Requirements Engineering

The process of establishing the **services** that a **customer requires** from a system and the **constraints(ограничувања)** under which it **operates** and is **developed**.
The system requirements are the **descriptions** of the **system services and constraints** that are generated during the requirements engineering process.

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**.

System requirements

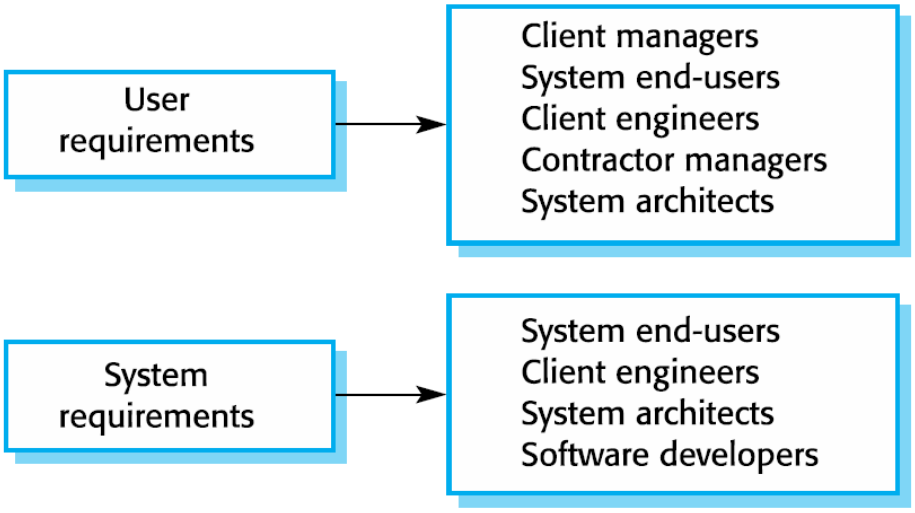
- A structured document setting out **detailed descriptions** of the system's **functions**, **services** and operational **constraints**. Defines **what should be implemented** so may be part of a contract between client and contractor.

User requirements definition

1. The Mentcare system 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 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. 10mg, 20mg, etc) separate reports shall be created for each dose unit.
1.5 Access to drug cost reports shall be restricted to authorized users as listed on a management access control list.



System stakeholders(ЗАСЕГНАТИ СТРАНИ- ЧИНИТЕЛИ)

Any **person** or **organization** who is **affected** by the system in some way and so who has a legitimate **interest**

Stakeholder types:

- End users
- System managers
- System owners
- External stakeholders

Agile methods and requirements

Agile methods usually use **incremental requirements engineering** and may express requirements as '**user stories**'.

This is **practical** for **business** systems but problematic for systems that require **pre-delivery analysis**(e.g. critical systems) or systems developed by **several teams**.

1)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**.

Requirements imprecision – проблеми се јавуваат кога барањата не се прецизно напишани.

МОРА ДА БИДАТ КОМПЛЕТНИ И КОНЗИСТЕНТНИ(Requirements completeness and consistency)

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.

Non-functional requirements may be **more critical** than functional requirements. If these are not met, the system may be **useless.**

Non-functional requirements may **affect** the overall **architecture** of a system rather than the **individual components**.

- Non-functional classifications
    - **Product requirements**(барањата кои се базираат на испорачани производи мора да се однесуваат на соодветен начин. Брзина на изработка, доверливост...)
    - **Organisational** requirements(барања кои се последица од организациските процедури. Имплементациски барања...)
    - **External** requirements(побарувања за интероперабилност, законодавни побарувања...)

Goal-A general intention of the user such as **ease of use**.

Usability requirements - Барања за употребливост.

Метрики кај не-функционалните барања се: брзина, големина, Леснотија на користење, доверливост, робустност, преносливност.

Domain requirements

Constraints on the system from the domain of operation

2)Requirements engineering processes

The processes used for RE vary widely depending on the **application domain**, the **people** involved and the **organisation** developing the requirements.

The requirements engineering process is an **iterative process** that includes requirements **elicitation**, **specification** and **validation**.

However, there are a number of generic activities common to all processes:

- **Requirements elicitation**;
- **Requirements analysis**;
- **Requirements validation**;
- **Requirements management**.

3)Requirements elicitation(Изнудување на барања)

Involves **technical staff** working with **customers** to find out about the application domain, the services that the system should provide and the system's operational **constraints**.

May involve end-users, managers, engineers involved in maintenance, domain experts, trade unions, etc. These are called **stakeholders**.

