MODULE – 2
REQUIREMENTS ANALYSIS AND DESIGN

- **Requirements ❼** the descriptions of the services that a system should provide and the constraints on its operation.

- **Requirements Engineering (RE)**
  - The process of finding out, analyzing, documenting and checking the services and constraints of a system.
  - The first stage of the software engineering process.

User Requirements

- High-level abstract requirements
- 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

- Detailed description of what the system should do.
- Detailed descriptions of the software system’s functions, services, and operational constraints.
- System requirements document (sometimes called a functional specification) should define exactly what is to be implemented.
- Often classified as functional or non-functional 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.

Fig: Mental health care patient information system (Mentcare) shows how a user requirement may be expanded into several system requirements

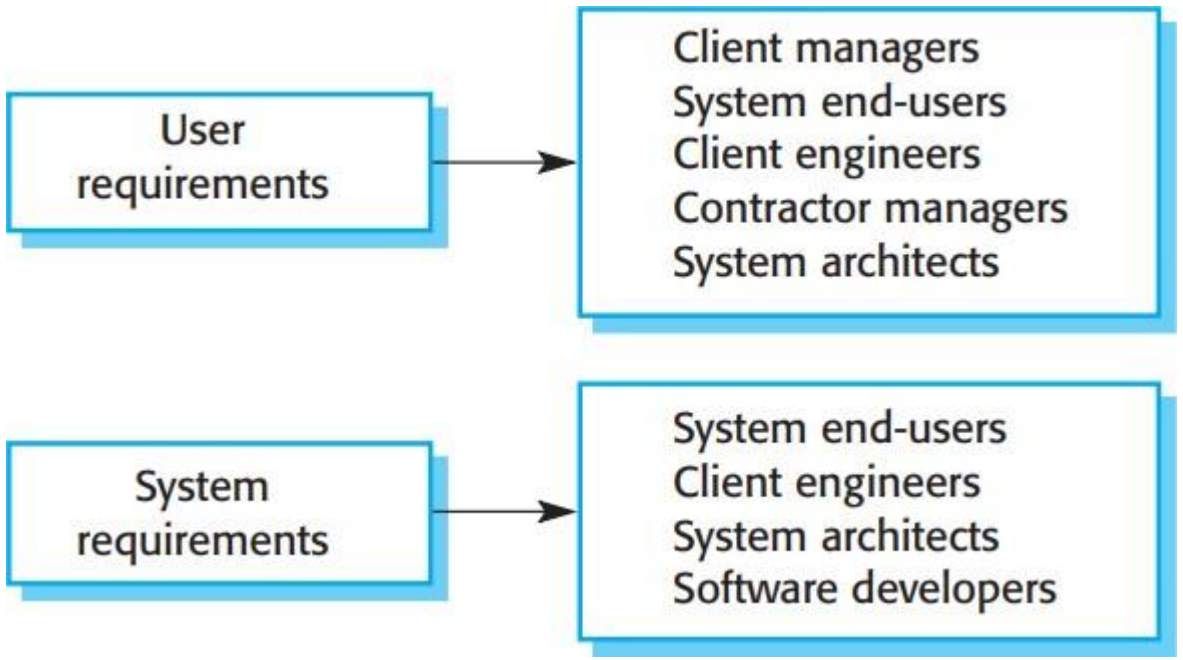


Fig: Readers of different types of requirements specification

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.
- Explicitly state what the system should not do.

Non-functional Requirements

- Constraints on the services or functions offered by the system.
- Include timing constraints, constraints on the development process, and constraints imposed by standards.

