Software Engineering

Dr. Fatma ElSayed

Computer Science Department
fatma.elsayed@fci.bu.edu.eg

Course Information

Contents

- Introduction to Software Engineering
- Software Processes
- Requirements Engineering
- System Modelling & Design
- System Modelling & Design
- System Architecture
- Software Testing Strategies
- Software Testing Techniques
- Technical Metrics for Software

Course Information

Contents

- Introduction to Software Engineering
- Software Processes
- Requirements Engineering
- System Modelling & Design
- System Modelling & Design
- System Architecture
- Software Testing Strategies
- Software Testing Techniques
- Technical Metrics for Software

Chapter 4: Requirements Engineering

Requirements Engineering

1. **State the different software process models?**

2. **State three reasons when to use waterfall model.**

3. **What is software prototyping and what is included?**

4. **State the incremental delivery advantages?**

Requirements Engineering

- **Requirement:** A function, constraint, or other property that the system must provide to fill the needs of the system’s intended user(s).
- **Engineering:** implies that systematic and repeatable techniques should be used.
- **Requirement Engineering** means that requirements for a product are defined, managed, and tested systematically.
    - The process of establishing the **services** that a customer requires from a system and the **constraints** under which it operates and is developed.

Requirements Completeness and Consistency

Functional requirements specification should be both **complete** and **consistent**.

1. **Complete**

   means that all services required by the user should be defined.

2. **Consistent**

   means that requirements should not have contradictory definitions

- For large, complex systems, it is practically impossible to achieve requirements consistency and completeness **Reasons:**
  - It is easy to make mistakes and omissions when writing specifications.
  - There are many stakeholders in a large system.

System Stakeholders

- Any person or organization who is affected by the system which is being developed.
- **Stakeholder types**

1. System owners *(those who are paying for the work),*
2. End users *(people who will use the system)*
3. System managers
4. Project groups that are involved in developing the system *(e.g., systems engineers, software engineers, training personnel, testers, etc.).*
5. External stakeholders *(marketing people, government,…)*

System Stakeholders

- **Stakeholder types**

1. System owners
    -
    -
    -

Each one of them has **different view of the system**

Recognize multiple points of view

- **Marketing group** concerned about features and functions to excite the potential market. To sell easily in the market.
- **Business manager** concerned about features built within budget and will be ready to meet the market.
- **End user** – Easy to learn and use.
- **Support engineer** – Maintainability of software.

Stakeholders in the MHC-PMS

*Mental Health Care-Patient Management System (**MHC-PMS**)*

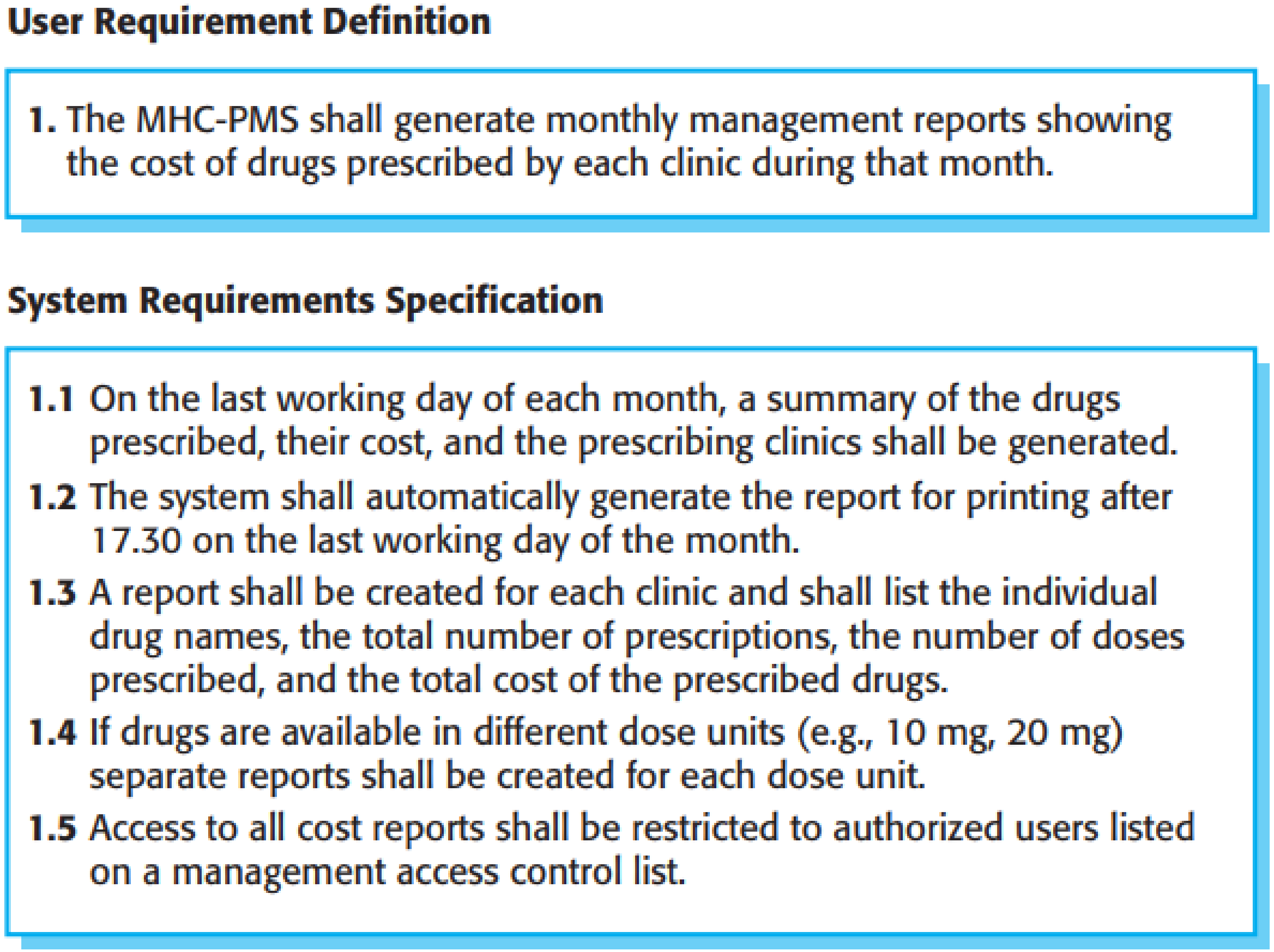
- **Patients** whose information is recorded in the system.
- **Doctors** who are responsible for assessing and treating patients.
- **Nurses** who coordinate the consultations with doctors and administer some treatments.
- **Medical receptionists** who manage patients' appointments.
- **IT staff** who are responsible for installing and maintaining the system.
- **A medical ethics manager** who must ensure that the system meets current ethical guidelines for patient care.

