COMP433: Software Engineering

Requirements Engineering (/elicitations & Discovery!)

(Chapter 4: Sommerville; Chapter 4: Bruegge)

Prof. Adel Taweel
ataweel@birzeit.edu

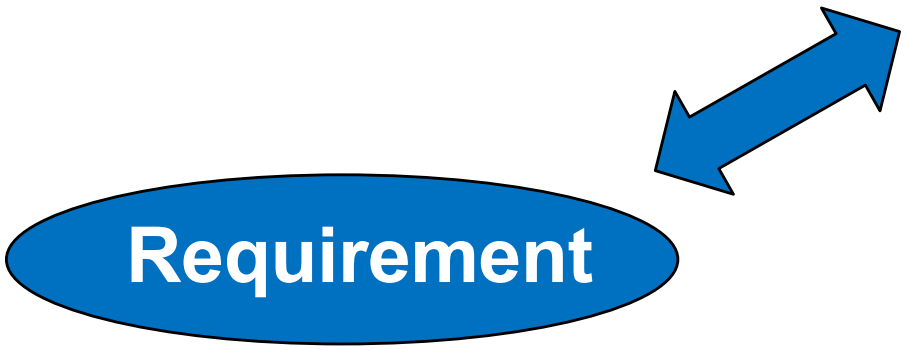
Chapter objectives:

- To introduce the concepts of **user, domain and system requirements**
- To describe **functional / non-functional requirements**
- To explain **techniques** for describing system requirements
- To explain **how software requirements may be organised** in a requirements document
- To introduce some methods for **requirements discovery**

Requirements engineering

Requirements engineering is the process of establishing

- the services (or functionalities) that the customer requires from a system
- the constraints under which it operates and is developed



The descriptions of the system services and constraints

that are generated during the requirements engineering process

What is a requirement?

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

This is inevitable as requirements may serve a **dual function**

May be the basis for a bid for a contract - therefore must be open to interpretation

May be the basis for the contract itself - therefore must be defined in detail

Both these statements may be called requirements

Types of requirements

User requirements

Statements in natural language [plus diagrams] of the services the system provides and its operational constraints. Written for **customers**

System requirements

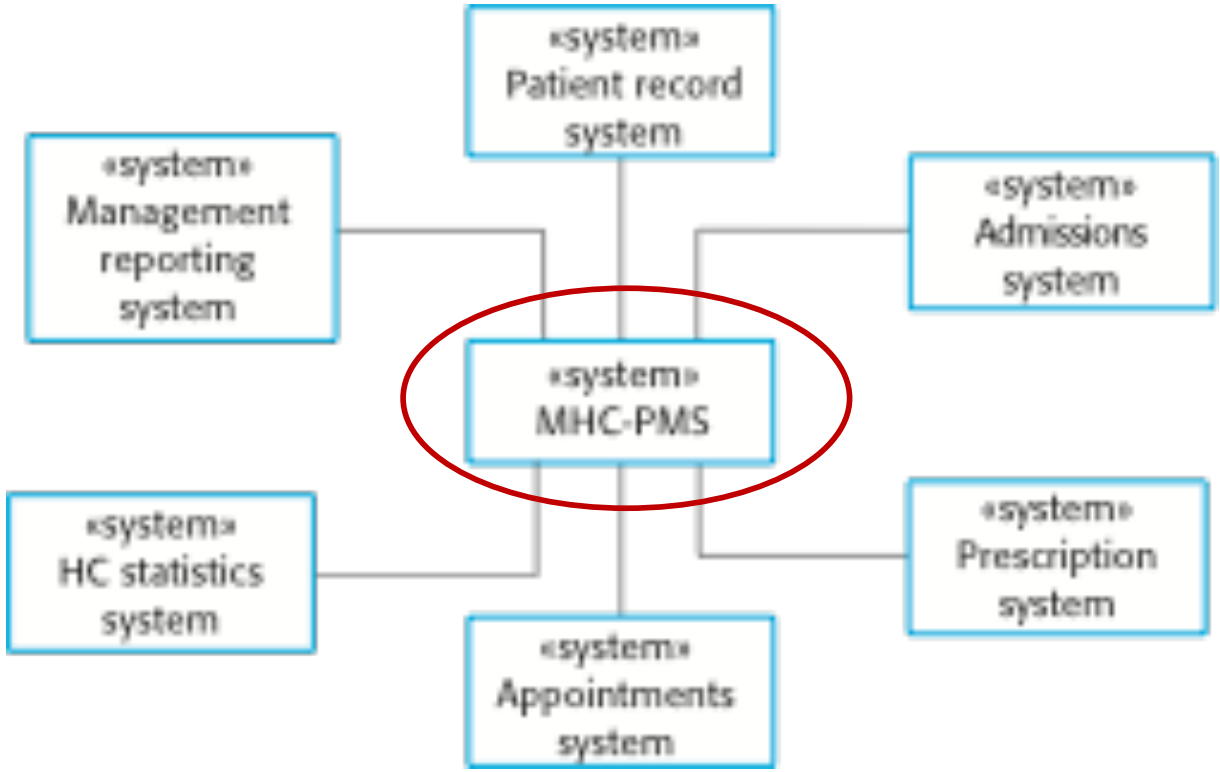
A structured document setting out detailed descriptions of the system services. Written for **customers**, and/or as a contract between **client (or customer)** and **contractor**

