



جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

قسم الانظمة الطبية الذكية كلية العلوم

المحاضرة الرابعة

Software engineering

المادة : Software engineering

المرحلة : الثالثة

اسم الاستاذ: م.د أحمد عدنان المحنا

Functional & Non-Functional Requirements

System requirements may be either *Functional* or *Non-functional* requirements.

1- Functional requirements

- Statements of services the system should provide,
- How the system should react to particular inputs,
- How the system should behave in particular situations.

2- Non-functional requirements

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

3- Domain requirements

- Requirements that come from the application domain of the system and that reflect characteristics of that domain.
- They may be functional or non-functional requirements.

1- Functional Requirements (FR)

Are statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situation.

The major attributes of FR are:

- 1- Describe functionality or the system services (Functional system requirements should describe the system services in detail).
- 2- Describe an interaction between the system and its environment.
- 3- Depend on the type of software, expected users and the type of system where the software is used.
- 4- **FR** are independent of the implementation of a solution, they do not place constraints on how the solution will be developed.
- 5- Functional user requirements may be high-level statements of what the system should.
- 6- Problems arise whenever **functional requirements** are not precisely stated (**Ambiguous Requirements**).
- 7- Ambiguous requirements may be interpreted in different ways by developers and users.

In principle, the functional requirements should be both complete and consistent.

- **Completeness**: all services required by the user should be defined and should be includes descriptions of all facilities required.
- **Consistency**: there should be no conflicts in the descriptions of the system facilities.

In practice, for large systems, it is mostly impossible to produce a complete and Consistent requirements document because of overall complexity and varying views.

2- Non-Functional Requirements (NFR)

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

The major attributes of NFR are:

- 1- Define system properties and constraints, e.g. of system properties are performance, reliability, response time and storage requirements. Constraints are I/O device capability, system representations, etc.
- 2- Process requirements may also be specified mandating a particular CASE toolset, programming language and a description of the process model which should be followed.

Customers impose these process requirements for two reasons:

- **System quality:** In general, a good process leads to a good product.
 - **System maintainability:** Process requirements may be imposed so that the development methods used to design and implement the system are compatible with those to be used for system maintenance.
- 3- **NFR** may be more critical than functional requirements. If non-functional requirements are not met, the system is useless.
 - 4- **NFR** arise through user needs, because of budget constraints, organizational policies, safety, and privacy and so on.

Types of Non-Functional requirements:

NFR can be classified depending on how they have been derived:

- A- Product requirements:** These are requirements which specify that the delivered product must behave in a particular way, e.g. execution speed; reliability; portability and usability requirements. This type of NFR may be derived directly from user needs.
- B- Organizational requirements:** These are requirements which are a consequence of organizational policies and procedures, e.g. Process standards used, Implementation requirements such as programming language or design method used, etc. This type of NFR may be derived from both the customer's and developer's organization.
- C- External requirements:** requirements arising from factors that are external to the system and its development process, such as examples :

interoperability requirements which define how the system interacts with systems in order organizations; **legislative** requirements which must be followed to ensure that the system operates within the law; and **ethical** requirements which may be placed on the system to ensure that it will be acceptable to its users and the general public.

The following **Figure 4.1** shows different types of Non-functional requirements.