Types of Requirement

- **User requirements**
  - User requirements are statements, in a natural language plus diagrams, of what services the system is expected to provide to system users and the constraints under which it must operate.

- **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 Requirement Definition

1. The MHC-PMS 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 automatically 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., 10 mg, 20 mg) separate reports shall be created for each dose unit.

1.5 Access to all cost reports shall be restricted to authorized users listed on a management access control list.

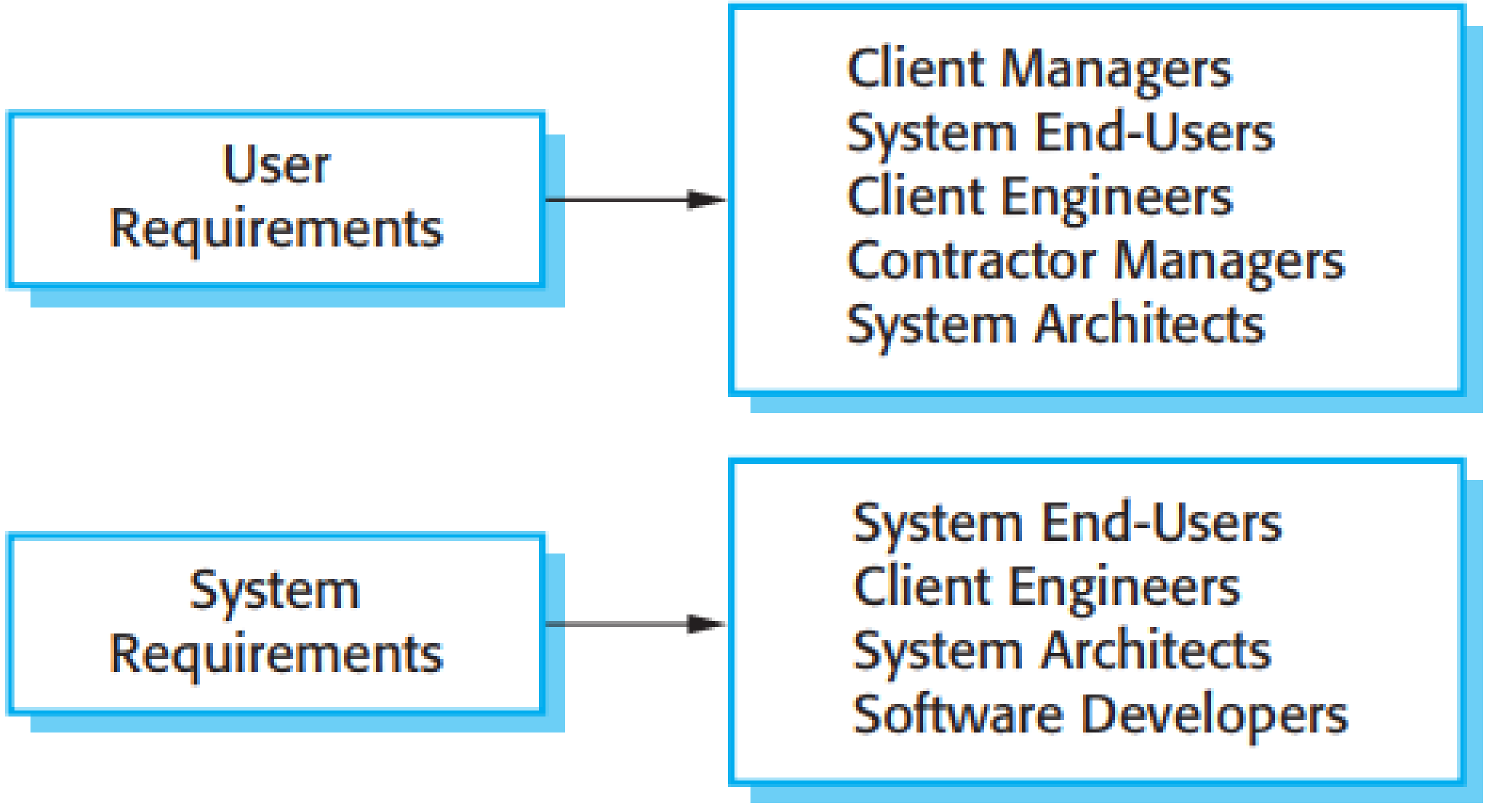
Business Requirement Specification (BRS)

it defines **"what"** the system should do from a business perspective.

System Requirement Specification (SRS)

specifies **"how"** the system will achieve those goals technically.

Requirements Readers



User Requirements

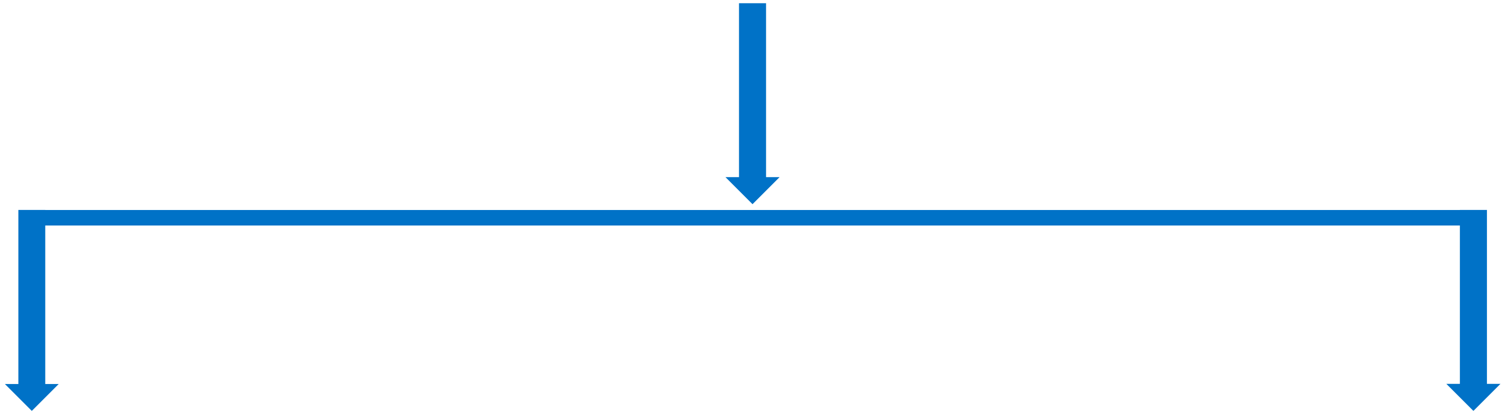
Client Managers
System End-Users
Client Engineers
Contractor Managers
System Architects

System Requirements

System End-Users
Client Engineers
System Architects
Software Developers

Requirements Classification

Software system requirements are often classified as:



Functional

- *Something the system must do* ***(what)***
- Generally phrased with noun + verb constructions

e.g. The system prints an invoice

Non-Functional

- *Describe how the system works* ***(how)***
- Are structured with adverb

e.g. The system prints invoices quickly

Functional Requirements

- They are sometimes called **behavioral / operational requirements** because they specify the inputs (stimuli) to the system, the outputs (responses) from the system, & behavioral relationships between them.
- Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.
- Depend on the type of software, expected users, and the type of system where the software is used.
- May state what the system should not do.

