UNIVERSITY of GUELPH

ENGINEERING

Introduction to Requirements Engineering

Fattane Zarrinkalam

Ian Sommerville, Software Engineering, Pearson, 10th edition (Chapter 4)

What is Requirements Engineering?

- The requirements for a system are the descriptions of the system services and constraints on its operation

- The process of finding out, analyzing, documenting and checking these services and constraints is called requirements engineering (RE).

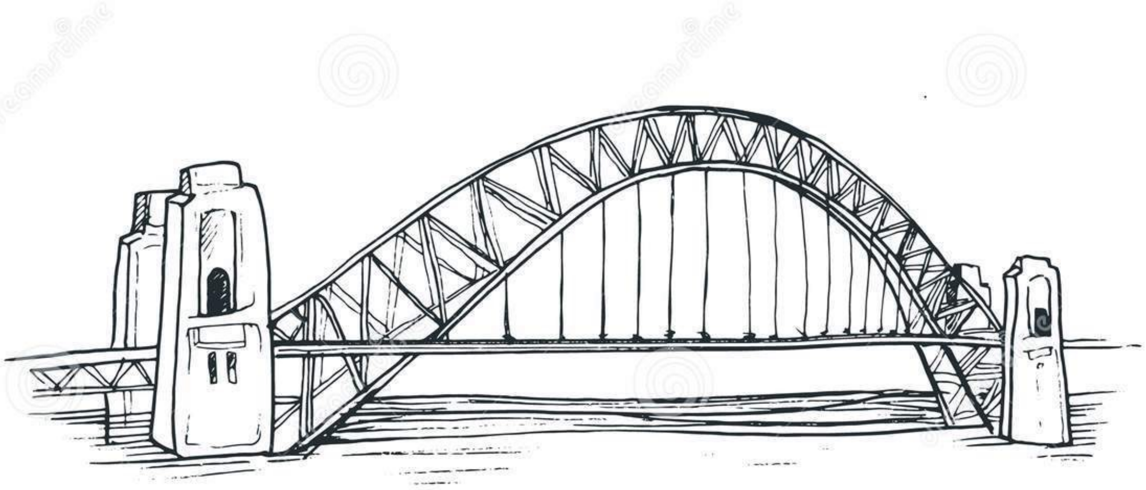
What is Requirements Engineering?

- Requirement engineering is usually presented as the first stage of the software engineering process.

- However, some understanding of the system requirements may have to be developed for feasibility study.
  - Feasibility study: assess whether or not the system is technically and financially feasible in order to help management decide whether or not to go ahead with the procurement or development of the system.

Why Requirements Engineering?

