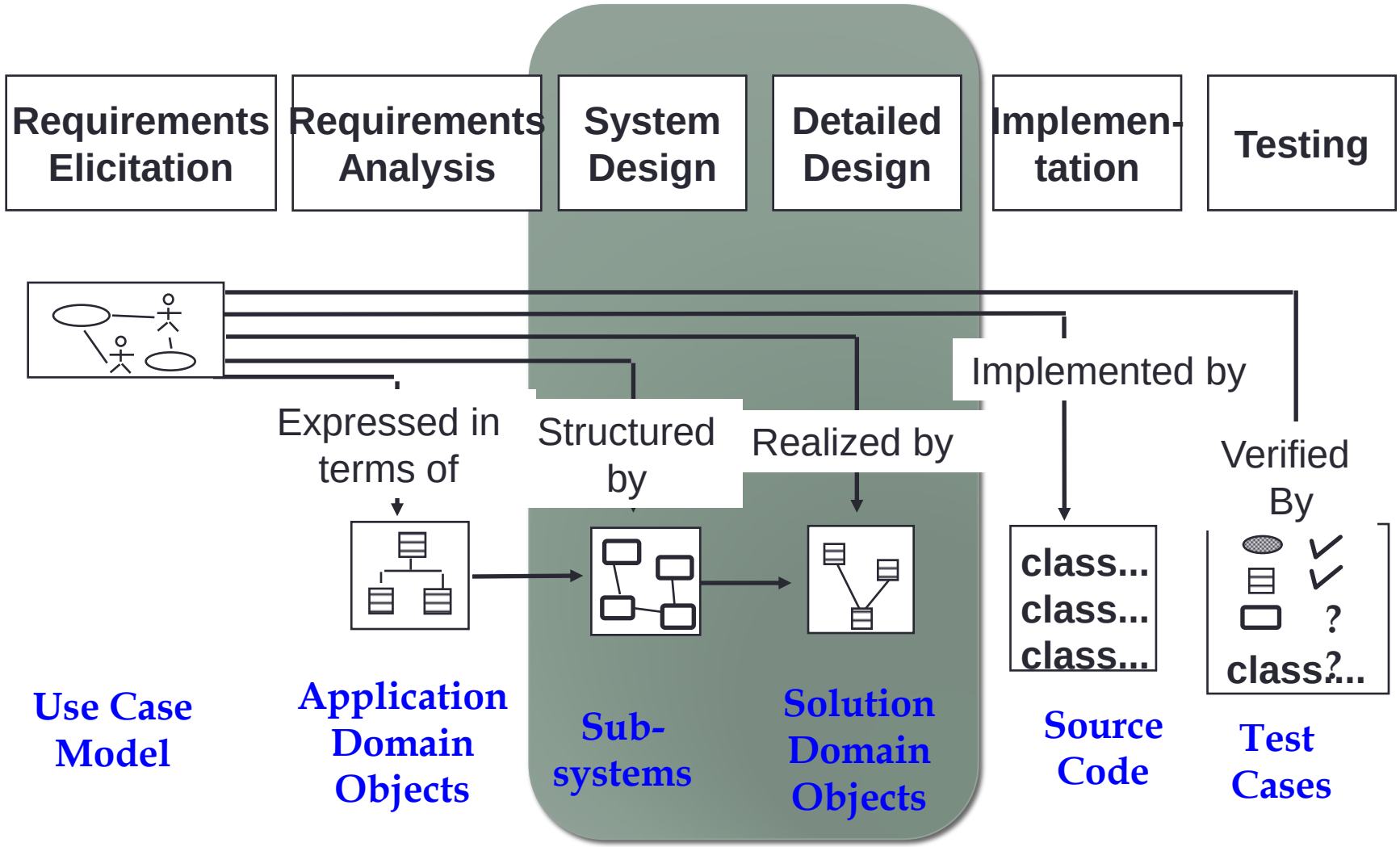
Software Lifecycle Activities



Software Lifecycle Activities



Software Lifecycle Activities



What is Requirements?

It may range from a high-level abstract statement of a service or of a system constraint to a detailed mathematical functional specification.

Requirements Engineering

- The process of establishing the services that
  - The customer requires from a system; and
  - The constraints under which it operates and is developed.

Requirements Engineering Components

Requirements Engineering Components

helps the customer to define what is required:

- what is to be accomplished,
- how the system will fit into the needs of the business, and
- how the system will be used on a day-to-day basis

Requirements Engineering Components

- refining and modifying the gathered requirement
- technical specification

Requirements Engineering Components

- documenting the system requirements using natural language to ensure
  - clarity,
  - consistency, and
  - completeness

How requirements are gathered?