Software specification

A detailed software description that can serve as a basis for a design or implementation. Written for **developers**

MHC-PMS: Context - Example

MHC-PMS: Patient Medical System



User requirements

Describe actions or abilities that need to be provided by the system to its users to perform or undertake to serve their business needs or services.

Should describe (functional and non-functional) requirements, so that they are understandable by system users who do not have detailed technical knowledge

User requirements are defined using natural language, **tables** and **diagrams** (will be discussed later)

User requirement definition

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

User Requirements Example

UR1.0: The PMS system shall generate monthly management reports, showing the cost of drugs prescribed by each clinic and doctor during that month.

UR2.0: The PMS system should be able to generate monthly management reports, showing details of admitted and discharged patients in each clinic during that month.

UR3.0: The PMS system should be able to generate monthly management reports, showing details of deceased patients in each clinic during that month.

UR4.0: The system administrator shall be able to register and create user accounts, for new users, as member of staff, of the PMS system.

UR5.0: The user, as a member of staff, shall be able to login into the PMS system, using a preregistered username and a password.

UR6.0: The PMS system shall enable authorised users, receptionist and Nurse, to register a patient and create a patient medical record.

UR7.0: The PMS system shall enable authorized users, Receptionist and Nurse, to activate and deactivate a patient medical record

UR8.0: Authorised users (Receptionist, Manager, Doctor and Nurse) shall be able be view patient personal information.

UR9.0: Authorised Users, Doctor and Nurse, shall be able to view and edit patient medical Record.

UR10.0: The PMS system shall be able to generate clinical reports, showing clinical information of patients of active cases under treatment.

UR11.0: ….

System requirements

Describe specifications detailing specific needs of **each User Requirement**, to support business services

– **More detailed specifications of user requirements**

Serve as a basis for designing the system

May be used as part of the system contract

System requirements may be expressed using, natural language and system models (will be discussed in later lectures)

User and system requirements

User requirement definition

1. The MHC-PMS 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 automatically 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, 20 mg, etc.) separate reports shall be created for each dose unit.
1.5 Access to all cost reports shall be restricted to authorized users listed on a management access control list.

User & System Requirements Example

UR4.0: The system administrator shall be able to register and create user accounts, for new users of the system.

SR4.1: The system shall enable the system administrator to register and create an account for the user, as a member of staff. Created account shall include user's personal information, including Full Name and Address.

SR4.2: The system shall ensure and enable user to specify a valid email address. The system shall only include a validated email address. The system should prevent the system administrator from registering or creating user accounts without a valid email address.