MHC-PMS: Functional Requirements

Requirements for the MHC-PMS system, used to maintain information about patients receiving treatment for mental health problems:

1. A user shall be able to search the appointment lists for all clinics.

2. The system shall generate each day, for each clinic, a list of patients who are expected to attend appointments that day.

3. Each staff member using the system shall be uniquely identified by his or her eight-digit employee number.

MHC-PMS: Functional Requirements

Requirements for the MHC-PMS system, used to maintain information about patients receiving treatment for mental health problems:

1. A user shall be able to **search** the appointment lists for all clinics.

2. The system shall genera ach day, for each clinic, a list of patients who are expected ments that day.

3. Each staff memb em shall be uniquely identified by his or her eight-di nber.

What do you think is meant by search

Functional Requirements Imprecision

- Problems arise when functional requirements are not accurately 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.

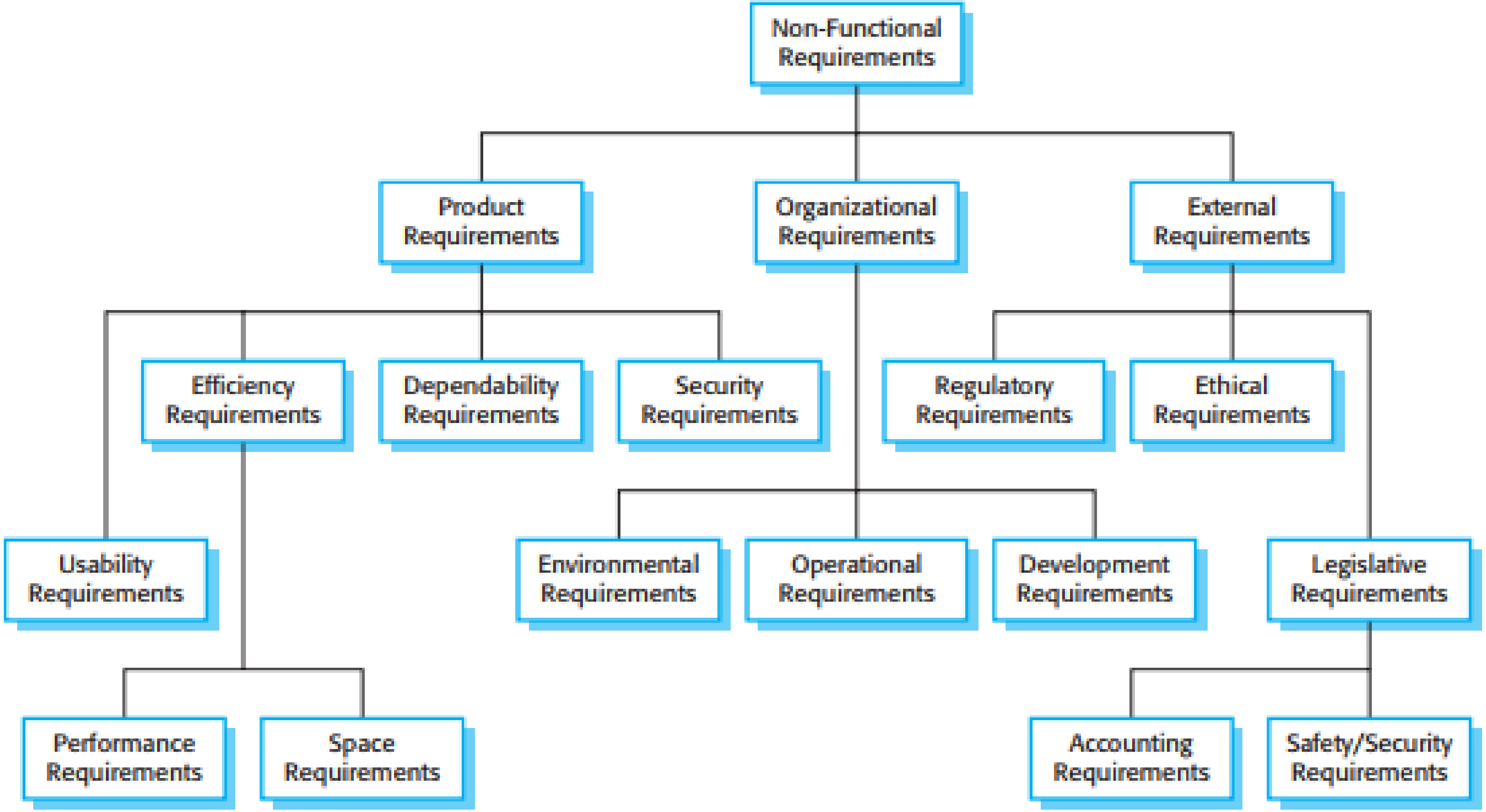
Non-functional Requirements

- These define system/software **properties** (such as reliability and safety), and **constraints** (such as response time), constraints on the development process, I/O device capability, system representations, standards, etc.