Requirements gathering → Requirements analysis → Requirements specification

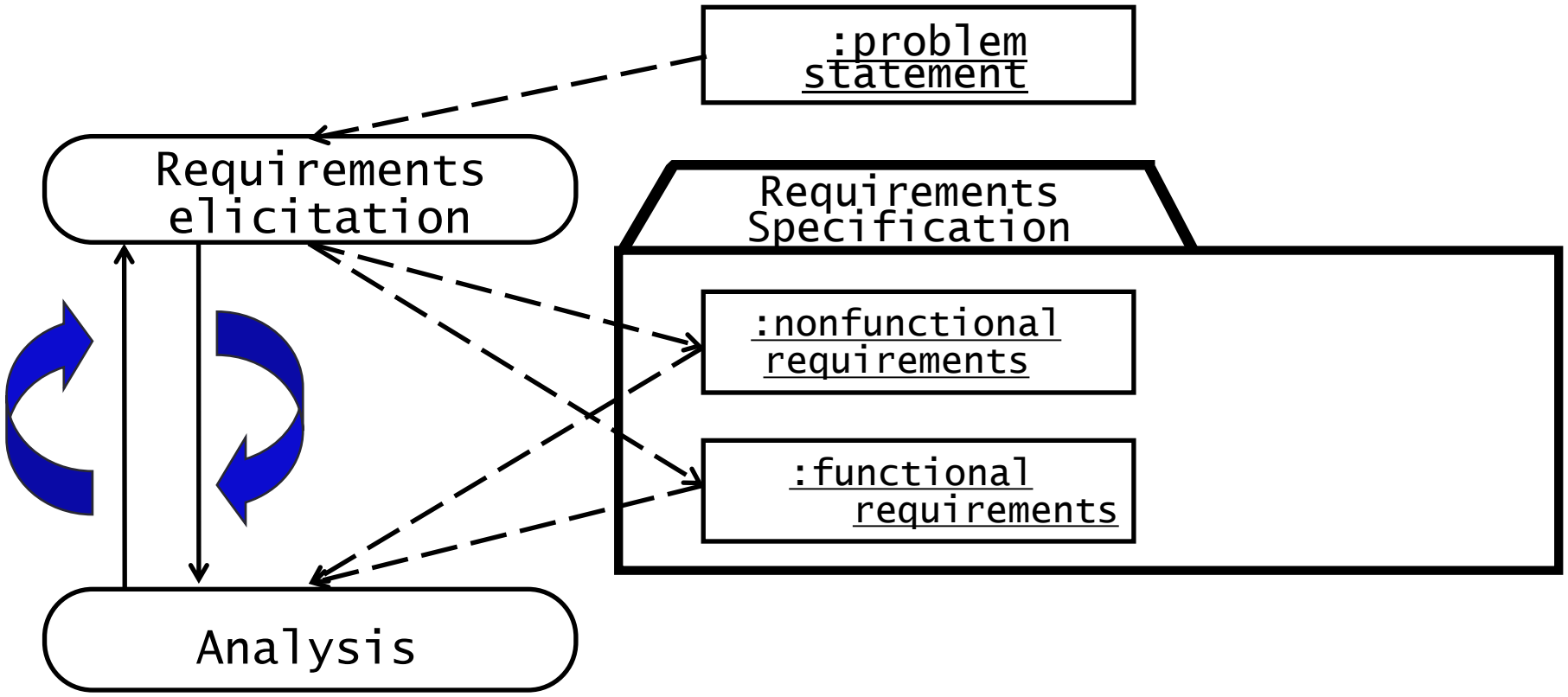
How?

Possible techniques used to gather requirements

- **Questionnaires:** Asking the end user a list of pre-selected questions
- **Task Analysis:** Observing end users in their operational environment
- **Scenarios:** Describe the use of the system as a series of interactions between a concrete end user and the system

How requirements are gathered?

Requirements Process



The analysis model uses a formal or semi-formal notation

Quiz !!

- Requirements elicitation is a cyclic process

a. True
b. False

- Why is Requirements Elicitation a difficult task ?

a. Problem of scope
b. Problem of understanding
c. Problem of volatility
d. All of the mentioned

- Which one of the following is not a step of requirement engineering?

a. Elicitation
b. Design
c. Analysis
d. Documentation

Types of Requirements

- Functional requirements
- Nonfunctional requirements
- Domain requirements

Functional Requirements

- Describe functionality or system services.
- How the system should react to particular inputs and how the system should behave in particular situations.
- Depend on the type of software, expected users and the type of system where the software is used.

