Case studies

- A personal insulin pump
  - An embedded system in an insulin pump used by diabetics to maintain blood glucose control.
- A mental health case patient management system
  - Mentcare. A system used to maintain records of people receiving care for mental health problems.
- A wilderness weather station
  - A data collection system that collects data about weather conditions in remote areas.
- iLearn: a digital learning environment
  - A system to support learning in schools

4 – Requirements Engineering

Topics covered

- What is a Software requirement
- Functional and non-functional requirements
- Requirements engineering processes
- Requirement Engineering Phases
- Case Study

What is a requirement?

- **A high-level abstract statement** of a service
- **or**
- **a system constraint to a detailed** mathematical functional specification.
- Requirement can be
  - 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.

Principly a Requirement should be

- Complete
  - **descriptions of all facilities required.**
- Consistent
  - **no conflicts or contradictions in the descriptions**
- In practice, because of system and environmental complexity, it is impossible to produce a complete and consistent requirements document.

Types of requirement

- Software functions is described
- **User requirements**
  - Written in natural language for customers with diagram
- **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.

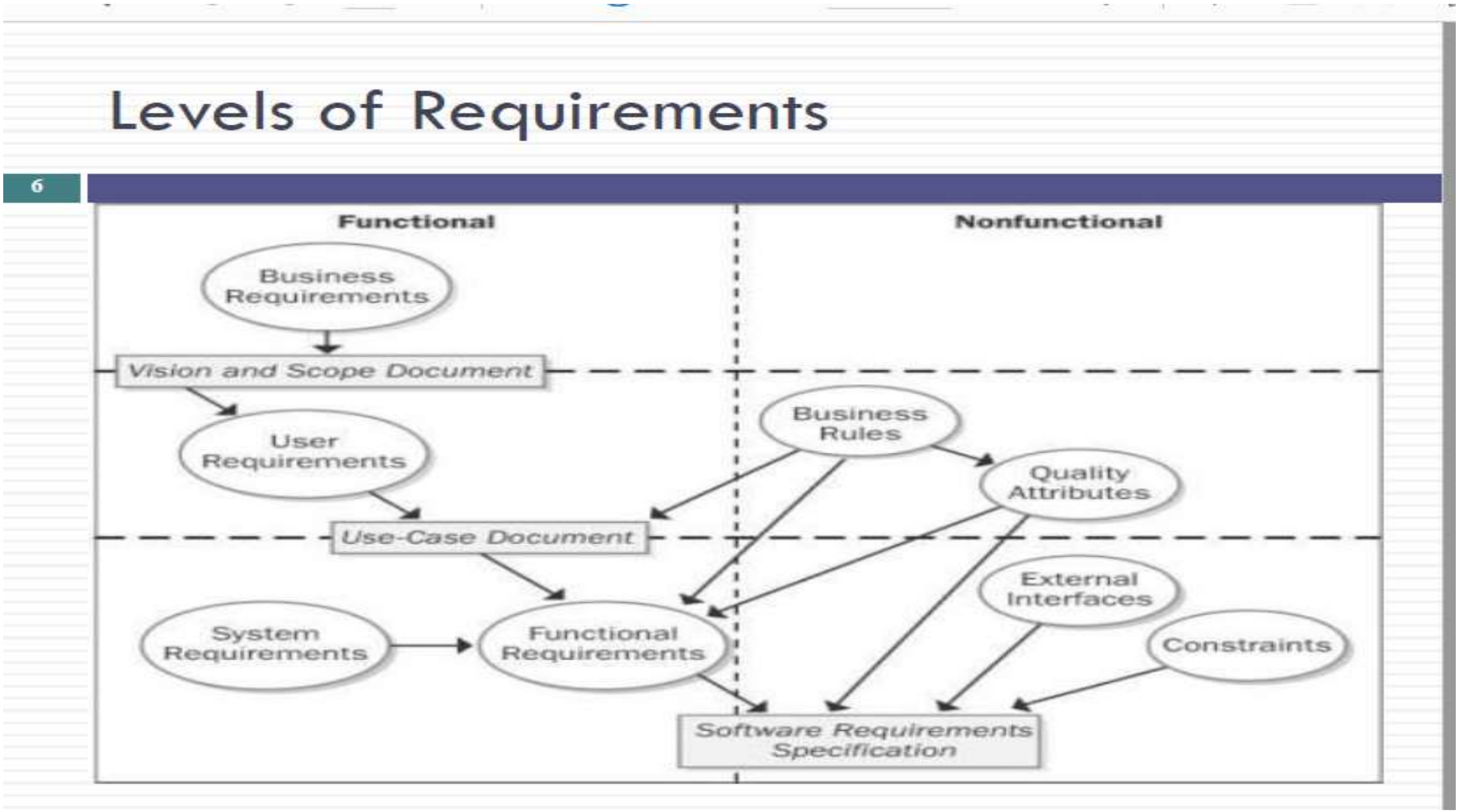
Requirements - Functional and non-functional