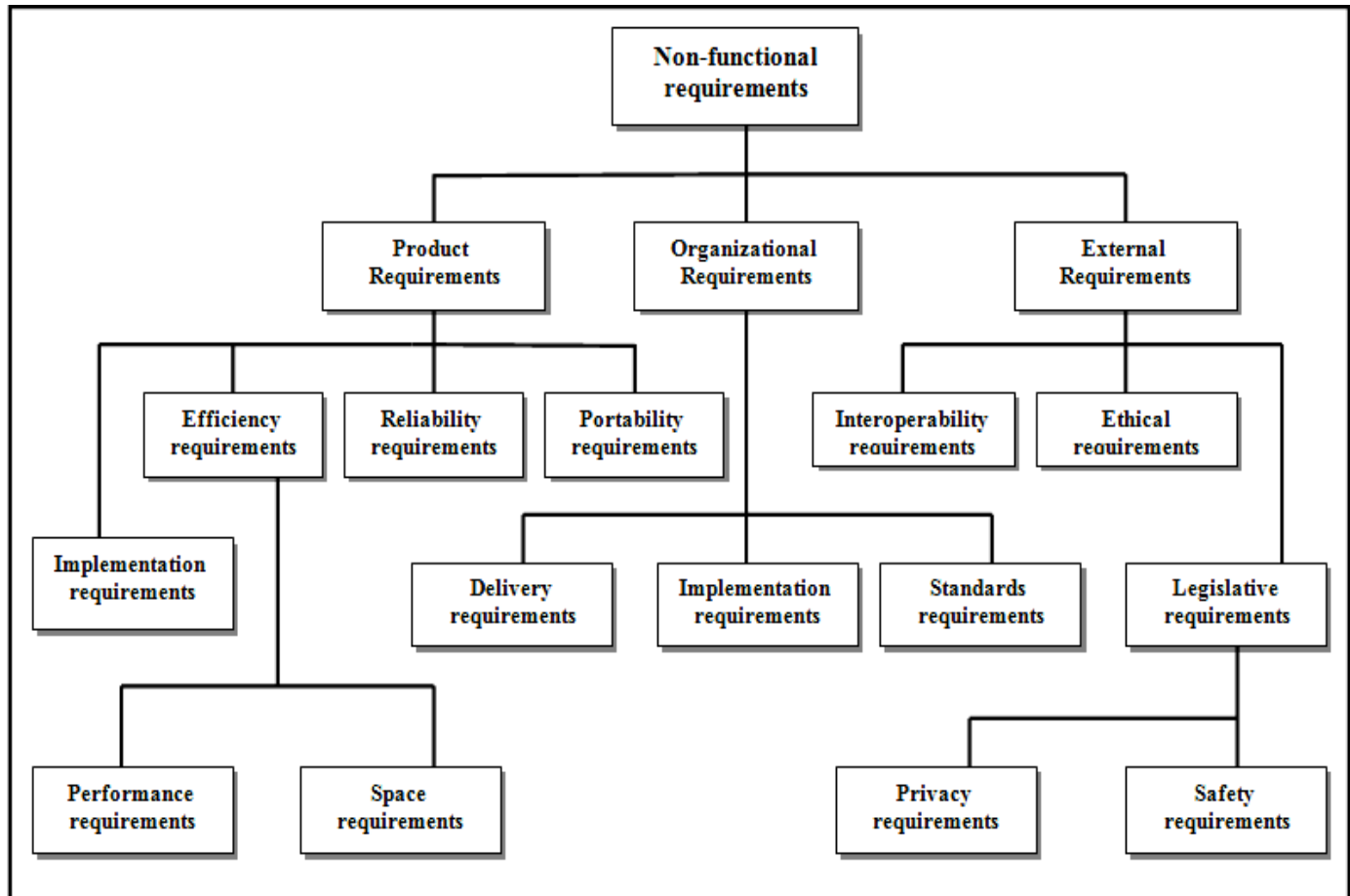


Figure 4.1: Types of Non-functional requirements.

- **NFR** may be very difficult to state precisely but imprecise requirements are routinely hard to verify.
- The best way to ensure that NFR are verifiable is to express them quantitatively based on a **measure** or **metric** that can be objectively tested.

Table 4.1 shows a number of possible measures that may be used to specify non-functional system properties.

- **The system goals** are helpful to developers as they convey the intentions of the system users. The system should be easy to use by experienced controllers and be organized in such a way that user errors are minimized.
- **Experienced controllers** shall be able to use all the system functions after a total of two hours training. After this training, the average number of errors made by experienced users shall not exceed two per day.
- **Conflicts** between different non-functional requirements are common in complex systems.
- **Optimizations** along the lines of several different goals might contradict each other; need to priorities goals!

Table 4.1: Metrics for specifying NFR properties.

Property	Measure
Speed	Processed transactions/second User / Event response time Screen refresh time
Size	K Bytes Number of RAM chips
Ease of use	Training time Number of help frames
Reliability	Mean time of 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



Al-Mustaqbal University
College of Science
Intelligent Medical System Department
