


# **SOEN 341**

## **Software Process**



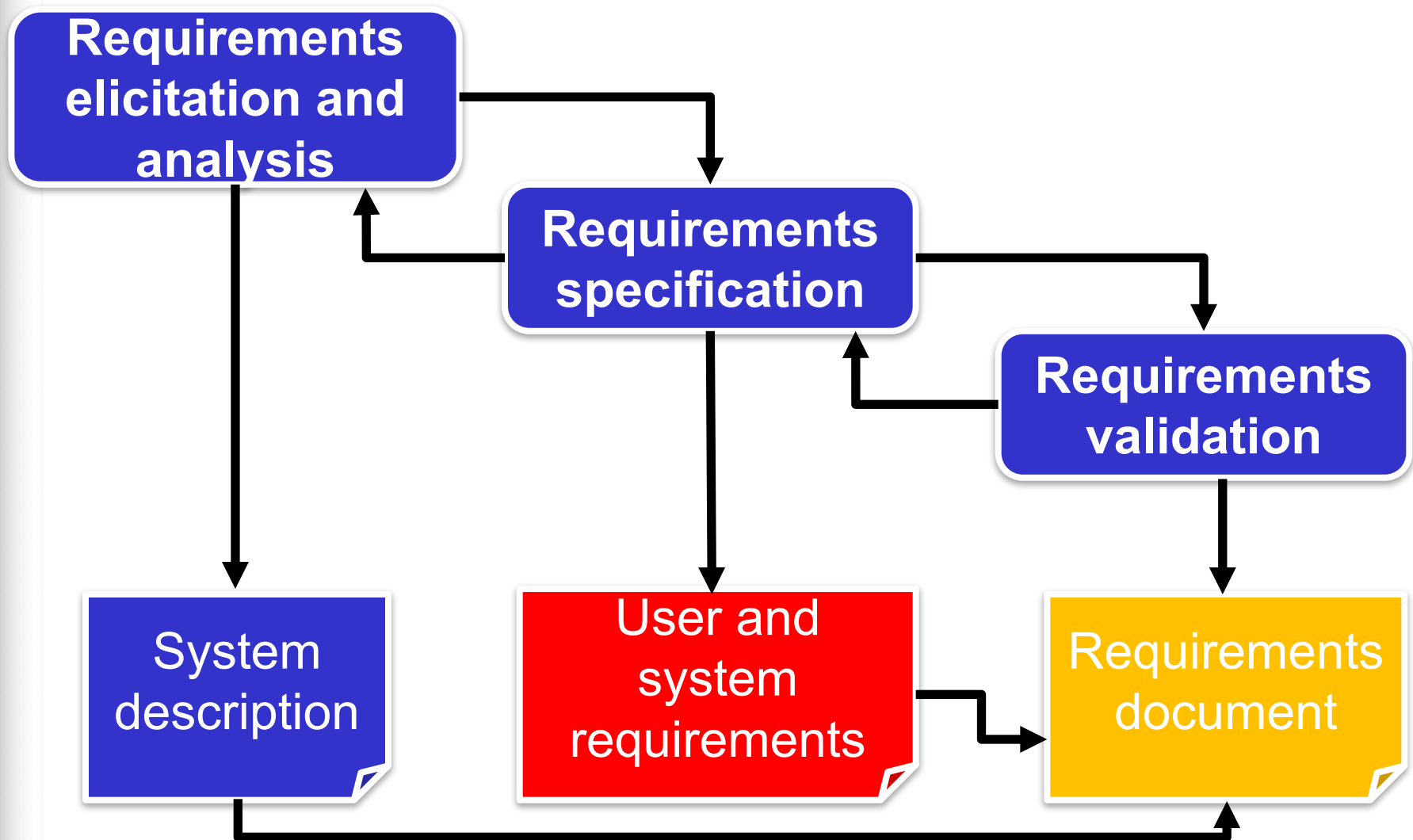
**Lecture 07:**  
**Software Requirements**  
**Emad Shihab, PhD**

# 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.

- What should be implemented?
- How the system will behave?
- Not ***how*** the product will be constructed

# Requirements/Specification



# Software Requirements

- Involve the **elicitation**, **analysis**, **specification** and **validation** of the different stakeholders
- An analyst/requirement engineer:
  - Elicits these demands/needs (raw requirements)
  - Analyzes them for consistency, feasibility, and completeness
  - Formulates them as requirements and write down a specification
  - Validates that the gathered requirements reflect the needs/demands of stakeholders

# Requirements

## Requirements are either:

- A condition or capability needed by a user to solve a problem or achieve an objective.
- A condition or capability that must be met or possessed by a system [...] to satisfy a contract, standard, specification, or other formally imposed document.