Functional Requirements

- A functional requirement for a milk carton
- Ability to contain fluid without leaking

http://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/#conversion-0

Functional Requirements

Software Clinical System

- "If a patient is known to be allergic to a particular medication, then prescription of that medication shall result in a warning message being issued to to the prescriber"

Non-functional Requirements

Can you tell what is it?

Non-functional Requirements

- These define system properties, behaviour and constraints e.g. reliability, response time and storage requirements.
- Constraints are I/O device capability, system representations, etc.
- Non-functional requirements may be more critical than functional requirements.
- If these are not met, the system may be useless.

Non-functional Requirements

- A non-functional requirement for a hard hat
- Must not break under pressure of less than 10,000 PSI

http://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/#conversion-0

Non-functional Requirements

Software Clinical System

- "The system shall be available to all clinics during normal working hours (Mon-Fri, 0830-1730). Downtime during normal working hours shall not exceed 5 seconds in any one day"

Non-functional classifications

- Product requirements
  - Requirements which specify that the delivered product must behave in a particular way e.g. execution speed, reliability, etc.
- Organisational requirements
  - Requirements which are a consequence of organisational policies and procedures e.g. Registered users of the clinical system shall authenticate themselves using their health authority identity card.
- External requirements
  - Requirements which arise from factors which are external to the system and its development process e.g. interoperability requirements, legislative requirements, etc.

Metrics for specifying nonfunctional requirements

Property	Measure
Speed	Processed transactions/second User/event response time Screen refresh time
Size	Mbytes Number of ROM chips
Ease of use	Training time Number of help frames
Reliability	Mean time to failure Probability of unavailability Rate of failure occurrence Availability
Robustness	Time to restart after failure Percentage of events causing failure Probability of data corruption on failure
Portability	Percentage of target dependent statements Number of target systems

Domain requirements

- The system's operational domain imposes requirements on the system.
  - For example, a train control system has to take into account the braking characteristics in different weather conditions.
- Domain requirements be new functional requirements, constraints on existing requirements or define specific computations.
- If domain requirements are not satisfied, the system may be unworkable.

Example

- A library system that provides a single interface to a number of databases of articles in different libraries.
- Users can search for, download and print these articles for personal study.

- **What are the functional and non-functional requirements?**

Examples of functional requirements

- The user shall be able to search either all of the initial set of databases or select a subset from it.
- The system shall provide appropriate viewers for the user to read documents in the document store.
- Every order shall be allocated a unique identifier (ORDER_ID) which the user shall be able to copy to the account's permanent storage area.

Examples of Non-functional requirements

- Product requirement
  - The system shall be available to all users during normal working hours of the library.
  - Search speed should be fast enough,
  - The user interface for LIBSYS shall be implemented as simple HTML without frames or Java applets.
- Organisational requirement
  - The system development process and deliverable documents shall conform to the process and deliverables defined in XYZCo-SP-STAN-95.
- External requirement
  - The system shall not disclose any personal information about customers apart from their name and reference number.

How requirements are gathered?

Requirements Specification Template

1. Introduction
2. Current system
3. Proposed system
   - 3.1 Overview
   - 3.2 Functional requirements
   - 3.3 Nonfunctional requirements
   - 3.4 Constraints (“Pseudo requirements”)
   - 3.5 System models
     - 3.5.1 Scenarios
     - 3.5.2 Use case model
     - 3.5.3 Object model
       - 3.5.3.1 Data dictionary
       - 3.5.3.2 Class diagrams
     - 3.5.4 Dynamic models
     - 3.5.5 User interfae
4. Glossary

Requirements Validation

What if the requirements are wrong?

Requirements Validation

Requirements validation is a quality assurance step, usually performed after requirements elicitation or after analysis

- Correctness:
  - The requirements represent the client's view
- Completeness:
  - All possible scenarios, in which the system can be used, are described
- Consistency:
  - There are no requirements that contradict each other.

Requirements Validation (2)

- Clarity:
  - Requirements can only be interpreted in one way
- Realism:
  - Requirements can be implemented and delivered
- Traceability:
  - Each system behavior can be traced to a set of functional requirements

References

- A number of slides in this talk is based on:
  - Object-Oriented Software Engineering Using UML, Patterns, and Java Chapter 4, Requirements Elicitation
    Bernd Bruegge & Allen H. Dutoit (Lecture Notes)
  - SOFTWARE ENGINEERING 9 Ed. by Ian Sommerville (lecture Notes)