- **Functional requirements**
  – Statements of services the system should provide

  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.

+

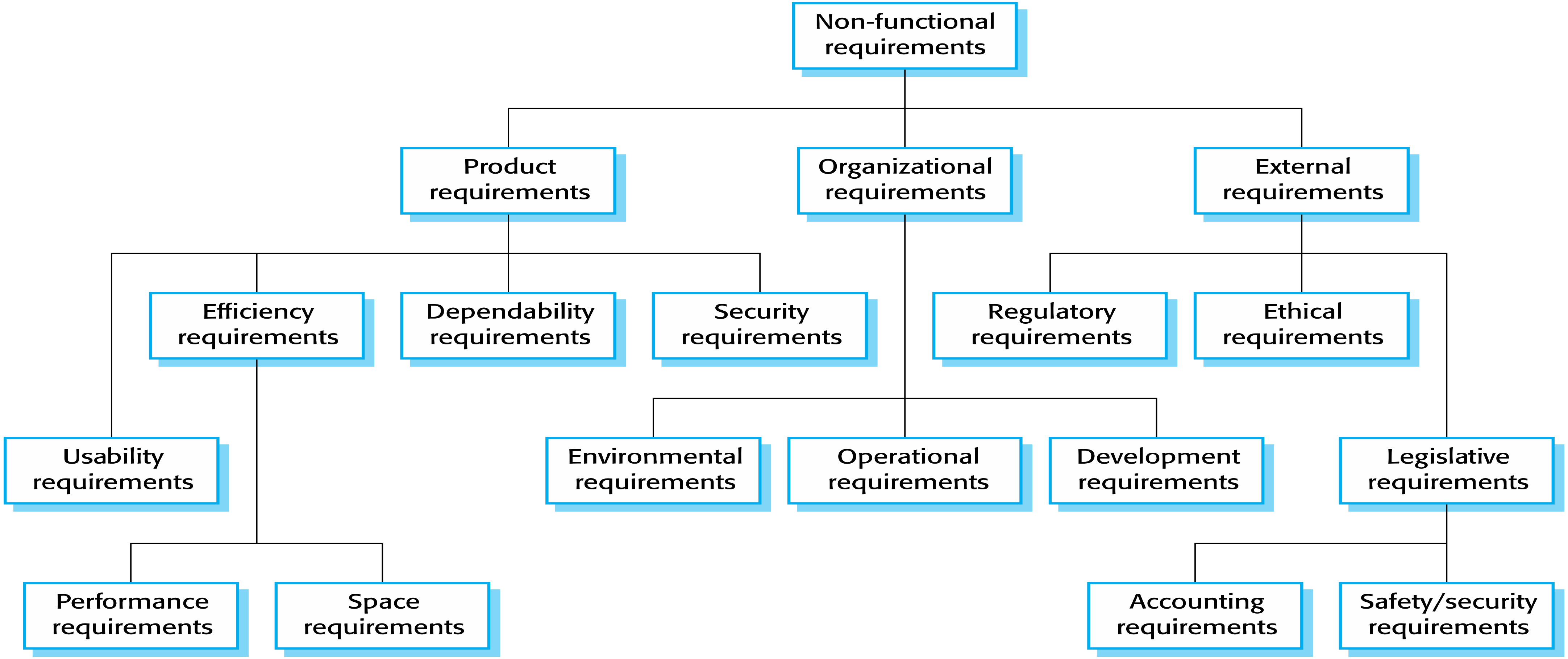
- **Domain requirements**
  – Constraints on the system from the domain of operation



Levels of Requirements

6

Types of nonfunctional 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

Non-functional requirements implementation

- One that can affect whole architecture example - Security

Non-functional classifications

- Architectural 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.

