

SOFTWARE REQUIREMENT ANALYSIS AND ESTIMATION

UNIT - 3

Some examples of valid software requirements

- Requirement #1:
 - The system shall maintain records of all payments made to employees on accounts of salaries, bonuses, travel/daily allowances, medical allowances, etc.

Examples of requirements

- Requirement #2:
 - The system shall interface with the central computer to send daily sales & inventory data from every retail store.

Examples of requirements

- Requirement #3:
 - The system shall maintain records of all library materials including books, serials, news papers & magazines, video & audio tapes, reports, collections of transparencies, CD-ROM, DVD, etc.

Examples of requirements

□ Requirement #4:

- The system shall allow user to search for an item by title, author, or by international standard book number.
- The system user interface shall be implemented using a web browser.

Examples of requirements

- Requirement #5:
 - The system shall support at least 20 transactions per second.
 - The system facilities which are available to public users shall be demonstrable in 10 minutes or less

Kinds of software requirements

- Functional requirements
- Non functional requirements
- Domain requirements
- Inverse requirements
- Design & implementation constraints

Functional Requirements

- Describing what the system does, or capture the functionality of the system
- Functional requirement are the backbone of the software requirement.
- It depends on the complexity of the software system.

Examples:

- Reaction of the particular input.
- Behavior of the particular situation
- Address the sequencing & parallelism task.
- Capturing the exception handling & abnormal behavior activities.

Examples of functional requirement:

□ Requirement #1:

- The system shall solve a quadratic equation using the following formula.

$$X = (-b \pm \sqrt{b^2 - 4 * a * c}) / 2 * a$$

Examples of functional requirement:

□ Requirement #2:

- The user shall be able to search either the entire database of patients or select a subset from it (admitted patients or patients with asthma, etc)

Examples of functional requirement:

- Requirement #3:
 - The system shall provide appropriate viewers for the user to read documents in the document store.

Examples of functional requirement:

- Requirement #4:
 - Every order shall be allocated a unique identifier (ORDER_ID) which the user shall use to access that order.

Examples of functional requirement:

- Requirement #5:
 - The system shall allow customers to return non perishable items within 15 days of the purchase. A customer must present the original sale receipt to return an item.

Kinds of Software Requirements

- Functional requirements
- Non-functional requirements
- Domain requirements
- Inverse requirements
- Design and implementation constraints

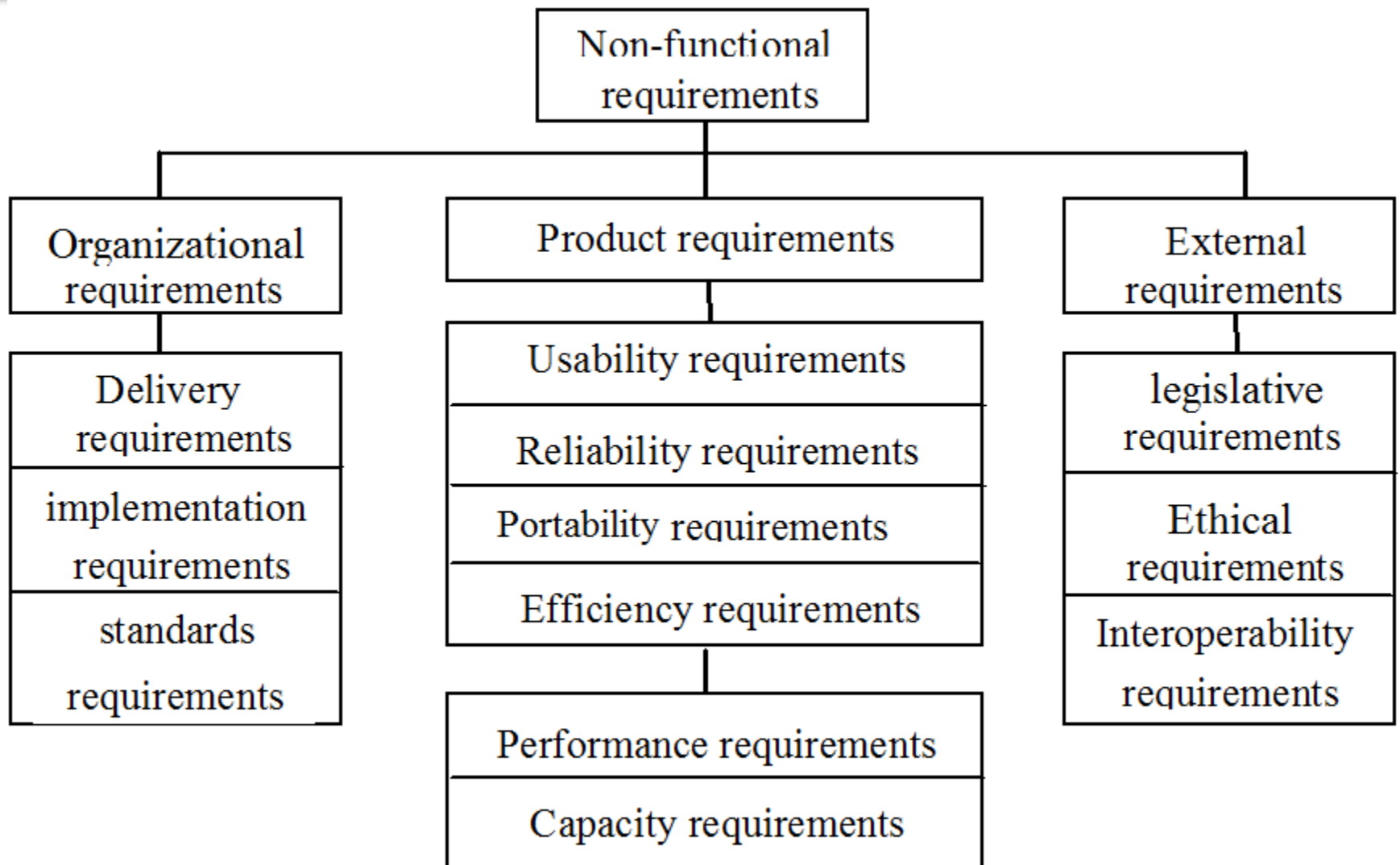
Non-functional requirements (NFR)

