

CMP 2101: Software Engineering

1. Requirements

It's not enough to do your best: You must KNOW what to do and THEN do your best

-- W. Edwards Deming

Requirements are a consistent and complete description of the services provided by the system and its operational constraints. They also capture the external behaviour of the system.

They reflect the needs/expectations of a customer.

Note: 1. Requirements are a description of **WHAT** the system does and **NOT HOW** it does things.

2. *Customers rarely know the system 'what' but know the 'how', the engineer can then guide on requirements.*

Goal of Requirements

To be sure that we understand the problem, before we set out to provide a solution.

1.1 Necessary properties of Requirements

- Unambiguous: The meaning should be clear
- Consistent: One requirement must not contradict another
- Complete: E.g. "The process is terminated if the wrong PIN has been entered more than a certain number of times." (this requirement statement is both incomplete and ambiguous- What is that "certain number of times"?)
- Verifiable: a requirement that cannot be tested is not a requirement. E.g. "The system should work in real-time mode." (What is "real time" here?)

1.2 Sources of Requirements

- Stakeholders – people who will be affected by the system, directly or indirectly e.g. End users, System administrators, Engineers maintaining the system, Business managers
- Documentation e.g. organisational charts, process models or standards, and user or other manuals of existing systems.
- Business area/domain
- Existing System(s); These may be external or internal to an organisation

To understand and come up with requirements one needs to look specifically at:

- Organization - formal structure
- Existing Systems - how they operate and how they are built
- Processes - operating procedures, description of tasks
- Improvements - what needs to be changed

1.3 Examples of Requirements

1. The system **shall** allow users to select search for a vehicle by make, model, or year of manufacture.
2. The system interface **shall** be implemented using a web browser.
3. The system **shall** support at least 10 transactions per second. - *performance and efficiency captured (constraint)*

CMP 2101: Software Engineering

1.4 Types of Requirements

A. Functional Requirements

These describe;

- what the system does
- the interaction between the system and the environment or how the system should respond to stimuli.

They are statements of **services** the system should provide, how the system should **react** to particular inputs and how the system should **behave** in particular situations.

Functional requirements should be both **complete** and **consistent**.

- a. Completeness: All services required by the user should be defined
- b. Consistency: Requirements should not have contrary definitions

For large and complex systems it is difficult to meet the above goals as it is easy to make mistakes and different stake holders may often have different and inconsistent needs

B. Non-functional Requirements:

Non-functional requirements define the overall qualities or attributes of the resulting system

Non-functional requirements place restrictions on the product being developed, the development process, and specify external constraints that the product must meet.

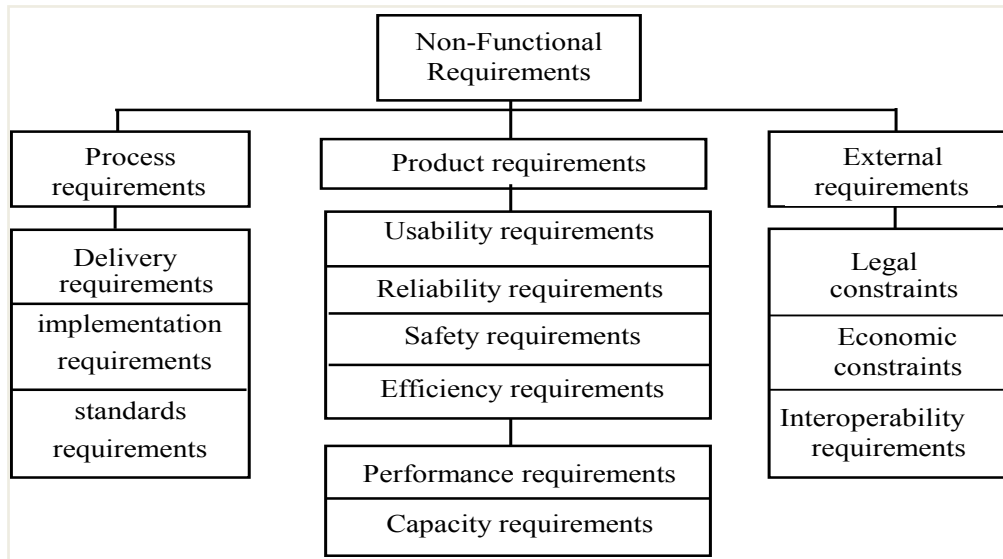
The 'IEEE-Std 830 - 1993' lists 13 non-functional requirements to be included in a Software Requirements Specification.