Requirement engineering

- Requirement engineering is the process of creating a specification for what an organization needs that includes gathering, understanding and documenting requirements.
- The process of understanding a customer's needs and transforming them into requirements for development.
- The process includes requirements eliciting, gathering, analysis, modeling, validation, specification, scoping, naming and sequencing requirements.

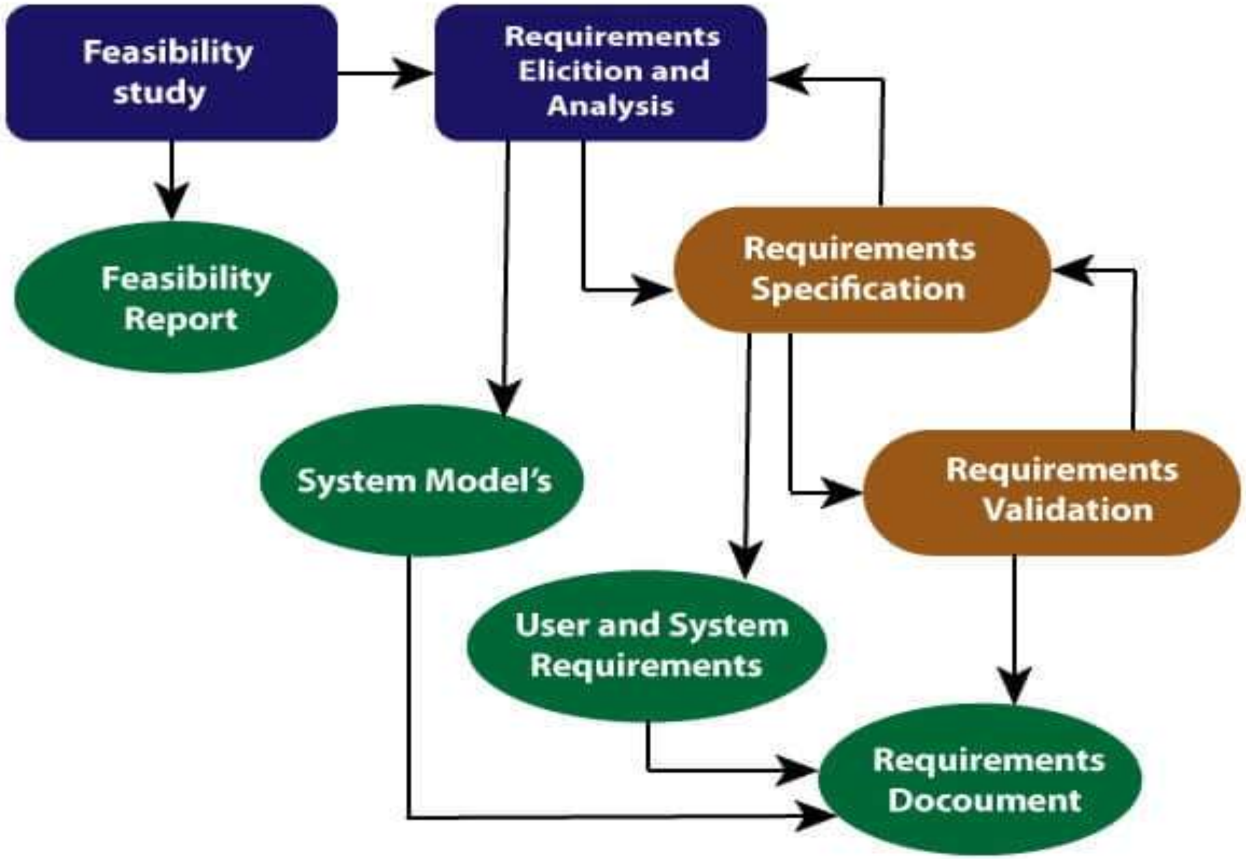
- In software engineering, it can be used to capture functional or non-functional requirements
- **in the form of use cases or user stories.**
- Requirements are often captured in documents called requirement specifications(**SRS**)
- That describe **what a system** should do **(functions)** and any constraints on **how it does** those things **(non-functional).**

- Requirement engineers work closely with clients to gather their needs in order to create specifications that will address these needs.
- They use skills in the English language (reading, writing), information technology (software design) and mathematics (data analysis)

- The **goal of requirement engineering** is
- to
- ***produce a documented set of requirements*** for any given system or product
- which can be used by designers as they build it from scratch or by developers as they modify existing systems.