- 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.
- Examples of NFR include safety, security, usability, reliability and performance requirements.

Classification of NFRs

- NFRs may be classified in terms of qualities that a software must exhibit (Boehm)
- A more general classification distinguishes between product, Organizational and external requirements

Classification of NFRs (contd.)



Product requirements

- Specify the desired characteristics that a system or subsystem must possess.
- Most NFRs are concerned with specifying constraints on the behaviour of the executing system.
- Product
 - Usability, reliability, Portability, efficiency (performance, space)

Example of product requirement

- The system shall allow one hundred thousand hits per minute on the website.
- The shall not have down time of more than one second for continues execution of the 1000 hours.

Organizational Requirement

- Derived from the policies and procedures of the customers or the developing organizations.
- Organizational
 - Standards, implementation, delivery

Example of organizational requirement

- The system development process and deliverable documents shall conform to the MIL-STD-2167A
- Any development work sub - contracted by the development organization shall be carried out in accordance with capability maturity model.

External Requirement

- Which are derives from the factors, external to the system, external to the software system and sometimes organization and its development process.
- External
 - Interoperability, ethical, legislative (privacy, safety)

Example of External requirement

- The system shall not disclose any personal information about members of the library system to other members, except system administrator

NFRs as Goals

25

- Non-functional requirements are sometimes written as general goals, which are difficult to verify
- They should be expressed quantitatively using metrics (measures) that can be objectively tested

Example: Goal converted into an NFR

26

- Goal (unverifiable)
 - The system should be easy to use by experienced controllers and should be organized in such a way that user errors are minimized
- Non-functional requirement (verifiable)
 - 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

Metrics for Non-Functional Requirements (NFRs)

Metrics for NFRs - 1

Property	Measure
Speed	1. Processed transactions/second 2. Response time 3. Screen refresh time

Metrics for NFRs - 2

29

Property	Measure
Size	1. K bytes

Metrics for NFRs - 3

Property	Measure
Ease of use	1. Training time 2. Number of help frames

Metrics for NFRs - 4

31

Property	Measure
Reliability	1. Mean time to failure 2. Probability of unavailability 3. Rate of failure occurrence 4. Availability

Metrics for NFRs - 5

32

Property	Measure
Robustness	1. Time to restart after failure 2. Percentage of events causing failure 3. Probability of data corruption on failure

Discussion on Metrics for NFRs

33

- With the help of these measures the NFRs can be verified quantitatively
- It should also be noted that the cost of quantitatively verifying each NFR may be very high

Domain Requirement

- Requirement (either functional or non-functional) that come from the application domain of the system, reflecting characteristics of the domain. (e.g. physical laws, regulatory standards)
- Domain requirements reflect the **environment in which the system operates** so, when we talk about an application domain we mean environments such as **train operation, medical records, e-commerce etc.**
- Domain requirements may be expressed using specialized domain terminology or reference to domain concepts. Because these requirements are specialized, software engineers often find it difficult to understand how they are related to other system requirements.
- Domain requirements are important because they often reflect fundamentals of the application domain. If these requirements are not satisfied, it may be impossible to make the system work satisfactorily.

Examples of Domain requirement

- The hospital management system should support the check-in and check out for IPD (In Patient) and oPD Out patient Department
- The system should have the ability to post Double entries
- The system should have the provision of maintaining chart of accounts
- System will keep claim records & losses reserves, Financial Transactions and Claim expense records
- Supporting associate agents at the policy level setup commission due based on written premium, on invoicing or on premium payments and generating agent statements

Inverse Requirement

- It describes the constraints on allowable behavior.
- In many cases, it is easier to state that certain behavior must never occur than to state requirements guaranteeing acceptable behavior in all circumstances.
- In simply which states that something should not be done.

Examples of Inverse Requirements

- User ID should only contain digits.
- The system should not use red color in the user interface, whenever it asking for a input from the user.

Design and Implementation Constrains

- Design and Implementation Constrains are boundary conditions on how the software is to be constructed and implemented.
- They are givens of the development within which the designer must work.
- Example: Software must run using a certain database system, must fit into the memory of 512 KB

Thank You



Classify the following requirement

Class Quiz

R1. A new client must pay 30 percent of the estimated consulting fee and travel expenses in advance.”

R2. Time-off approvals must comply with the company’s HR vacation policy.

R3. If the pressure exceeds 40.0 psi, the high-pressure warning light should come on.

R4. The user must be able to sort the project list in forward and reverse alphabetical order.

R5. “The manufacturing execution system must control the wafer sorter.

R6. The mobile app should send the check image to the bank after I photograph the check I’m depositing.

R7. Files submitted electronically cannot exceed 10 MB in size.

R8. The browser must use 256-bit encryption for all secure transactions