# User vs. System

## **User requirements:**

- Describe the services the system is expected to provide

## **System requirements:**

- Detailed descriptions of system functions and operational constraints

# Example of a user requirement

The system shall generate monthly management reports showing the cost of drugs prescribed by each clinic

# Example System Requirement

- On the last working day of each month, a summary of the drugs prescribed, their cost and the prescribing clinics shall be generated.
- The system shall automatically generate the report for printing after 17.30 on the last working day of the month.
- A report shall be created for each ... 8



# Question...

Difference between system and user requirements?



# **Functional vs. Non- functional Requirements**

# Types of Requirements

## Functional requirements:

- Describe what the system should do (from a user's or system's perspective)
- Typical form: “***The system shall/should do ...***”

# Example Functional Requirements

- A user shall be able to search the appointments lists for all clinics.
- The system shall generate each day, for each clinic, a list of patients who are expected to attend appointments that day.
- Each staff member shall be uniquely identified by his or her 8-digit employee number.

# Imprecision in Requirements

- **Ambiguous requirements** may be interpreted in different ways by developers and users.
- Consider the term 'search' in a requirement
  - **User intention** – search for a patient name across all appointments in all clinics;
  - **Developer interpretation** – search for a patient name in an individual clinic. User chooses clinic then search.

# Main concern

- Development teams have a knack for addressing **ambiguous requirements** with the **simplest solution to implement**
- Imprecision may lead to a solution that does not match user expectations

# Main concern

## Completeness:

- Should describe **all of the system features** that are required

## Consistency:

- **Conflicts or contradictions** among requirements should be avoided

# Non-functional Requirements

## Non-functional requirements:

- **Specify properties and constraints of the system**
  - e.g. reliability, response time and storage requirements. Constraints are I/O device capability, system representations, etc.
- Typical form: “***The system shall/should be ...***”
- Although often overlooked, NFRs may be critical to the success/failure of a system



# NFR Metrics

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

# Example of NFR

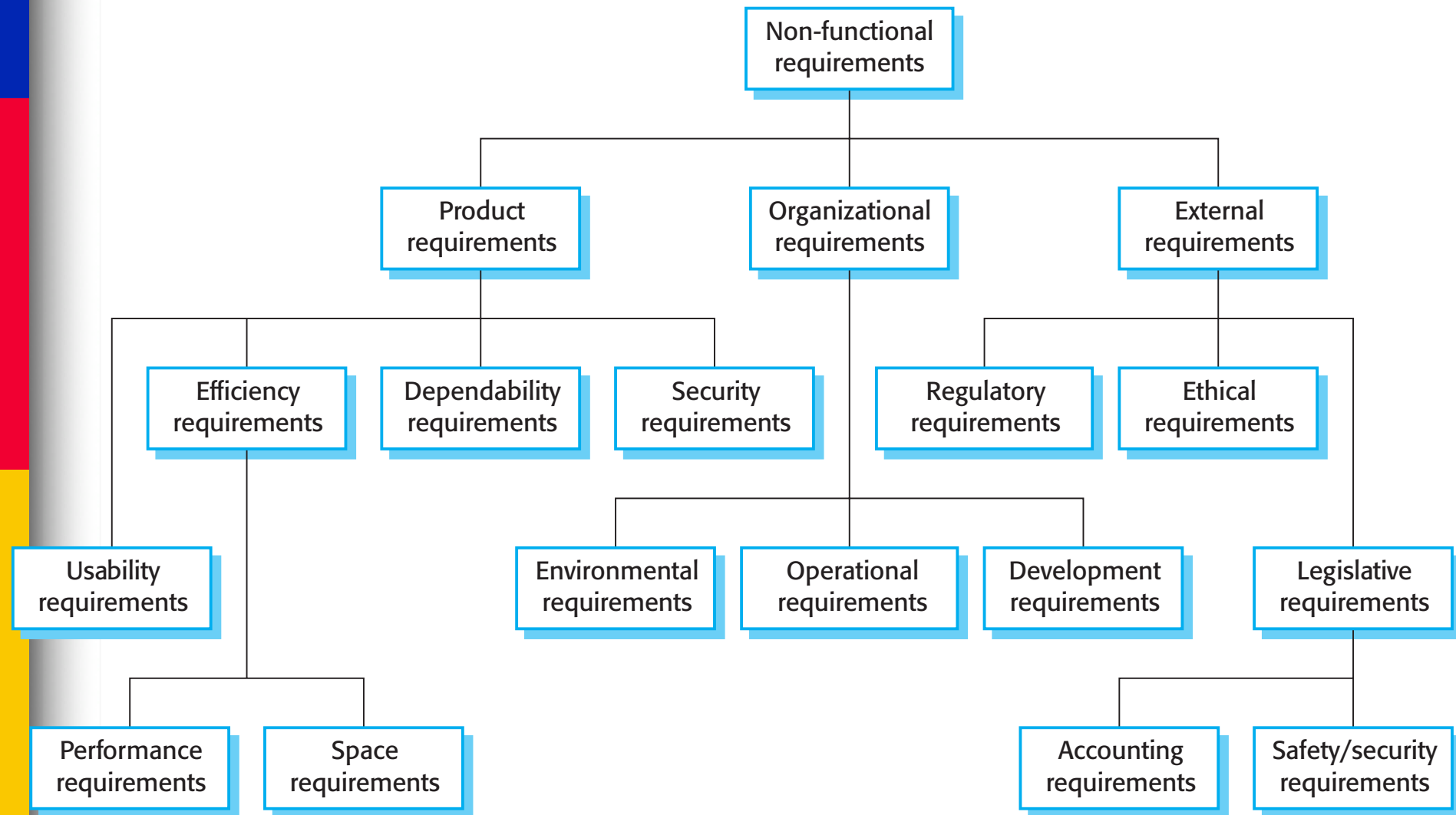
- The Mentcare system shall be available to all clinics during normal working hours (Mon–Fri, 0830–17.30).
- Downtime within normal working hours shall not exceed five seconds in any one day.