- Performance requirements
- Interface requirements
- Operational requirements
- Resource requirements
- Verification requirements
- Acceptance requirements
- Documentation requirements
- Security requirements
- Portability requirements
- Quality requirements
- Reliability requirements
- Maintainability requirements
- Safety requirements

CMP 2101: Software Engineering

Classification of Non-Functional Requirements

A more general classification distinguishes between product, process and external requirements.



CMP 2101: Software Engineering

A. Product requirements

These specify the desired characteristics that a system or subsystem must possess.

Some product requirements can be formulated precisely, and thus easily quantified e.g. Performance, Capacity while others are more difficult to quantify and, consequently, are often stated informally e.g. Usability.

Examples:

- ☞ *The System service X shall have an availability of 999/1000 or 99%. - reliability requirement*
- ☞ *System Y shall process a minimum of 8 transactions per second. – performance requirement.*
- ☞ *The executable code of System Z shall be limited to 512Kbytes. - space requirement which specifies the maximum memory size of the system*
- ☞ *The system shall be developed for PC and Macintosh platforms. - portability requirement which affects the way in which the system may be designed.*
- ☞ *The system must encrypt all external communications using the RSA algorithm. - security requirement which specifies that a specific algorithm must be used in the product.*

B. Process requirements

These are constraints placed upon the development process of the system. They include:

- Requirements on development standards and methods which must be followed
- CASE tools which should be used
- The management reports which must be provided

Examples

- ☞ The development process to be used must be explicitly defined and must be conformant with ISO 9000 standards.
- ☞ The system must be developed using the XYZ suite of CASE tools
- ☞ Management reports setting out the effort expended on each identified system component must be produced every two weeks
- ☞ A disaster recovery plan for the system development must be specified

C. External requirements

These may be placed on both the product and the process and are derived from the environment in which the system is developed. They are based on:

- application domain information
- organisational considerations
- the need for the system to work with other systems
- health and safety or data protection regulations e.t.c.

Examples:

- ☞ ***Medical data system:** The organisation's data protection officer must certify that all data is maintained according to data protection legislation before the system is put into operation.*

The requirement above comes from the need for the system to conform to data protection legislation.

CMP 2101: Software Engineering

Requirements documentation Jargon

- ☞ Must/Shall - system has to do this, cannot be delivered without
- ☞ Should - important, but not essential for the system to do this
- ☞ Will - possibility, but could be eliminated or postponed to another release

Note:

There is no clear distinction between functional and non-functional requirements. Whether or not a requirement is expressed as a functional or a non-functional requirement may depend on:

- the level of detail to be included in the requirements document
- the degree of trust which exists between a system customer and a system developer.

For example: *The system shall ensure that data is protected from unauthorised access.*

The above stated requirement would conventionally be considered as a non-functional requirement because it does not specify specific system functionality which must be provided. However, it could have been specified in slightly more detail as follows:

The system shall include a user authorisation procedure where users must identify themselves using a login name and password. Only users who are authorised in this way may access the system data.

In this form, the requirement looks rather more like a functional requirement as it specifies a function (user login) which must be incorporated in the system.

1.5 User Requirements vs System Requirements

- *User Requirements*
 - Used to describe the requirements in informal language and in broad terms.
 - Intended, e.g., to solicit bids from software companies
- *System Requirements*
 - – More precise than user requirements
 - – Used in the contract phase to define how the system should work

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., 10 mg, 20 mg) 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.

Figure 1: Example of User Requirements vs System Requirements