SR4.3: The system administrator shall be able to specify a username, made up of a minimum of 8 characters, that include alphanumeric only.

SR4.4: the system shall enable the system administrator to specify a password, of a minimum of 8 characters. The password must include minimum three types of characters, of alphabets, numbers, special characters

SR4.5: the system shall validate that usernames are uniquely identified in the system.

SR4.6: the system shall validate that the password are stored encrypted in a non-reversable form, using MD5 hash algorithm.

SR4.7: The system should enable users to reset their passwords. The system shall send Reset passwords to the registered validated email address.

UR5.0: The user, as a member of staff, shall be able to login into the system, using a preregistered username and a password.

SR5.1: The system shall enable the user, as a member of staff, to write their preregistered username and password, to login to access their functions as per their pre-defined Role-based privileges.

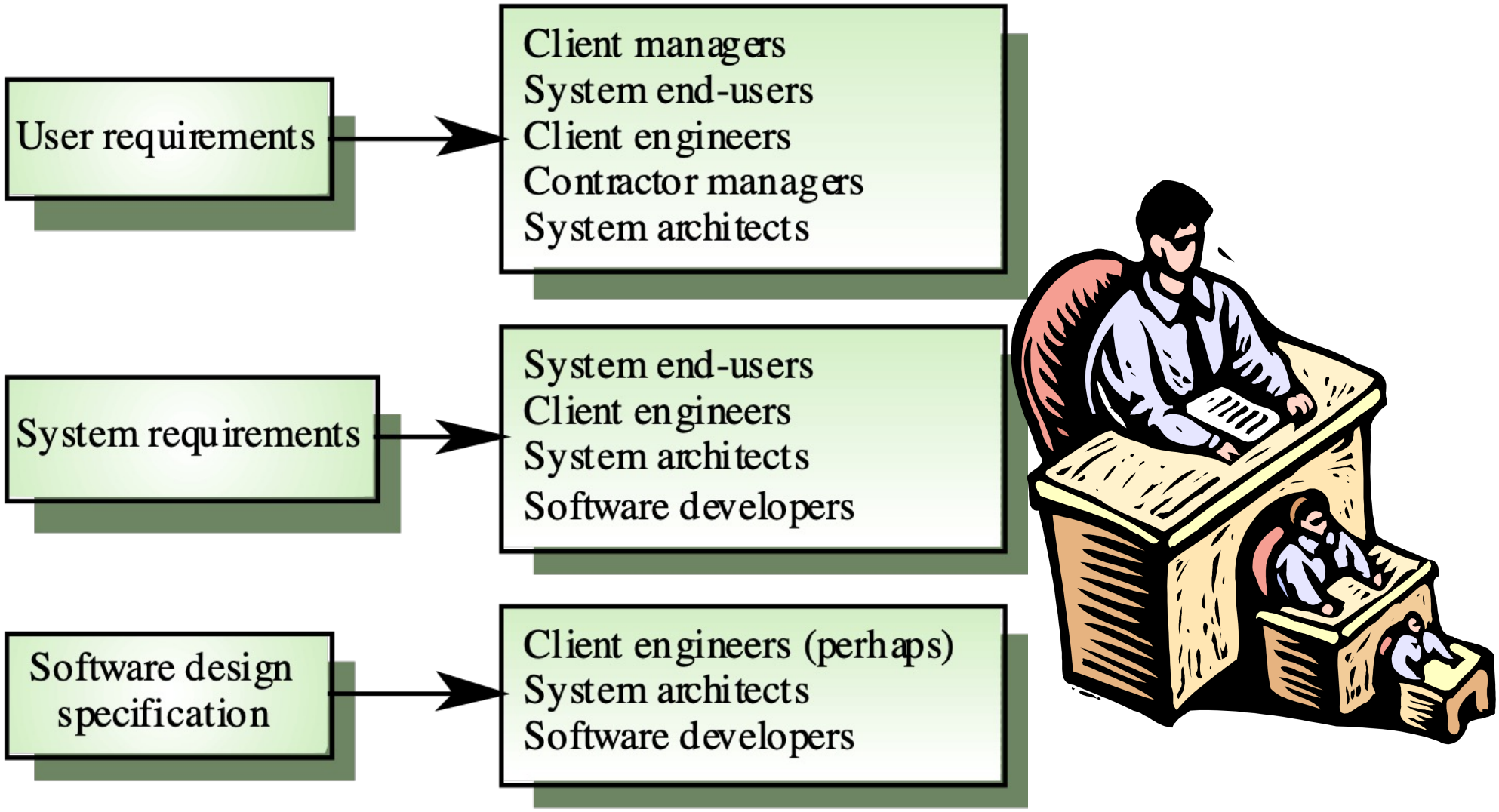
SR5.2: The user shall be to reset their password. The system shall send an email to the user validated email address containing a URL link to reset/write a new password. The new password shall conform to SR2.4.