Requirements engineering processes

- The processes used for RE vary widely depending on the application domain, the people involved and the organisation developing the requirements.
- However, there are a number of generic activities common to all processes
- Requirement **Inception/Feasibility study**
  - Requirements **elicitation;**
  - Requirements **analysis;**
  - Requirements **validation;**
  - Requirements **management.**



Requirement Engineering Process

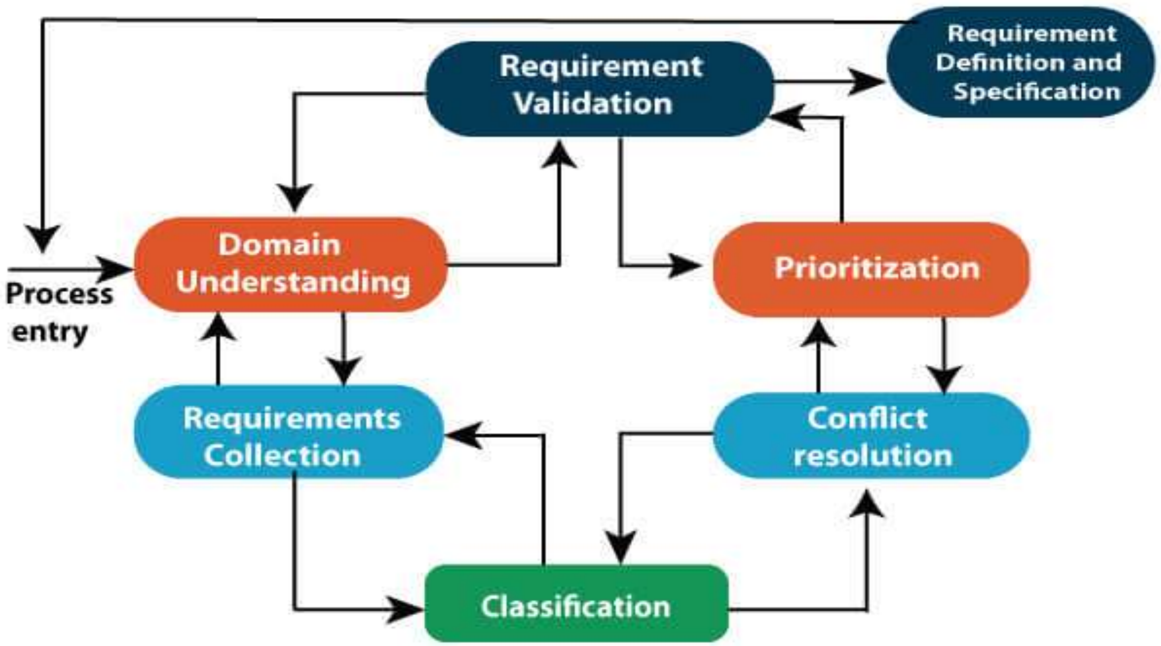
Feasibility Study

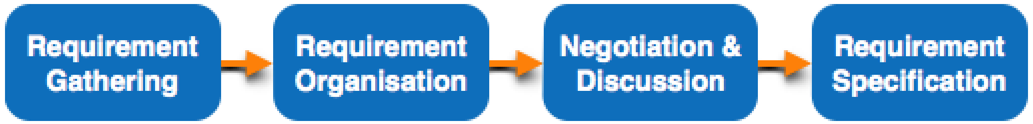
- The main aim of a feasibility study is to create reasons for the development of the software that the users accept, that is flexible enough and open to changes, and abide by the standards chosen for software development and maintenance.
- There are several types of feasibility. They are:
- **Technical Feasibility:** The **current technologies are evaluated** using technical feasibility, and they are necessary to achieve the requirements of the customer within the given time and budget.
- **Operational Feasibility:** The assessment of the range for software in which the required software performs a series of levels to **solve the problems in business and the customer's requirements.**
- **Economic Feasibility:** If the required software can **generate profits** in the area of finance is decided by economic feasibility.