Stages include:

- **Requirements discovery**, разговор со чинителите за да ги дознаеме нивните барања
- **Requirements classification and organization**, групирање на барањата и нивно спојување во кохерентни групи.

Requirements prioritization and negotiation, приоритет на барањата и решавање на конфликтите во истите.

Requirements specification, барањата се документираат.

ПРОБЛЕМ: The requirements **change** during the analysis process. **New stakeholders** may emerge and the business environment may change

Formal or **informal** interviews with stakeholders are part of most RE processes.

Types of interview:

Closed interviews based on pre-determined list of questions

Open interviews where various issues are explored with stakeholders.

Етнографија – луѓето не треба да објаснуваат нивната работа(секој има свое право да работи што сака)

Stories and **scenarios** are a description of how a system may be used for a particular task.

Scenarios should include:

A description of the starting situation; ,**ОБЈАСНУВАЊЕ НА СЕ**

A description of the normal flow of events; **ОБЈАСНУВАЊЕ НА СИТЕ СЛУЧКИ**

A description of what can go wrong; **ШТО МОЖЕ ДА СЕ РАСИПЕ**

Information about other concurrent activities; **СЕГАШНИ АКТИВНОСТИ**

A description of the state when the scenario finishes. **СОСТОЈБА ПО КРАЈОТ**

4)Requirements specification

The process of **writing** down the **user** and **system requirements** in a requirements **document**.

User requirements have to be understandable by **end-users and customers** who do not have a technical background.

System requirements are more detailed requirements and may include more **technical information**.

In principle, **requirements** should state **what the system should do** and the **design** should describe **how it does this**.

Requirements are written as natural language **sentences** supplemented by **diagrams** and **tables**.

Problems with natural language

- Lack of clarity
- Precision is difficult without making the document difficult to read.
- **Requirements confusion (конфузија со барањата)**
- Functional and non-functional requirements tend to be mixed-up.
- **Requirements amalgamation(соединување со барања)**
- Several different requirements may be expressed together.

Form-based specifications

- Definition of the function or entity.
- Description of inputs and where they come from.
- Description of outputs and where they go to.
- Information about the information needed for the computation and other entities used.
- Description of the action to be taken.
- Pre and post conditions (if appropriate).
- The side effects (if any) of the function.

Tabular specification (кога сакаме да дефинираме поголем број алтернативни нивоа на акција)

The software requirements document is the **official statement** of **what is required** of the system developers.

- Should include both a **definition** of user requirements and a **specification** of the system requirements.
- It is **NOT** a **design** document. As far as possible, it should set of **WHAT** the system should do rather than **HOW** it should do it.

5)Requirements validation(барања за валидација-справување со било какви проблеми)

Concerned with **demonstrating** that the **requirements define the system** that the customer really wants.

- Requirements **error costs are high** so validation is very important
- **Fixing** a requirements **error** after delivery may cost up to **100 times the cost** of fixing an implementation error.

Requirements checking

Validity. Does the system provide the functions which best support the customer's needs?

Consistency. Are there any requirements conflicts?

Completeness. Are all functions required by the customer included?

Realism. Can the requirements be implemented given available budget and technology

Verifiability. Can the requirements be checked?

Requirements validation techniques

Requirements reviews

- Systematic manual analysis of the requirements.

Prototyping

- Using an executable model of the system to check requirements.

Test-case generation

- Developing tests for requirements to check testability.

Review checks

Verifiability

- Is the requirement realistically testable?

Comprehensibility

- Is the requirement properly understood?

Traceability

- Is the origin of the requirement clearly stated?

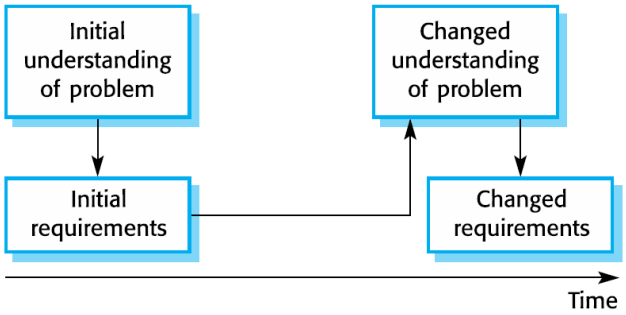
Adaptability

- Can the requirement be changed without a large impact on other requirements?

6)Requirements change(барање на промени)

Large systems usually have a diverse user community, with many users having different requirements and priorities that may be conflicting or contradictory.

Requirements evolution



Requirements management

Requirements management is the process of managing changing requirements during the requirements engineering process and system development.