

Software Engineering

Week 5

Muhammad Daniyal Liaquat

Topics

- ✧ Requirement Engineering Process
- ✧ Functional and non-functional requirements

Requirements engineering

- ✧ The process of establishing the services that the customer requires from a system and the constraints under which it operates and is developed.
- ✧ The requirements themselves are 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 requirement

❖ User requirements

- Statements in natural language plus diagrams of the services the system provides and its operational constraints. Written for customers.
- Purpose: They help the customer understand the system's features and services without needing technical knowledge. (window & Linux)

❖ System requirements

- A structured document setting out detailed descriptions of the system's functions, services and operational constraints. Defines what should be implemented so may be part of a contract between client and contractor.

❖ Business Requirements:

These are used to state the high-level business objective of the organization or customer requesting the system or product.

They are used to document main system features and functionalities without going into their nitty-gritty details.

They are captured in a document describing the project vision and scope.

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.

1. System requirements specification

1. On the last working day of each month, a summary of the drugs prescribed, their cost and the prescribing clinics shall be generated.
2. The system shall automatically generate the report for printing after 17.30 on the last working day of the month.
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.
4. If drugs are available in different dose units (e.g. 10mg, 20 mg, etc.) separate reports shall be created for each dose unit.
5. Access to all cost reports shall be restricted to authorized users listed on a management access control list.

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.
- May state what the system should not do.
- They bring-in the system's view and define from the system's perspective the software functionality the developers must build into the product to enable users to accomplish their tasks stated in the user requirements - thereby satisfying the business requirements.

◊ Non-functional requirements

- Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.
 - Often apply to the system as a whole rather than individual features or services.
- Define a software requirement as a document that describes all the services provided by the system along with the constraints under which it must operate.

Functional and non-functional requirements

- Constraints are restrictions that are placed on the choices available to the developer for design and construction of the software product.
- These kinds of requirements are called Non-Functional Requirements.
- These are used to describe external system interfaces, design and implementation constraints, quality and performance attributes.
- These also include regulations, standards, and contracts to which the product must confirm.
- If these requirement are no captured properly, system may not fulfil some of the basic business needs
- These requirements determine system architecture and framework. Without proper care system can collapse.
- An example: A software may required to run on sun Solaris. If System is built only for windows platform than its useless for clients. This functionality therefore also has an impact on system architecture without a change in the functionality.
- All functional requirements must be derived from user requirements
- As it is definition phase where emphasis is on what and not how therefore requirements must not include design and implementation details.

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.
- ✧ Functional system requirements should describe the system services in detail.

Functional requirements for the MHC-PMS

- ✧ A user shall be able to search the appointments lists for all clinics.
- ✧ Example: "A user shall be able to search the appointment lists across all clinics by patient name or date."
- ✧ The system shall generate each day, for each clinic, a list of patients who are expected to attend appointments that day.
- ✧ Each staff member using the system shall be uniquely identified by his or her 8-digit employee number.

Requirements imprecision

- ✧ Problems arise when requirements are not precisely stated.
- ✧ Ambiguous requirements may be interpreted in different ways by developers and users.
- ✧ Consider the term ‘search’ in requirement 1
 - 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 clinic then search.

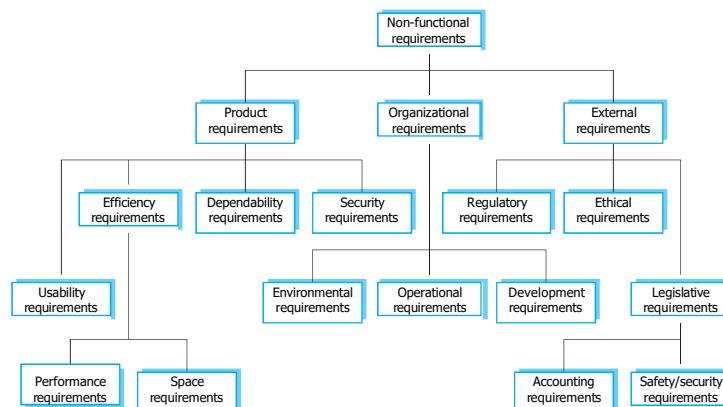
Requirements completeness and consistency

- ✧ In principle, requirements should be both complete and consistent.
- ✧ Complete
 - They should include descriptions of all facilities required.
- ✧ Consistent
 - There should be no conflicts or contradictions in the descriptions of the system facilities.
- ✧ In practice, it is impossible to produce a complete and consistent requirements document.

Non-functional requirements

- ✧ These 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 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.

Types of nonfunctional requirement



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. patient secure

Examples of nonfunctional requirements in the MHC-PMS

Product requirement

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

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

External requirement

The system shall implement patient privacy provisions as set out in HStan-03-2006-priv.

Usability requirements

- ✧ The system should be easy to use by medical staff and should be organized in such a way that user errors are minimized. (Goal)
- ✧ Medical staff shall be able to use all the 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)

Metrics for specifying nonfunctional requirements

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

- ✧ The system's operational domain imposes requirements on the system.
 - For example, a train control system has to take into account the braking characteristics in different weather conditions.
- ✧ Domain requirements be new functional requirements, constraints on existing requirements or define specific computations.
- ✧ If domain requirements are not satisfied, the system may be unworkable.