Elicitation

- This process is also called requirements gathering. If there are any existing processes available and with the help of customers, requirements gathering is done. Elicitation of requirements is the starting step of requirements analysis. Inconsistencies, omissions, defects, etc., can be identified by analyzing the requirements. Requirements are described in relationship terms to resolve if there are any conflicts. There are a few challenges with respect to the elicitation of requirements and analysis. They are:
- Involvement of all the right people and only the right people.
- The stakeholders not knowing what they need.
- The requirements are expressed in terms of stakeholders.
- The requirements of the stakeholders may be conflicting.
- The changes in requirements during the process of analysis.
- Several factors influence the requirements of the system in organization and politics.

Elicitation and Analysis Process





Elaboration

- info obtain during inception n elicitation is expanded and refined .
- RE activity focuses on developing a model of S/W functions , features , constraints.
- User scenarios , they are parsed into class

  Diagrams n other UML diagrams r produced.

  Analysis model is one produced at end

Validation of Software Requirements

- After the development of the specification of requirements, the requirements laid down in the document are validated. There may be requirements from the users that are illegal or which cannot be accomplished, or the experts can misunderstand the needs. The requirements must satisfy the following conditions:
- If the requirements can be implemented practically.
- If the requirements are correct and they are according to the software functionality.
- If there are any confusions.
- If the requirements are full.
- If the requirements can be described.
- There are several techniques for the validation of requirements. They are:
    - Inspection of requirements or reviews of requirements.
- This includes an analysis of the requirements systematically and manually.
    - Prototyping
- The requirements of the model are checked by using a model that is executable.
    - Generation of test case
- The testability is checked for the requirements by the development of tests.
    - Automated consistency analysis
- The consistency of the descriptions of requirements is checked.

Management of Software Requirements

- The process of **managing the requirements that keep changing** during the process of requirements engineering and development of the system is called management of software requirement.
- During the process of software management, there are new requirements with the change in the needs of business, and there is the development of a better understanding of the system.
- During the process of development, the requirements priority changes from different views. During the process of development, the technical and business environment of the system changes.