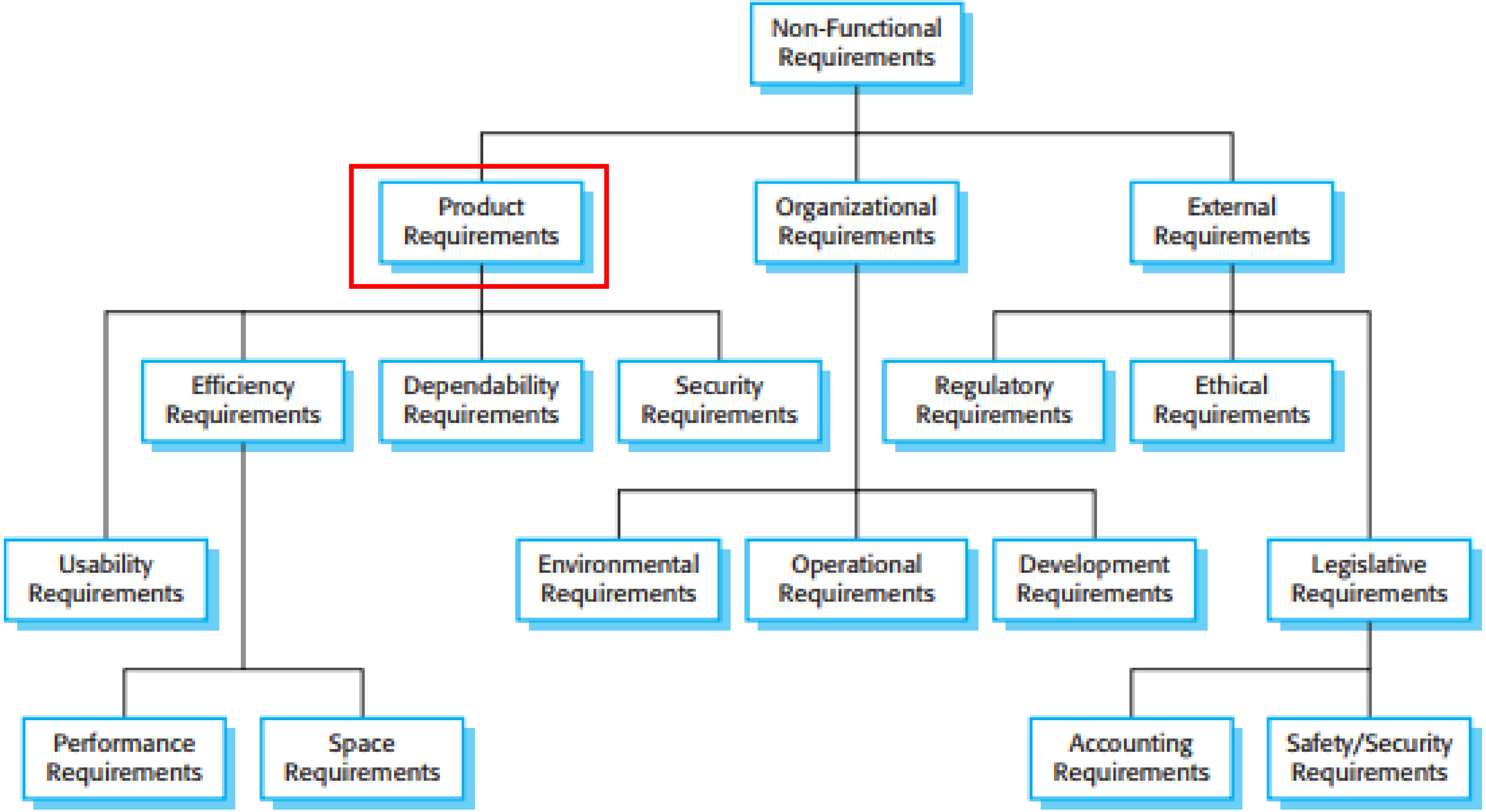
- Non-functional requirements may be **more critical** than functional requirements. If these are not met, the system may be useless.

- Often apply to the entire system/software a **whole** (rather than individual features or services)

Types of non-functional Requirements



Types of non-functional Requirements



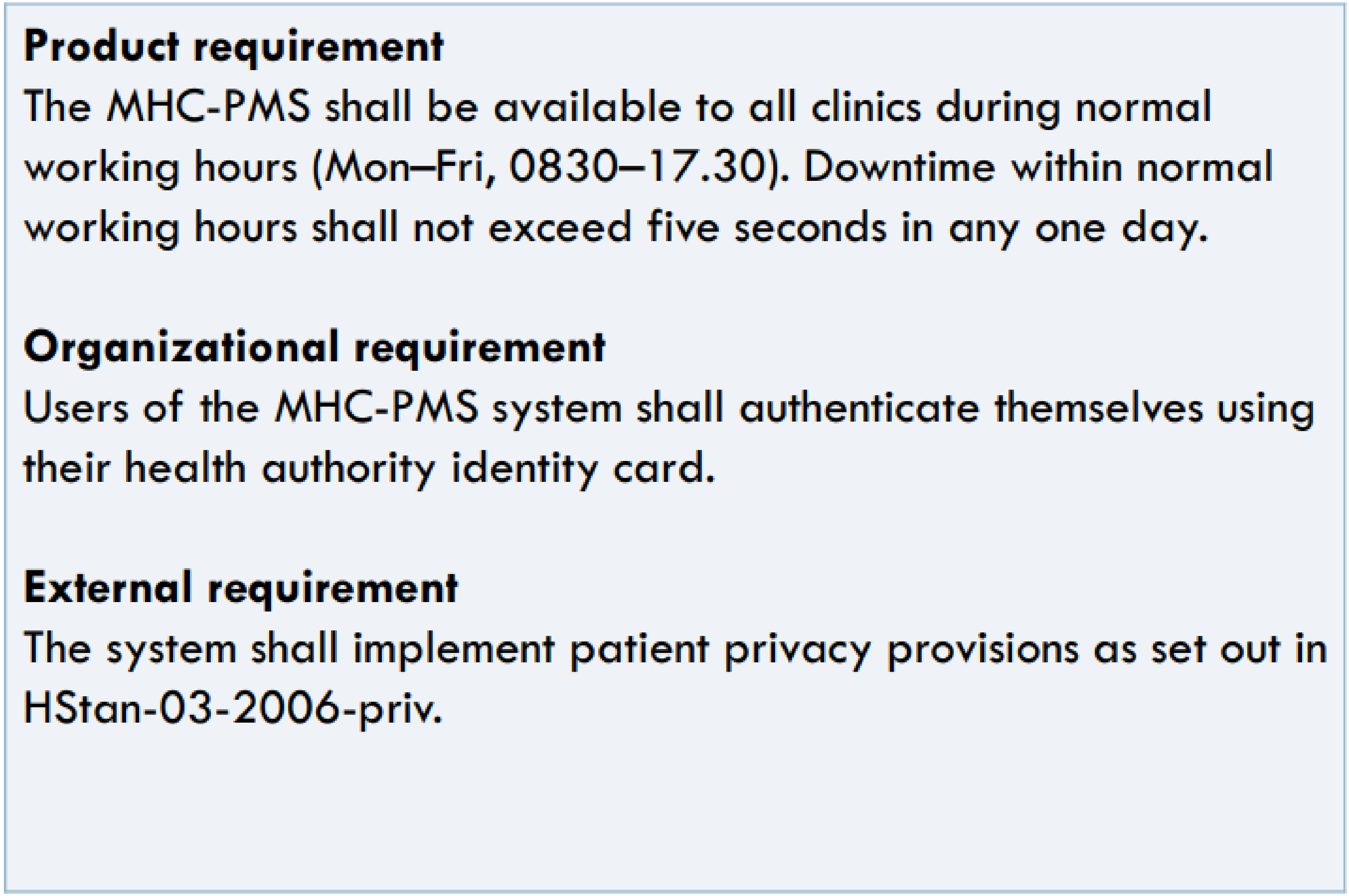
Non-functional Requirements Classifications

1. **Product requirements** (quality attributes)
   - Requirements which specify that the delivered product must behave in a particular way e.g. execution speed, reliability, etc.