SR5.3: …. etc

UR6.0: The PMS system shall be enable authorised users, receptionist and Nurse, to register a patient and create a patient medical record.

SR6.1: …. etc

Requirements readers



Functional and non-functional 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.

Non-functional requirements

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

Domain requirements

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

Domain requirements can result into additional functional and non-functional requirements

Functional Requirements

Describe functionality or system services

Depend on the type of software, expected users and the type of system where the software is used

Functional user requirements may be high-level statements of what the system should do **BUT** **functional system requirements** should describe the system services in detail

Functional requirements: Examples

2.3: A user shall be able to search the appointments lists for all clinics.

...

5.2: The PMS system shall generate each day, for each clinic, a list of patients who are expected to attend appointments that day.

...

6.1: Each staff member using the system shall be uniquely identified by an 8-digit employee number.

Non-functional requirements

Define system properties and constraints e.g. reliability, response time and storage requirements. Constraints are I/O device capability, system representations, etc.

Process requirements may also be specified mandating a particular development environment (IDE), programming language or development method

Non-functional requirements may be more critical than functional requirements. If these are not met, the system may be useless!

Non-functional classifications

Product requirements

Requirements which specify that the **delivered product must behave in a particular way** e.g. execution speed, reliability, etc.

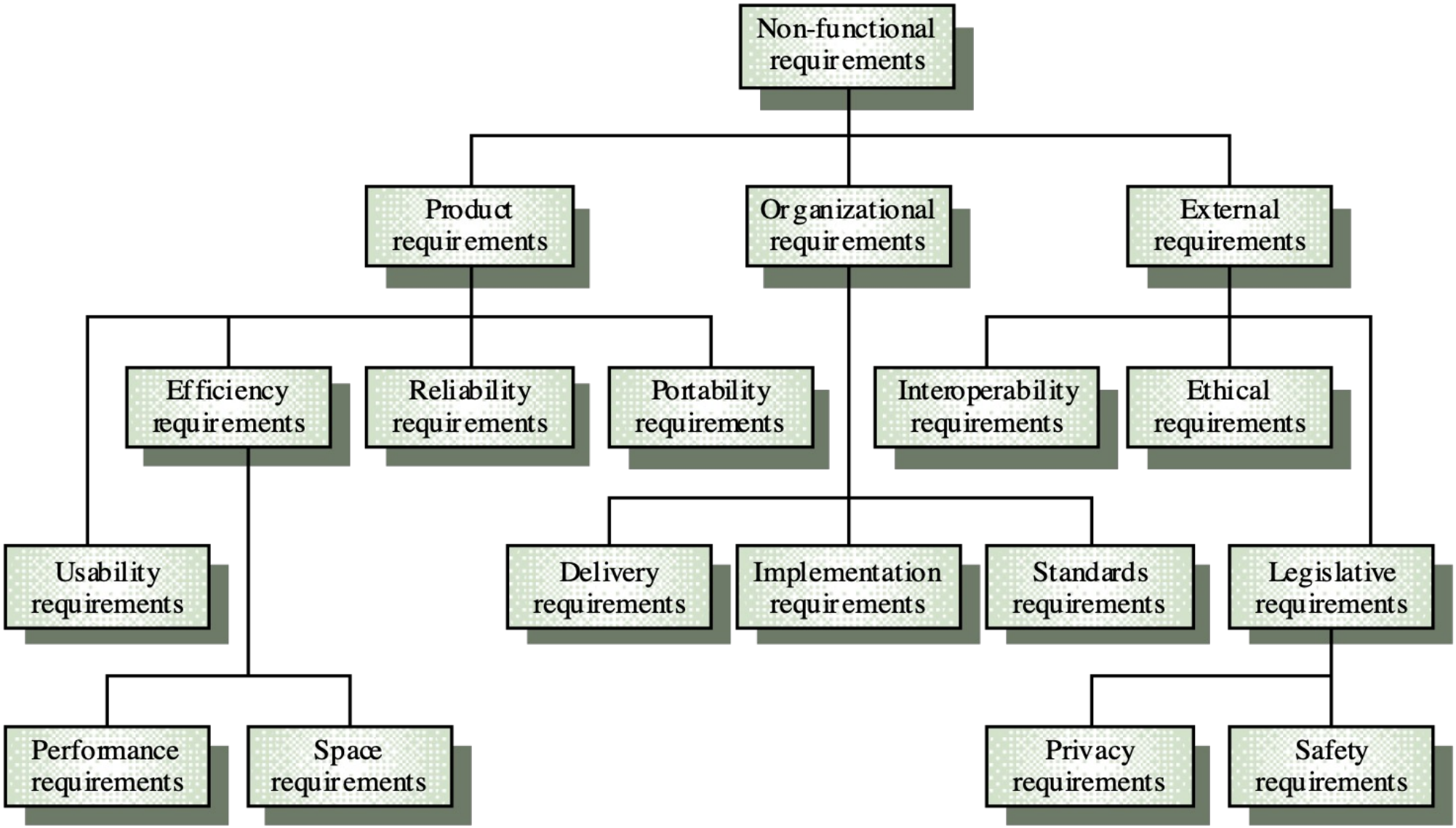
Organisational requirements

Requirements which are a **consequence of organisational policies and procedures** e.g. process standards used, implementation requirements, etc.

External requirements

Requirements which arise from factors which are **external to the system and its development process** e.g. interoperability requirements, legislative requirements, etc.