Real World

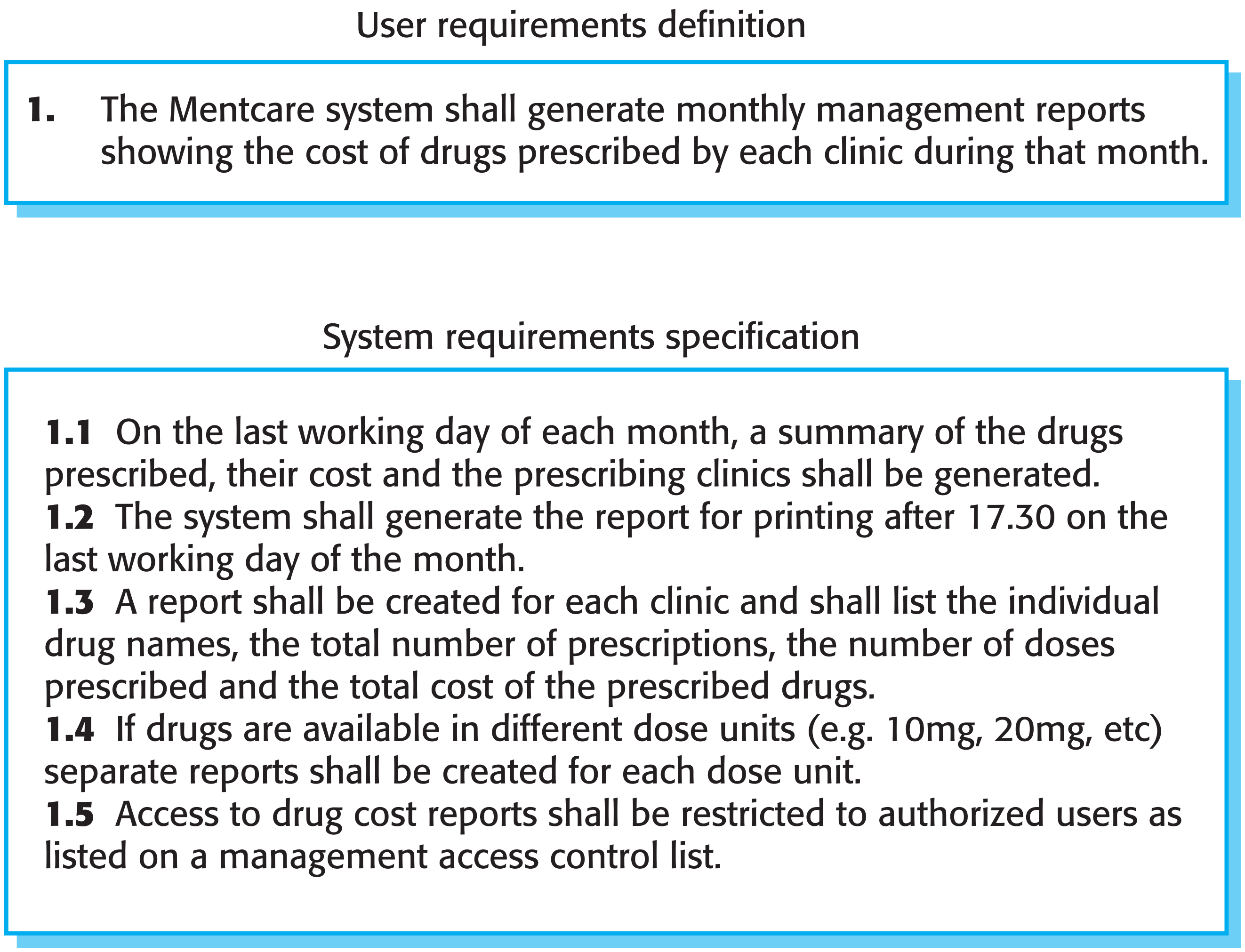


Software System

Types of Requirements

- 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 and System Requirements



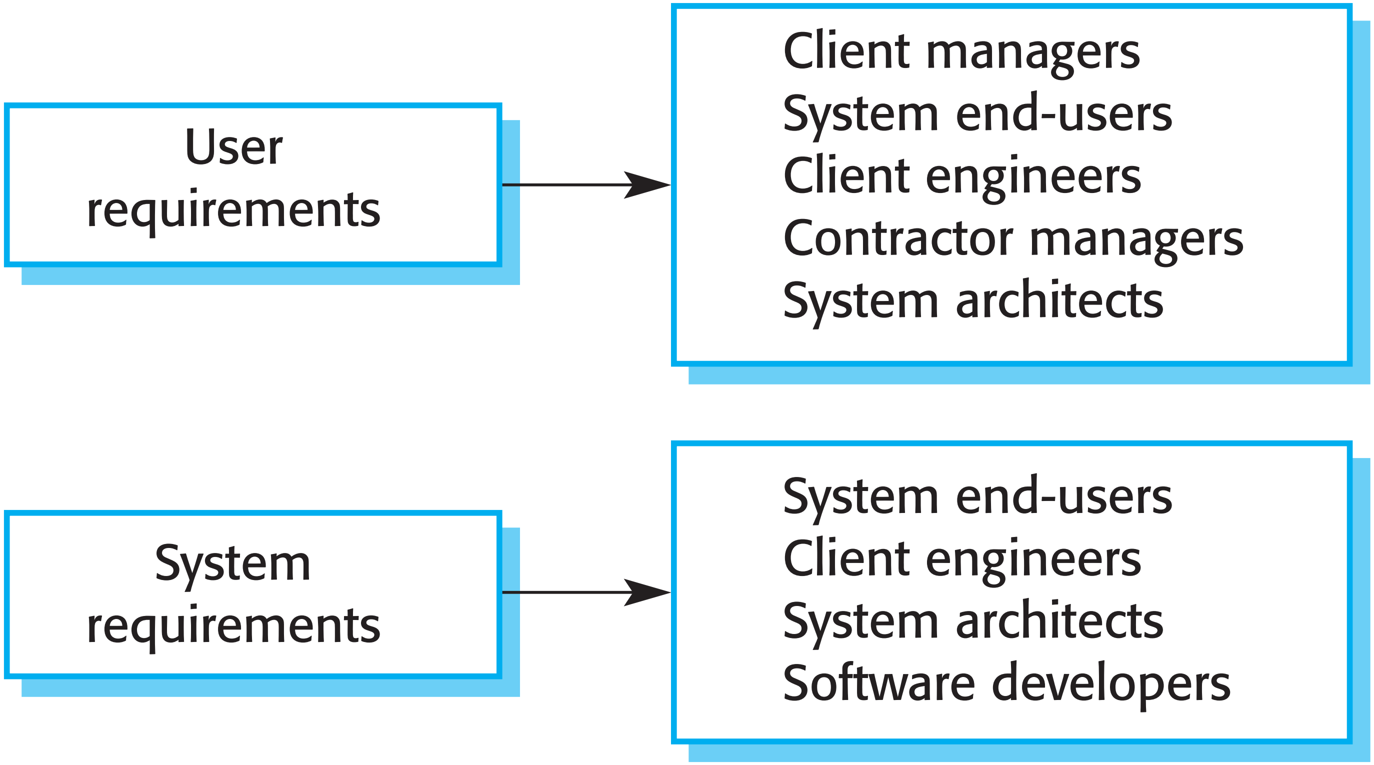
User requirements definition

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.

Readers of Different Types of Requirements Specification



User requirements

Client managers
System end-users
Client engineers
Contractor managers
System architects

System requirements

System end-users
Client engineers
System architects
Software developers

System Stakeholders

- 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

Stakeholders in the Mentcare System

- Patients whose information is recorded in the system.
- Doctors who are responsible for assessing and treating patients.
- Nurses who coordinate the consultations with doctors and administer some treatments.
- Medical receptionists who manage patients’ appointments.

Stakeholders in the Mentcare System

- A medical ethics manager who must ensure that the system meets current ethical guidelines for patient care.
- Health care managers who obtain management information from the system.
- Medical records staffs who are responsible for ensuring that system information can be maintained and preserved, and that record keeping procedures have been properly implemented.
- IT staffs who are responsible for installing and maintaining the system.

Agile Methods and Requirements

- Many agile methods argue that producing detailed system requirements is a waste of time as requirements change so quickly.
- The requirements document is therefore always out of date.
- 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.

Functional and Non-functional Requirements

Functional Requirements

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 system requirements vary from
  - general requirements covering what the system should do
  - to very specific requirements reflecting local ways of working or an organization's existing systems
- May state what the system should not do.

Functional Requirements for Mentcare

- A user shall be able to search the appointments lists for all clinics.
- 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 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, because of system and environmental complexity, it is impossible to produce a complete and consistent requirements document.

Non-functional Requirements

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.

Non-functional Requirements Implementation

- Non-functional requirements may affect the overall architecture of a system rather than the individual components.
  - For example, to ensure that performance requirements are met, you may have to organize the system to minimize communications between components.
- A single non-functional requirement, such as a security requirement, may generate a number of related functional requirements that define system services that are required.
  - It may also generate requirements that restrict existing requirements.

Non-functional Classification

- 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.