2. **Organizational requirements**
   - Requirements which are broad system requirements derived from policies and procedures in the customer’s and developer’s organization.

3. **External requirements**
   - Requirements arise from factors that are external to the system and its development process e.g. interoperability requirements, legislative requirements, etc.

Examples of non-functional Requirements: 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.

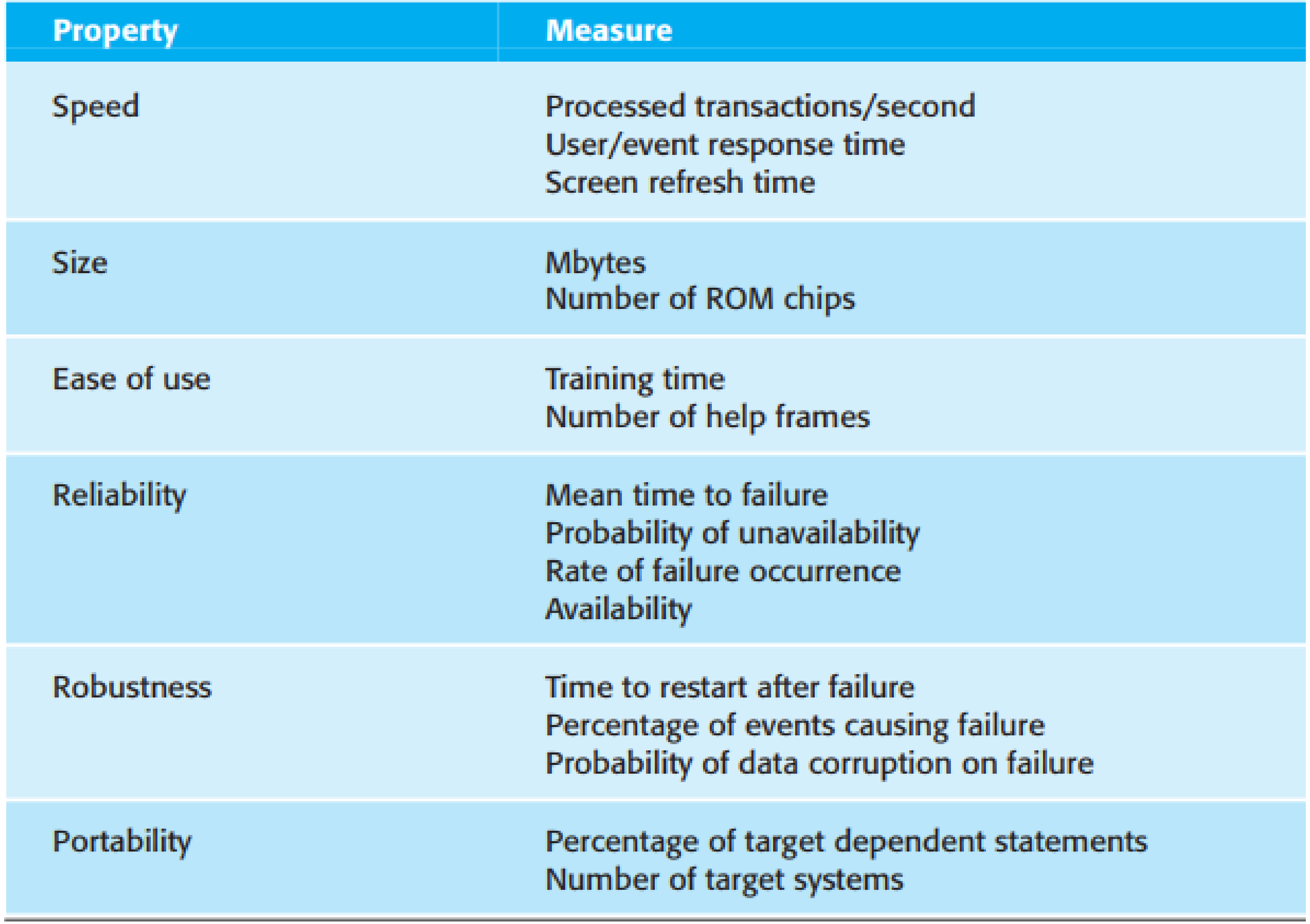
Organizational requirement

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.