Train protection system

- ✧ This is a domain requirement for a train protection system:
- ✧ The deceleration of the train shall be computed as:
 - $D_{train} = D_{control} + D_{gradient}$
 - where $D_{gradient}$ is $9.81\text{ms}^2 * \text{compensated gradient}/\alpha$ and where the values of $9.81\text{ms}^2 / \alpha$ are known for different types of train.
- ✧ It is difficult for a non-specialist to understand the implications of this and how it interacts with other requirements.

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.

Difference b/w different requirements

- A word processing system without a spell checker. To sell this software spellchecker is must
- **Business Requirement:** “User will be able to correct spelling errors in a document efficiently.”
- Identifying tasks to accomplish the above mentioned requirements will result in **user requirements:** “finding spelling errors in the document and decide whether to replace each misspelled word with the one chosen from a list of suggested words”.
- What is the functionality provided by the system and how will it help the user to accomplish these tasks. The functional requirements therefore are “the spell checker will find and highlight misspelled words. It will then display a dialog box with suggested replacements. The user will be allowed to select from the list of suggested replacements. Upon selection it will replace the misspelled word with the selected word. It will also allow the user to make global replacements.”

Difference b/w Requirements

Non functional Requirements could require that it must be integrated into the existing word processor that runs on windows platform.

Stake Holders:

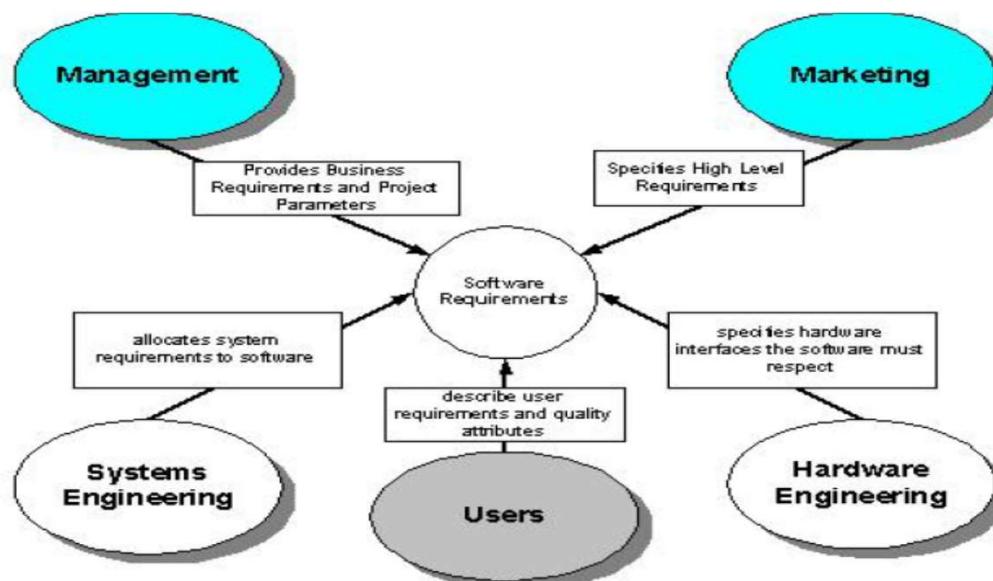
It is important to involve all kind of users in the RE. Toward this goal first phase would be to identify all stake holders.

Management important but actual users who use system and accept or reject system are very important.

Ignoring any stake holder may result in system failure.

Diagram: The role of different stake holders in setting the system requirements

Role of Stake Holders in RE



Requirement Statement and Requirement Specification Document

Different levels of software requirements are documented in different documents.

The **two main documents** produced during this phase are **Requirement Statement** and **Requirement Specification**. They are also called *Requirement Definition* and *Functional Specification*.

These are used to document user requirements and functional requirements respectively.

Good Requirement Statement Characteristics

- **Complete** - Each requirement must fully describe the functionality to be delivered.
- **Correct** - Each requirement must accurately describe the functionality to be built.
- **Feasible** - It must be possible to implement each requirement within the known capabilities and limitations of the system and its environment.
- **Prioritised** - An implementation priority must be assigned to each requirement, feature or use case to indicate how essential it is to a particular product release.
- **Unambiguous** - All readers of a requirement statement should arrive at a single, consistent interpretation of it.
- **Verifiable** – User should be able to devise a small number of tests or use other verification approaches, such as inspection or demonstration, to determine whether the requirement was properly implemented.

Good Requirement Specification Characteristics

- Complete - No requirement or necessary information should be missing.
- Consistent – No requirement should conflict with other software

Example. The following set of non-functional requirements was stated for a particular embedded system. (1) All programs must be written in Ada (2) The program must fit in the memory of the embedded micro-controller

These requirements conflicted with one another because the code generated by the Ada compiler was of a large footprint that could not fit into the micro-controller memory.