Non-functional requirement types



Non-functional requirements: Examples

- **Product requirement**
  3.C.8.The MHC-PMS 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.

- **Organizational requirement**
  5.4.3 Users of the MHC-PMS system shall authenticate themselves using their health authority identity card.

- **External requirement**
  7.2.3 The system shall implement patient privacy provisions as set out in the regulation HStan-03-2006-priv.

Goals and requirements

Non-functional requirements may be very difficult to state precisely and imprecise requirements may be difficult to verify.

Goal

A general intention of the user such as ease of use

Verifiable non-functional requirement

A statement using some measure that can be objectively tested

Goals are helpful to developers as they convey the intentions of the system users

Example: Usability requirements

- **A system goal**
  G.8.1 The PMS system should be easy to use by medical staff and should be organized in such a way that user errors are minimized. (**Goal**)

- **A verifiable non-functional requirement**
  8.4.3 Medical staff shall be able to use all the PMS 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. (**Testable non-functional requirement**)

Non-functional requirements Metrics

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

Derived from the application domain and **describe system characteristics and features** that reflect the domain

- Example1: a train control system has to take into account the braking characteristics in different weather conditions.
- Example2: a PMS has to enforce all confidentiality rules in accordance with national medical domain practices

May generate new functional requirements, and new constraints on existing requirements or define specific computations

If domain requirements are not satisfied, the system may be unworkable

Domain requirements problems

Understandability

Requirements are expressed in the language of the application domain

This is often not understood by software engineers developing the system

Implicitness

Domain specialists understand the area so well that they do not think of making the domain requirements explicit