Metrics for Specifying nonfunctional Requirements



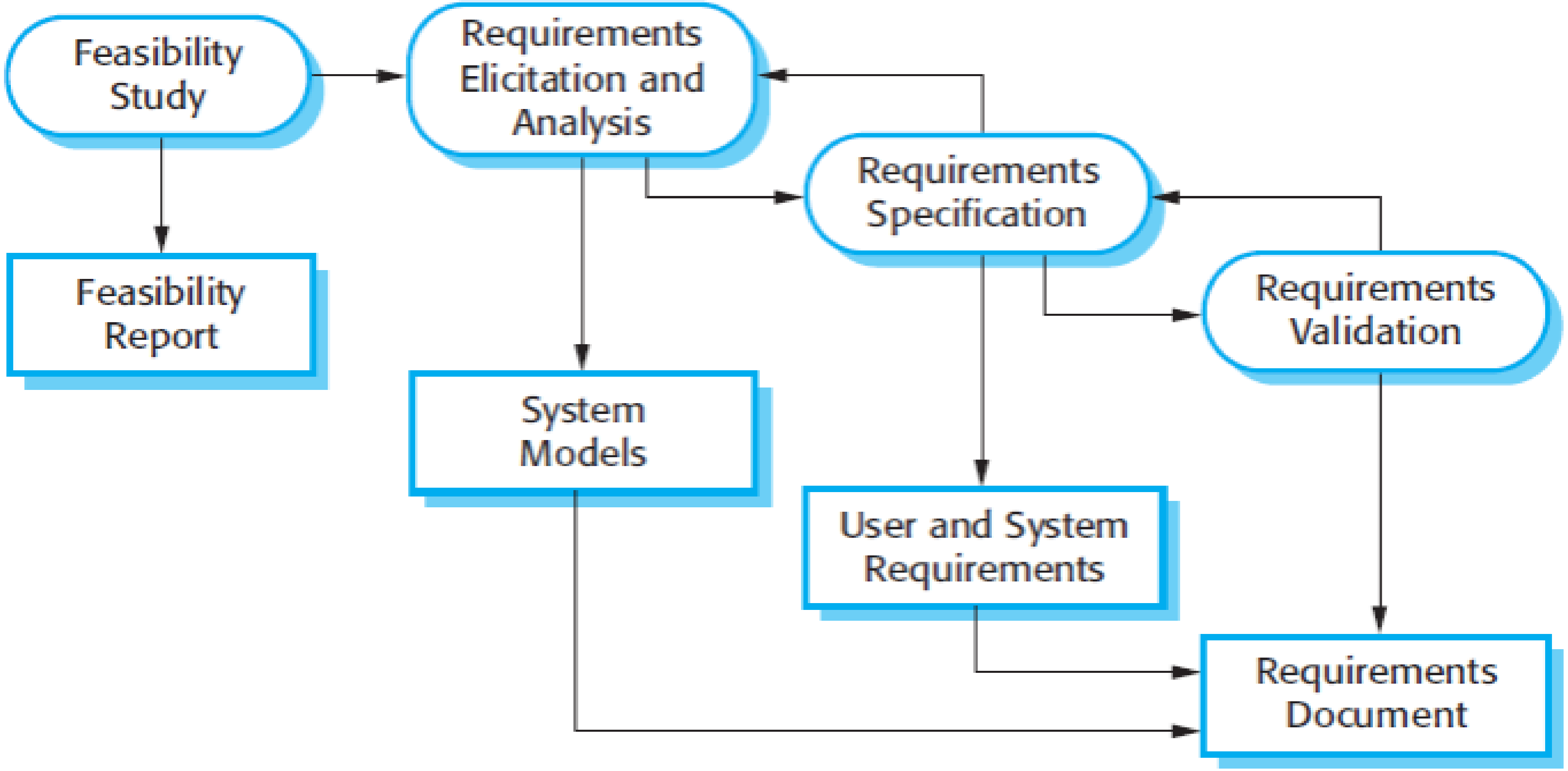
Property	Measure
Speed	Processed transactions/second User/event response time Screen refresh time
Size	Mbytes Number of ROM chips
Ease of use	Training time Number of help frames
Reliability	Mean time to failure Probability of unavailability Rate of failure occurrence Availability
Robustness	Time to restart after failure Percentage of events causing failure Probability of data corruption on failure
Portability	Percentage of target dependent statements Number of target systems

You can measure these characteristics when the system is being tested to check whether or not the system has met its non-functional requirements.

Requirements Engineering Processes

Requirements Engineering Processes

- Requirements engineering is an iterative process in which the activities are interleaved.



Requirements Engineering Processes

- **Feasibility Study:** decides whether or not the proposed system is worthwhile and can be engineered using current software and hardware technologies.
  - Checks if it can be developed within existing budget constraints.
  - Should be relatively cheap and quick.
  - Should inform the decision of whether or not to go ahead with a more detailed analysis.
  - If the system can be integrated with other systems that are used.
- **Requirements Elicitation and Analysis:** Draw out the requirements from stakeholders.
  - Involves technical staff working with customers to find out about the application domain, the services that the system should provide, the required performance of the system, and hardware constraints.

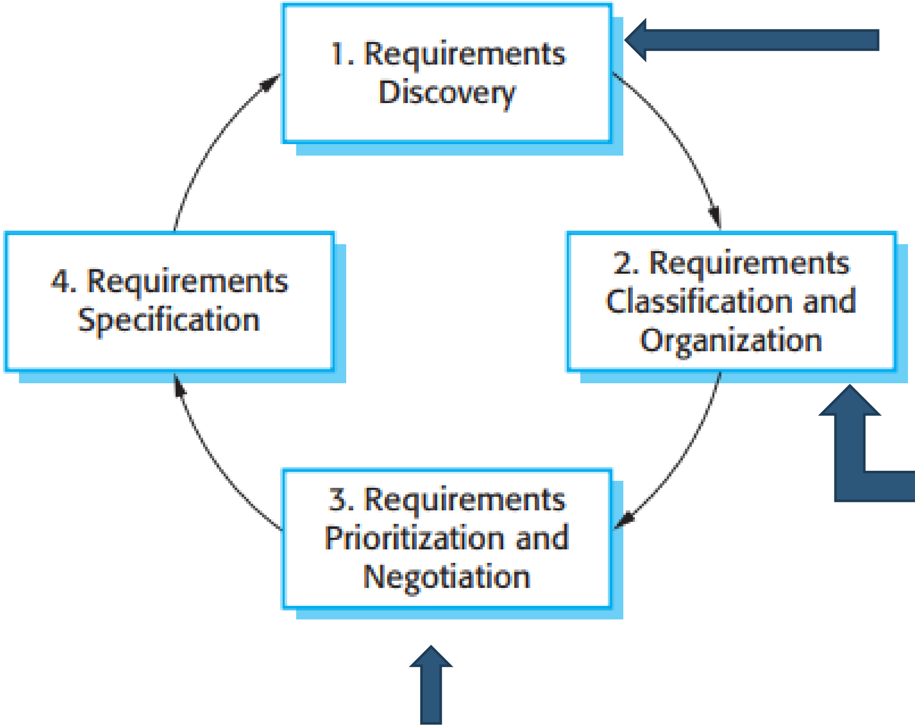
Requirements Engineering Processes

- **Requirements Specification:**
  - Requirements are documented.

- **Requirements Validation:**
  - Review the requirement specification for errors, ambiguities, omissions, and conflicts.

- **System Requirements Document ☺**

1-The Requirements Elicitation and Analysis Process



1. Requirements Discovery
2. Requirements Classification and Organization
3. Requirements Prioritization and Negotiation
4. Requirements Specification

- Interacting with stakeholders to discover their requirements.
- Domain requirements are also discovered at this stage.

- Grouped related requirements and organized them into clusters.
- Relations among requirements identified.
- Requirements reviewed for correctness.

Prioritizing requirements based on customer and resolving requirements conflicts. Negotiation about requirements; project cost and project timeline.

Requirements Discovery: Interviewing

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

Types of interview

- ***Closed*** interviews based on a pre-determined list of questions
- ***Open*** interviews where various issues are explored with stakeholders.

- Normally a **mix** of closed and open-ended interviewing.