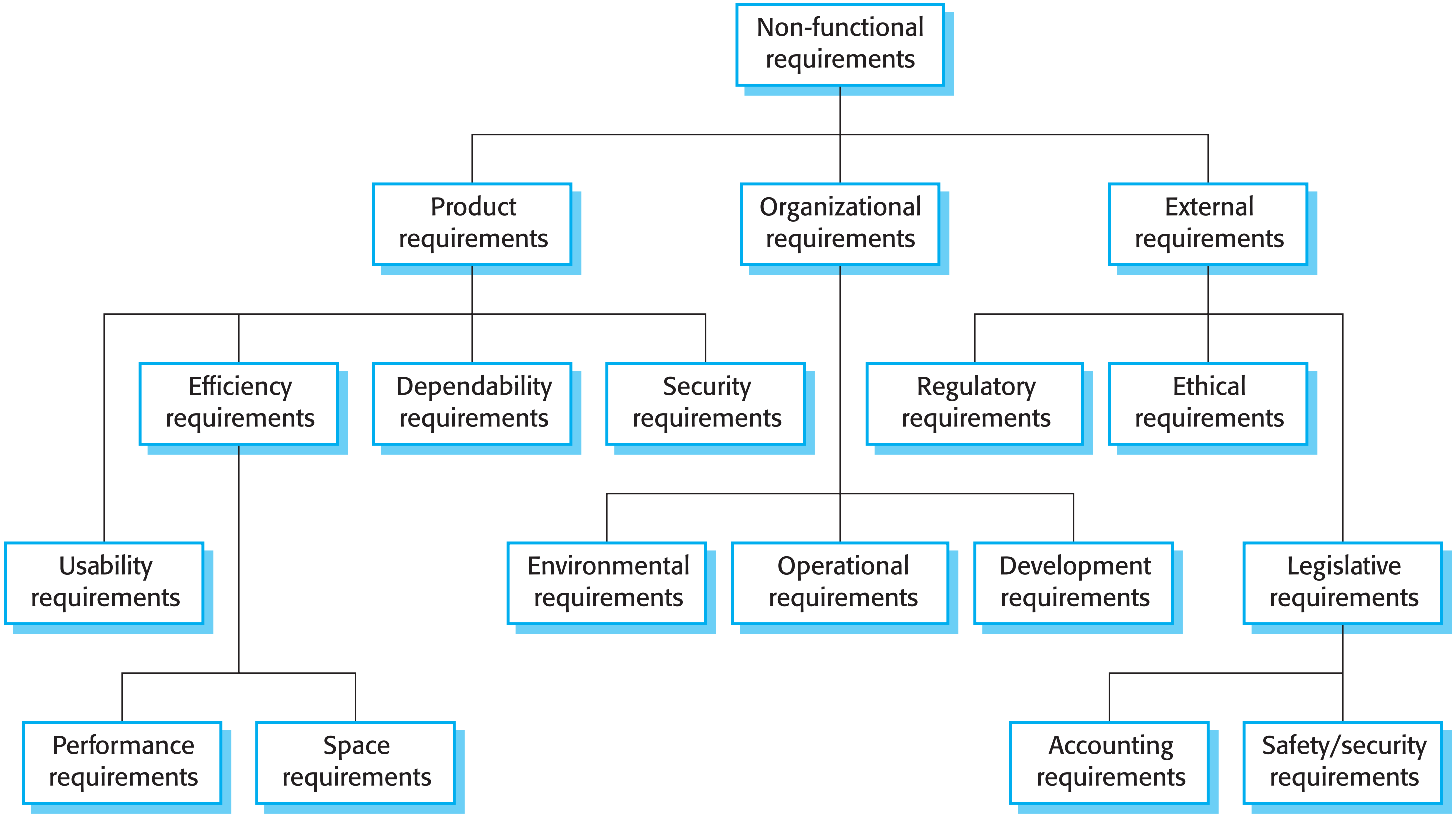
Non-functional Requirements in Mentcare

Product requirement
The Mentcare system 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 Mentcare 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.

Types of Non-functional Requirements



Goals and Requirements

- Non-functional requirements may be very difficult to state precisely, and imprecise requirements may be difficult to verify.
- Goal
  - A general intention of the user such as ease of use.
- Verifiable non-functional requirement
  - A statement using some measure that can be objectively tested.

Goals and 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 Non-Functional 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

Why Requirements Engineering is Important?

- If the requirements are wrong
  - The system may be delivered late and cost more that originally expected
  - They may not use its facilitate or may even decide to scrap it all together
  - The system may be unreliable in use with regular system errors and crashes disrupting normal operation.
  - If the system continues in use, the cost of maintain and evolving the system are very high

Why Requirements Engineering is Difficult?

- Difficulties with requirements:
  - Businesses operate in a rapidly changing environment so their requirements for system support are constantly changing
  - Multiple stakeholders with different goals and priorities are involved in the requirements engineering process.
  - System stakeholder don’t have clear idea of what they need
    - They can only describe their requirements in a vague and ambiguous way
  - Requirements are often influenced by political and organizational factors that stakeholders will not admit to publicly.

Summary

Summary

- Requirements define what a system should do and a desirable qualities of the system.
- Requirements engineering is a set of systematic techniques for defining and validating system requirements
- Functional requirements are statements of the services that the system must provide or are descriptions of how some computations must be carried out.
- Non-functional requirements often constrain the system being developed and the development process being used.
- Difficulties and problems with requirements are the major cause of system problems and project failures