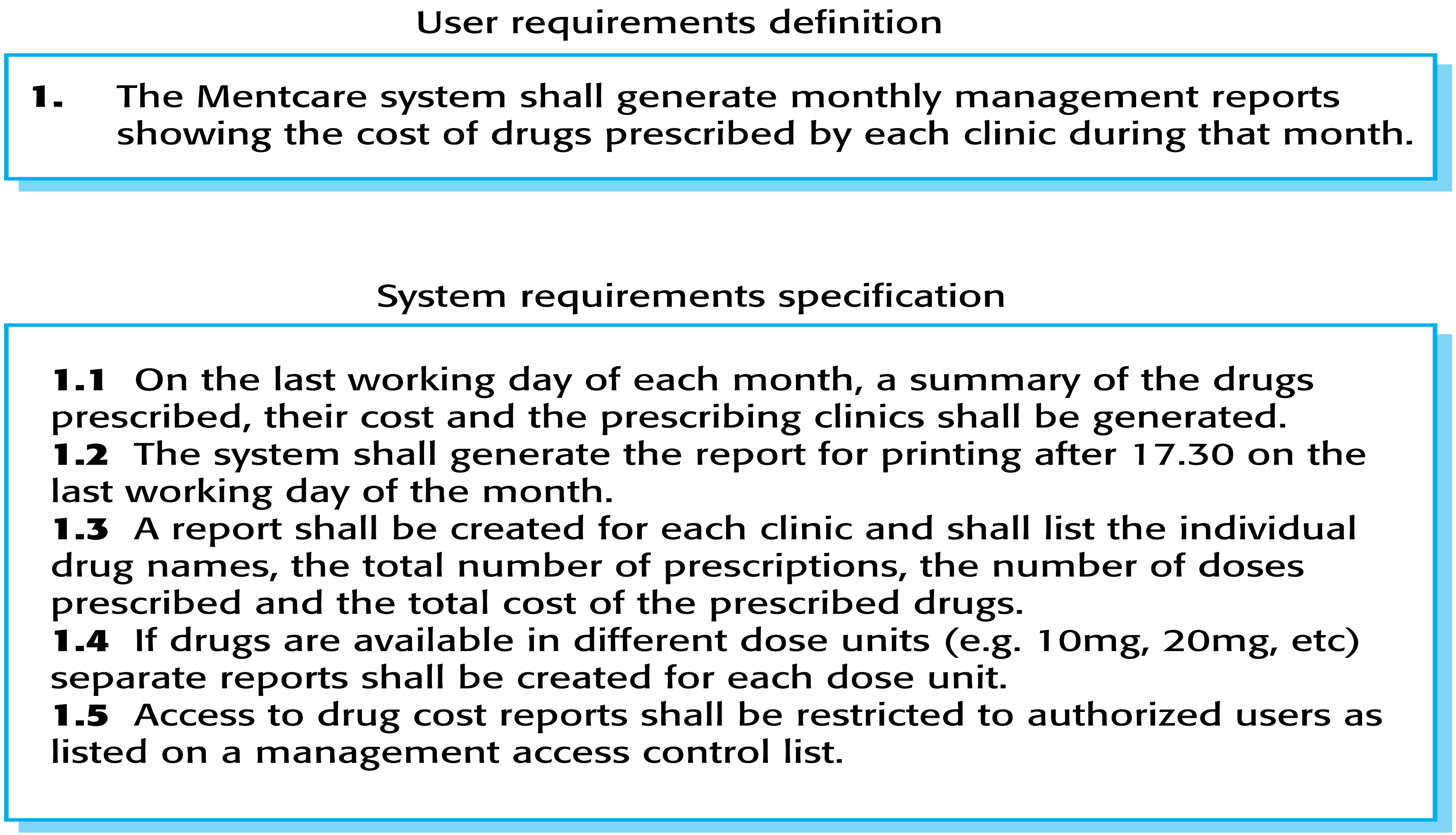
Specification of Software Requirements

- A document consisting of requirements that are collected from various sources like the requirements from customers expressed in an ordinary language and created by the software analyst is called a specification document for software requirements. The analyst understands the customers’ requirements in ordinary language and converts them into a technical language that the development team can easily understand. Several models are used during the process of specification of software requirements like Entity-Relationship diagrams (ER diagrams), data flow diagrams (DFD), data dictionaries, function decomposition diagrams (FDD), etc.
- **Data Flow Diagrams (DFD):** The modeling of requirements can be done using Data Flow Diagrams (DFD). The flow of data within the system can be seen by using data flow diagrams (DFD). The system here can be a company, an organization, a hardware system in a computer, a software system in a computer, a set of procedures or a combination of everything. The data flow diagrams (DFD) are also called bubble charts or data flow graph.
- **Data Dictionaries:** The data defined using a data flow diagram (DFD) is stored as information in the form of repositories called data dictionaries. The customer’s data items must be defined by the data dictionaries at the stage of requirements gathering to make sure that the customers and developers use the same methodologies and definitions.
- **Entity Relationship Diagrams:** One of the other tools used for the specification of requirements is entity-relationship diagrams. It is also called as ER diagrams. The detailed representation of logic for the organization is done using entity relationship diagrams (ER diagrams). They make use of three types of constructs: relationships, entities of data, and the attributes associated with them.