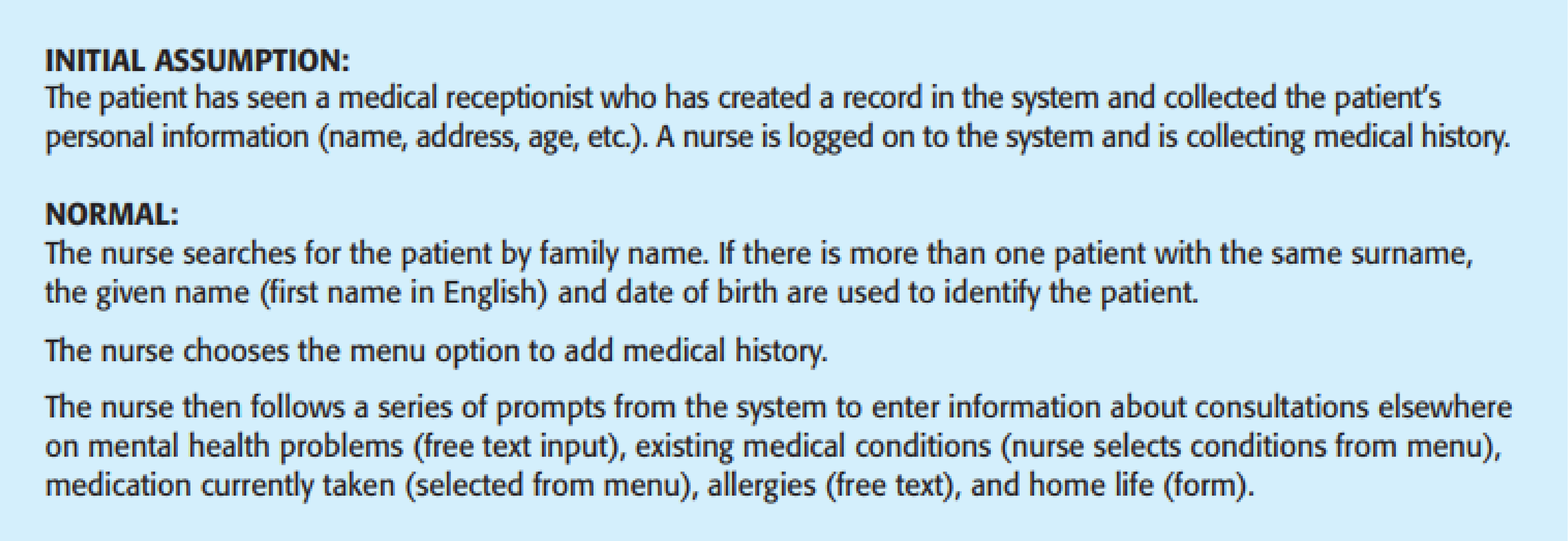
Effective interviewing

- Be **open-minded**, avoid pre-conceived ideas about the requirements and are willing to listen to stakeholders.
- Prompt the interviewee to get **discussions** going using questions, a requirements proposal, or by working together on a prototype system.

Requirements Discovery: Scenarios

- Scenarios are real-life examples of how a system can be used.
- Scenarios are a description of how a system may be used for a particular task.
- **Scenarios should include**
  - A description of the starting situation;
  - A description of the normal flow of events;
  - A description of what can go wrong;
  - Information about other concurrent activities;
  - A description of the state when the scenario finishes.

Scenario for Collecting Medical History: MHC-PMS



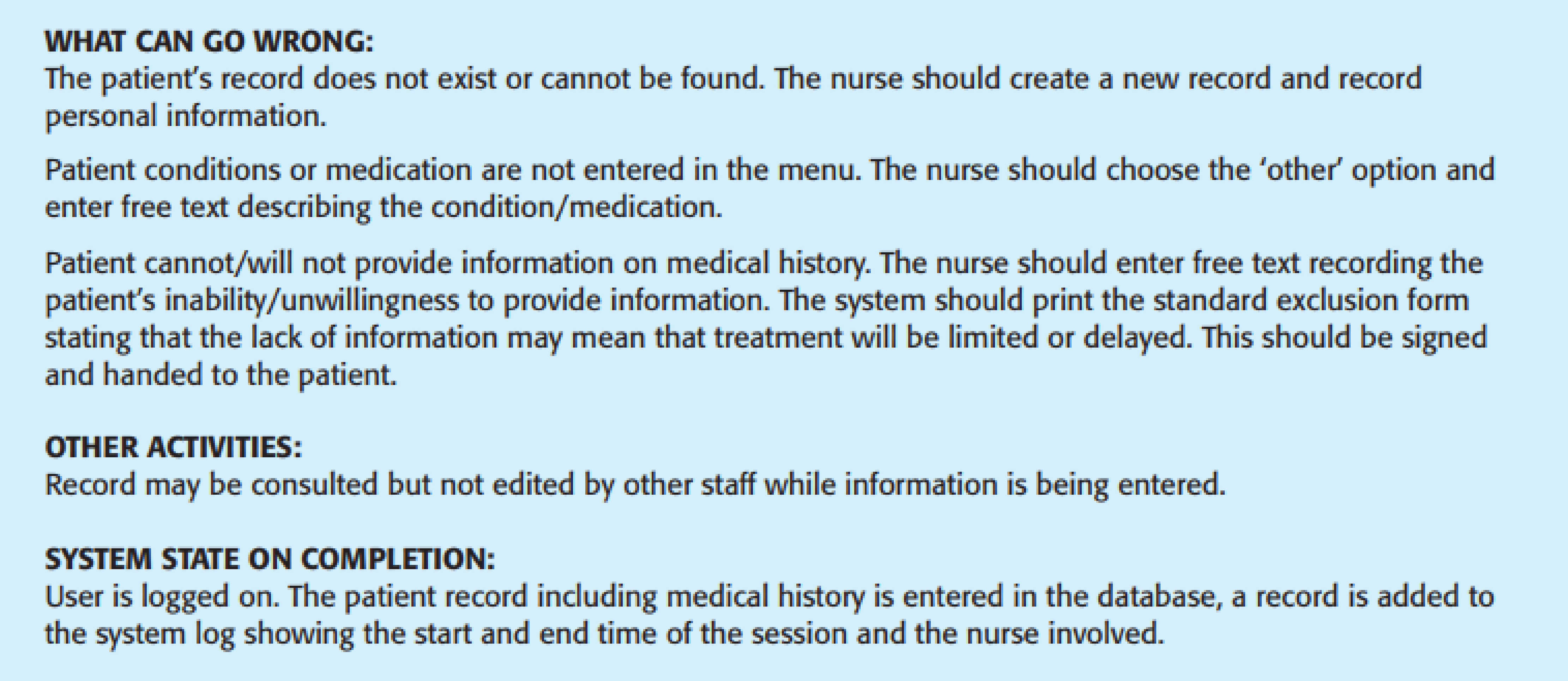
INITIAL ASSUMPTION:
The patient has seen a medical receptionist who has created a record in the system and collected the patient's personal information (name, address, age, etc.). A nurse is logged on to the system and is collecting medical history.