# Example NFRs

The system should be **easy** to use by medical staff and should be organized in such a way that user errors are **minimized**.

Medical staff shall be able to use all the system functions after **four** hours of training. After this training, the average number of errors made by experienced users shall **not exceed two per hour** of system use.

# Classification of NFR



# Importance of NFR

Typically, functional reqs must be met (ie. mandatory)

NFRs could be:

- Mandatory: eg. response time for a valve to close

  - The system is unusable

- Not mandatory: eg. response time for a UI

  - The system is usable but provides a non-optimal experience

NFRs are very important: **20% of the requirements, hardest to elicit and specify**

NFR: importance increases as market matures

# Expressing NFRs

- Functional are usually expressed in Use-Case form
- NFR cannot be expressed in Use-Case form
  - usually do not exhibit externally visible functional behaviour

# The effects of NFRs on high level design and code

- Their implementation does not map to a particular subsystem
- Very hard to **modify a NFR** since such modifications **require major changes**:
  - Consider making an already implemented system more secure, more reliable, etc.

# Question...

Difference between functional and non-functional requirements?



# **Good vs. Bad Requirements**

# Anatomy of a bad requirement

The user quickly sees their current account balance on their laptop screen

# Anatomy of a good requirement

The online banking system shall allow the user to access their current account balance in less than 5 seconds

# Characteristics of good requirements

## Complete sentences!

- Not a list of buzzwords and acronyms

## Use plain language!

- “The system shall utilize an alphabetical schematic to organize records” **NO!**
- “The system shall organize records alphabetically” **YES!**

# Characteristics of good requirements

## Subject and predicate:

- **Subject** = system being discussed
- **Predicate** = a condition, action, or intended result
- **Verb in predicate:**
  - “Shall” = mandatory
  - “May” = optional

# Characteristics of Good Requirements

- **Realism (Feasibility):**
  - Avoid wishful thinking
- **Validity (Necessity):**
  - Provides the details of the desired end goal
- **Verifiability (Testability):**
  - Contains quantifiable success criteria
- **Consistency:**
  - No conflicts
- **Completeness:**
  - All functions

# Pitfalls to Avoid

## What vs. how:

- Never describe how the system should deliver the requirement
- Focus on what the system should do

The system shall use Microsoft Outlook to send a confirmation email to the customer.`

The system shall inform the customer of an order confirmation within 2 minutes.

# Pitfalls to Avoid

## **Escape clauses:**

- Danger signs: “if”, “but”, “when”, “except”, “unless”, “although”

## **Ambiguity:**

- Danger signs: “or”, “etc”, “and so on”

## **Vague, undefinable terms:**

- Danger signs: “user-friendly”, “highly versatile”, “flexible”, “approximately”, “as much as possible”



The system shall  
be easy-to-use and  
require minimal  
training except for  
professional mode.

# Pitfalls to avoid

## Conjunctions:

- One requirement per sentence
- Danger signs: “and”, “or”, “with”, “also”

## Speculation:

- Danger signs: “usually”, “generally”, “often”, “normally”, “typically”

## Wishful thinking:

- Danger signs: “100% reliable”, “bug-free”, “handle all failures”, “run on all platforms”

The system shall  
be fully adaptable  
to all situations and  
often require no  
reconfiguration by  
the user.

# Quiz

What is the main difference between system and user requirements?

