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.

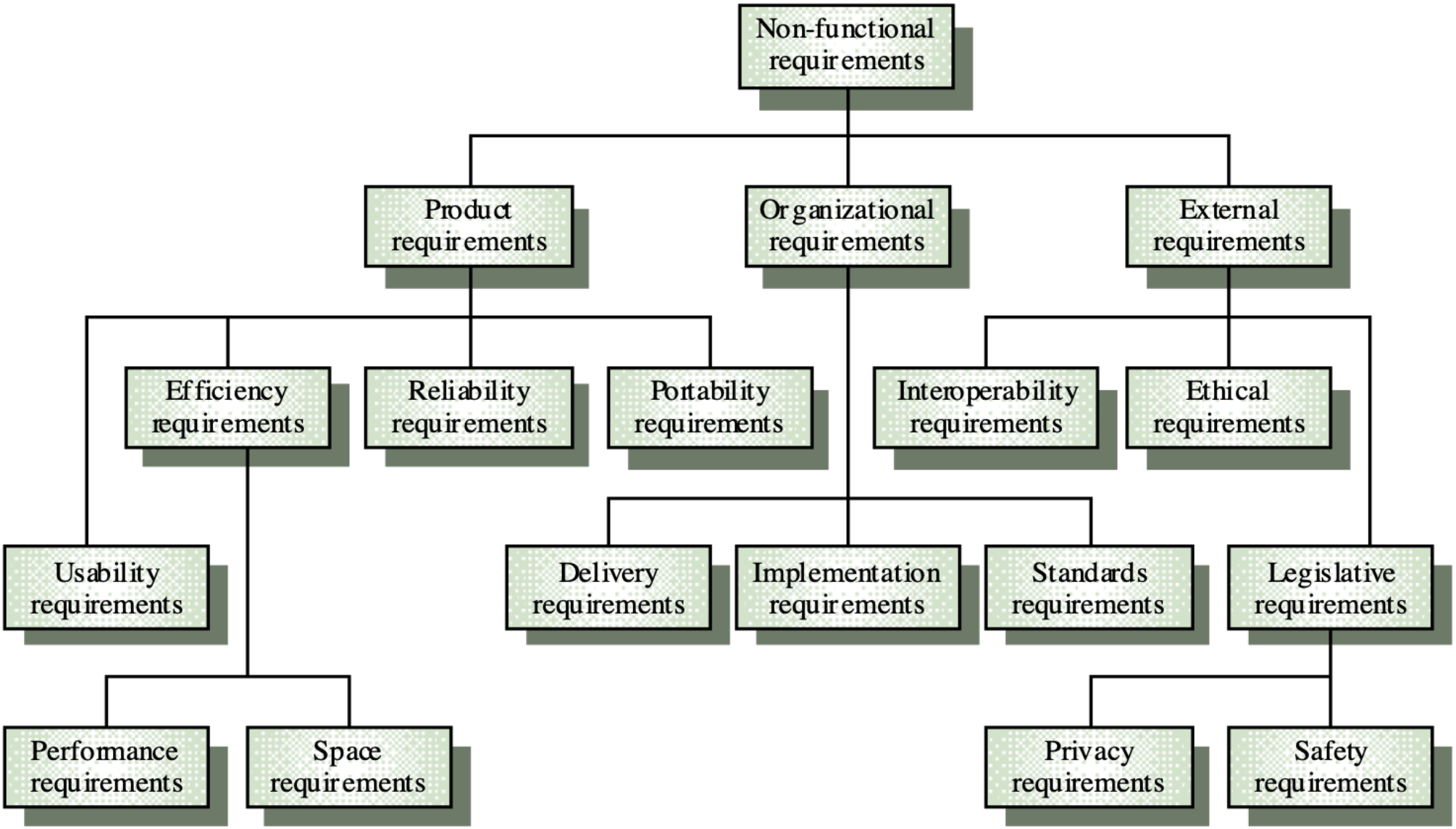
Organisational requirements

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

Requirements Characteristics

Requirements Correctness

A requirement is correct when it is part of the actual needs of the system.

Problems arise when requirements are implied or derived and become beyond the scope of the actual needs of system

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

Consider the term 'search' in requirement 2.3

User requirement– search for a patient name across all appointments, of all patients in all clinics;

Implied requirements– search for a specific patient on their name, Date of Birth, Address in a clinic.

Requirements precision/unambiguous

Problems arise when requirements are not precisely stated

Ambiguous requirements may be interpreted in different ways by developers and users

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

Consider the term 'search' in requirement 2.3

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 a clinic at a time then search? OR

- search for a patient name in all clinics at once. User enters patient details and search in all available clinics?

Requirements completeness and consistency

In principle requirements should be both complete and consistent:

Complete
They should include descriptions of all services required including externally imposed/implied requirements.

Consistent
There should be no conflicts or contradictions in the descriptions of the system services

In practice, in large systems, it is very difficult or, most probably, impossible to produce a complete and consistent requirements document

Requirements Consistency: Example

Conflicts between different non-functional requirements are common in complex systems

Spacecraft system

To minimise weight, the number of separate chips in the system should be minimised

To minimise power consumption, lower power chips should be used

However, using low power chips may mean that more chips have to be used.

Which is the most critical requirement?

Requirement Traceability

Each requirement must be traceable

Traceability is critical for requirement verification, validation and user acceptance.

Requirements are often uniquely identified by a unique number to be traced/referenced in validation and testing phases.

Example: requirement has its own unique ID/Number

R2.3: A user shall be able to search the appointments lists for all clinics.