NORMAL:
The nurse searches for the patient by family name. If there is more than one patient with the same surname, the given name (first name in English) and date of birth are used to identify the patient.

The nurse chooses the menu option to add medical history.

The nurse then follows a series of prompts from the system to enter information about consultations elsewhere on mental health problems (free text input), existing medical conditions (nurse selects conditions from menu), medication currently taken (selected from menu), allergies (free text), and home life (form).

Scenario for Collecting Medical History: MHC-PMS



WHAT CAN GO WRONG:
The patient's record does not exist or cannot be found. The nurse should create a new record and record personal information.

Patient conditions or medication are not entered in the menu. The nurse should choose the 'other' option and enter free text describing the condition/medication.

Patient cannot/will not provide information on medical history. The nurse should enter free text recording the patient's inability/unwillingness to provide information. The system should print the standard exclusion form stating that the lack of information may mean that treatment will be limited or delayed. This should be signed and handed to the patient.

OTHER ACTIVITIES:
Record may be consulted but not edited by other staff while information is being entered.

SYSTEM STATE ON COMPLETION:
User is logged on. The patient record including medical history is entered in the database, a record is added to the system log showing the start and end time of the session and the nurse involved.

Requirements Discovery: Ethnography

- An analyst **immerses** himself or herself in the working environment where the system will be used.
- Spends a considerable time **observing** and analyzing how people actually work.
- People often find it very difficult to articulate details of their work. They understand their own work but may not understand its relationship to other work in the organization.
- Helps discover implicit system requirements (Social and organizational factors) that reflect the actual ways that people work.
- Ethnography is effective for understanding existing processes but cannot identify new features that should be added to a system.

Requirements Discovery: Use Cases

In Detail next Lecture ISA

2-Requirements Specification

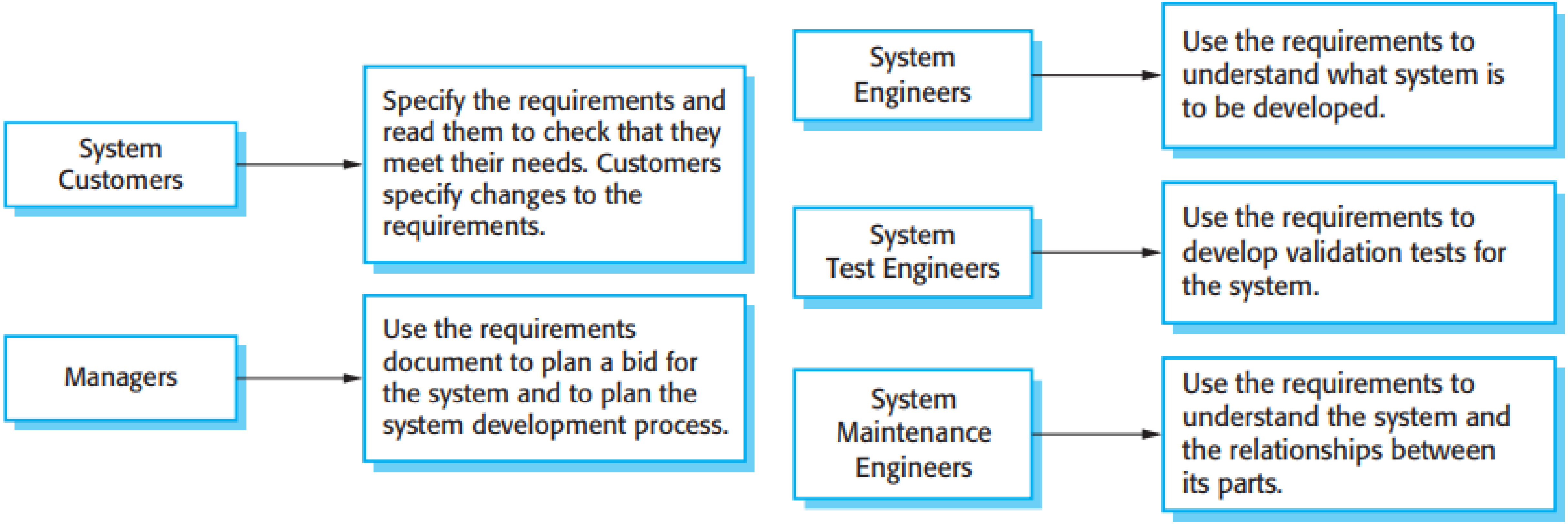
- The process of writing down the user and system requirements in a requirements document.
- Ideally, the user and system requirements should be clear, unambiguous, easy to understand, complete, and consistent.
- It is used by software engineers as the starting point for the system design.
- The requirements may be part of a contract for the system development and should therefore be a complete and detailed specification of the whole system.

Software Requirements Document

- The software requirements document *(sometimes called the software requirements specification or SRS)* is the official statement of what is required of the system developers.
- It should include both a definition of user requirements and a specification of the system requirements.
- If there are a **large** number of requirements, the detailed system requirements may be presented in a separate document.
- The level of detail in a requirements document depends on the type of system being developed and the development process used.

Users of Software Requirements Document

The requirements document has a diverse set of users



System Customers

Specify the requirements and read them to check that they meet their needs. Customers specify changes to the requirements.

Managers

Use the requirements document to plan a bid for the system and to plan the system development process.

System Engineers

Use the requirements to understand what system is to be developed.

System Test Engineers

Use the requirements to develop validation tests for the system.

System Maintenance Engineers

Use the requirements to understand the system and the relationships between its parts.

3-Requirements Validation

- A process where customers and the system developers *read the requirements document in detail* and check for errors, ambiguities, and inconsistencies. They then negotiate how the identified problems should be solved.

Check for:

- **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.

Requirements Validation Techniques

There are a number of requirements validation techniques that can be used individually or in conjunction with one another, for example:

- **Requirements reviews**
  - Systematic manual analysis of the requirements.
- **Prototyping**
  - Using an executable model of the system to check requirements.

Finally

Task

- What are the ways of writing SRS (SRD)?
- What is the typical structure of the SRS?

Example

- Software Requirements Specification (SRS) Book E-Commerce System (BECS):

https://www.cse.msu.edu/~chengb/RE-491/Papers/SRS-BECS-2007.pdf

Thank You