Case Study

- Example of Mentcare System

User and system requirements



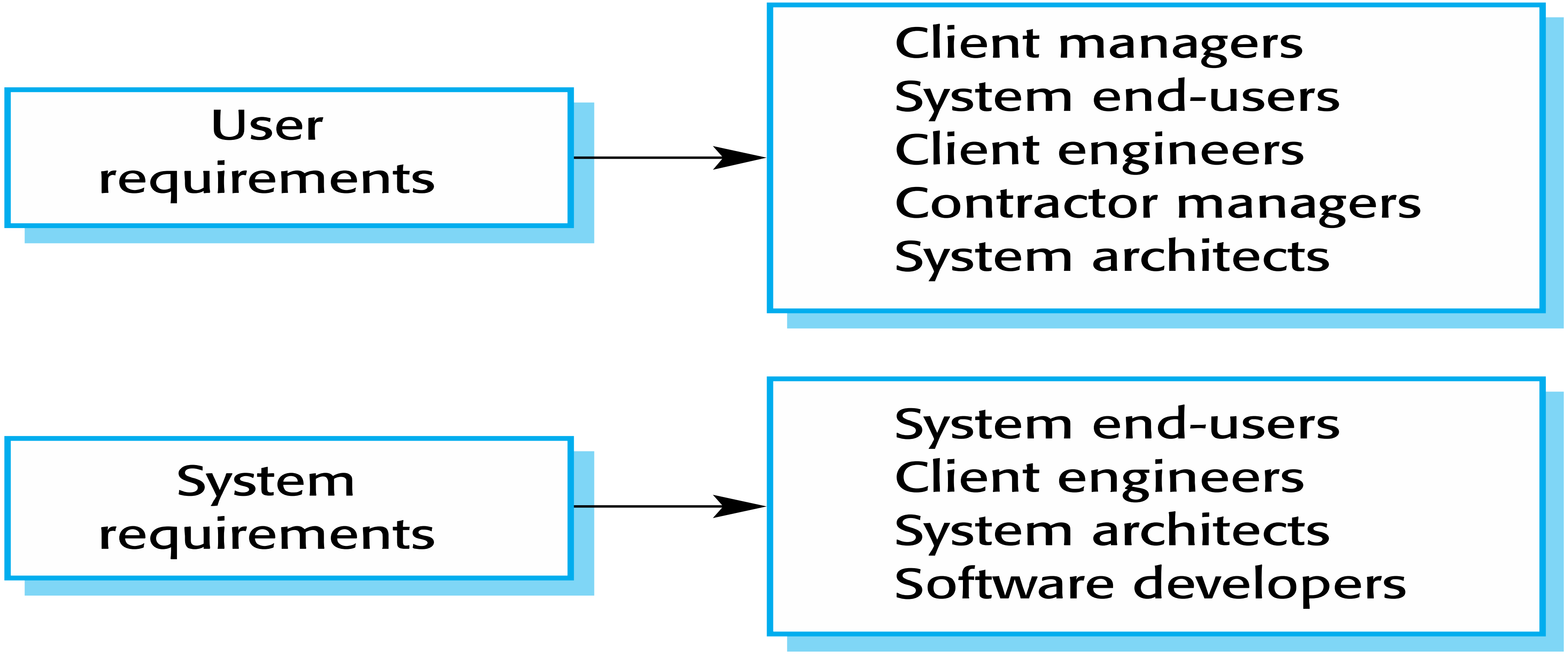
User requirements definition

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 –Who?



System stakeholders

- Who has a legitimate interest
- Stakeholder types
    - End users
    - System managers
    - System owners
    - External stakeholders

Stakeholders in the Mentcare system

- **Patients**
- **Doctors**
- **Nurses**
- **Medical receptionists**
- **IT staff**
- **A medical ethics manager**
- **Health care managers**
- **Medical records staff**

Mentcare system: functional requirements

- 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.

Examples of nonfunctional requirements in the Mentcare system

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.

SOFTWARE REQUIREMENTS SPECIFICATION

- **A Software Requirements Specification [SRS]** is a document that is created when a detailed description of all aspects of the software to be built must be specified before the project is to commence.

- When a software is to be developed by a third party, when a lack of specification would create severe business issues, or when a system is extremely complex or business critical, an SRS is prepared.

- Karl Wiegers has developed a worthwhile template at **www.processimpact.com/process_assets/srs_template.doc**

SOFTWARE REQUIREMENTS SPECIFICATION

SRS TEMPLATE

Table of Contents :

1. **Introduction**
   1.1 Purpose
   1.2 Document Conventions
   1.3 Intended Audience and Reading Suggestions
   1.4 Project Scope
   1.5 References
2. **Overall Description**
   2.1 Product Perspective
   2.2 Product Features
   2.3 User Classes and Characteristics
   2.4 Operating Environment

SOFTWARE REQUIREMENTS SPECIFICATION

SRS TEMPLATE … contd.

2.5 Design and Implementation Constraints
2.6 User Documentation
2.7 Assumptions and Dependencies

3. **System Features**

3.1 System Feature 1
3.2 System Feature 2 (and so on)

4. **External Interface Requirements**

4.1 User Interfaces
4.2 Hardware Interfaces
4.3 Software Interfaces
4.4 Communications Interfaces

SOFTWARE REQUIREMENTS SPECIFICATION

SRS TEMPLATE … contd.

5. **Nonfunctional Requirements**

   5.1 Performance Requirements
   5.2 Safety Requirements
   5.3 Security Requirements
   5.4 Software Quality Attributes

6. **Other Requirements**

   Appendix A : Glossary
   Appendix B : Analysis Model
   Appendix C : Issues List