Functional Requirements

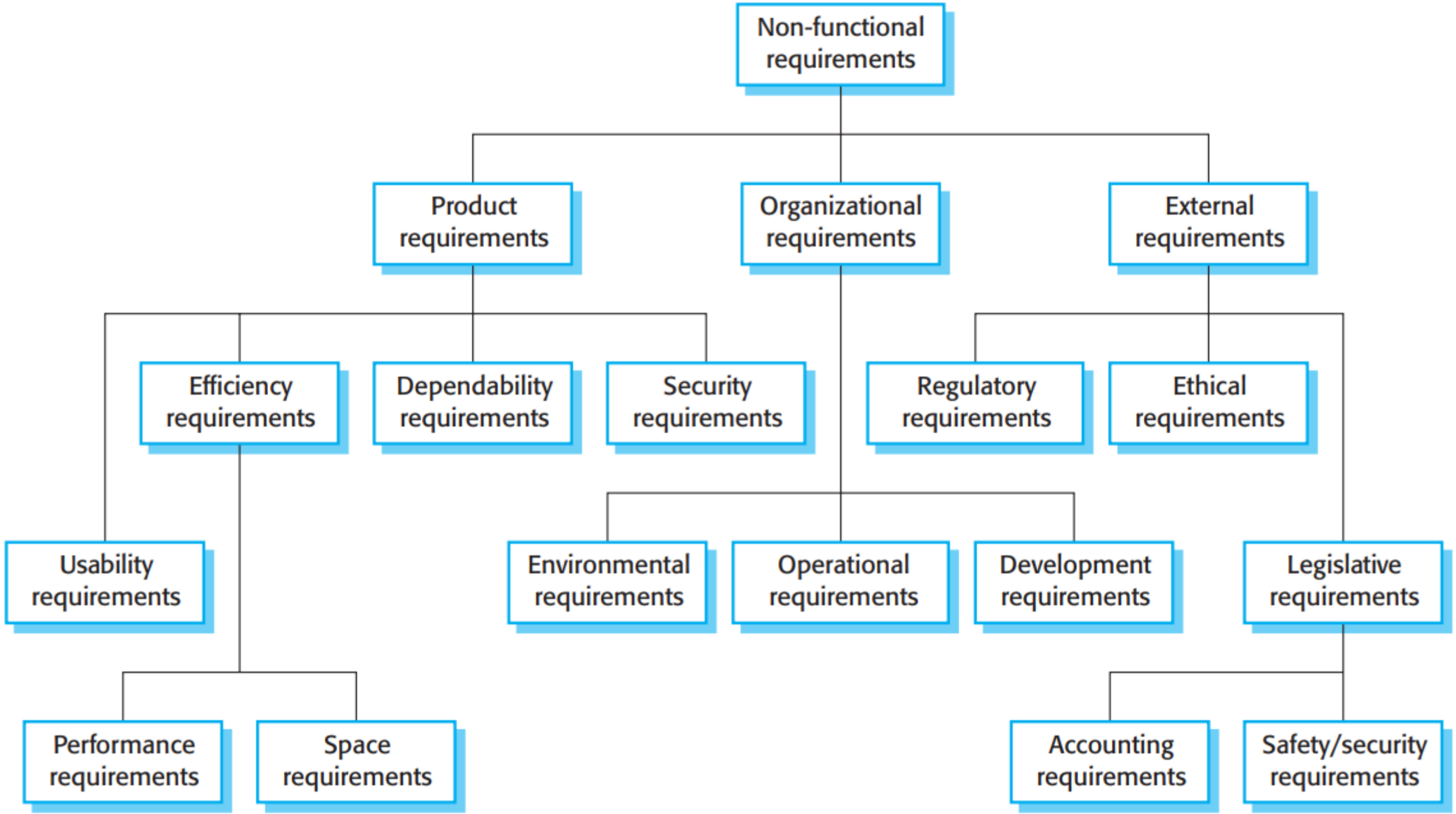
- Describe what the system should do.
- Depends on the type of software being developed, the expected users of the software, and the general approach taken by the organization when writing requirements.
- When expressed as user requirements, it should be written in natural language so that system users and managers can understand them.
- Functional system requirements are written for system developers.

Examples for functional requirements for the Mentcare system:

1. A user shall be able to search the appointments 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.

Non- functional Requirements

- Usually specify or constrain characteristics of the system as a whole.
- Often more critical than individual functional requirements.
- Failing to meet a non-functional requirement can mean that the whole system is unusable.
- Arise through user needs because of budget constraints, organizational policies, the need for interoperability with other software or hardware systems, or external factors such as safety regulations or privacy legislation.
- The implementation of these requirements may be spread throughout the system, for two reasons:
    1. May affect the overall architecture of a system rather than the individual components.
    2. May generate several, related functional requirements that define new system services that are required if the non-functional requirement is to be implemented.


Non-functional requirements
Product requirements
Organizational requirements
External requirements
Efficiency requirements
Dependability requirements
Security requirements
Regulatory requirements
Ethical requirements
Usability requirements
Environmental requirements
Operational requirements
Development requirements
Legislative requirements
Performance requirements
Space requirements
Accounting requirements
Safety/security requirements

1. **Product requirements ❼** specify or constrain the runtime behavior of the software.
   - Examples include how fast the system must execute, how much memory it requires, etc.
2. **Organizational requirements ❼** broad system requirements derived from policies and procedures in the customer's and developer's organizations.
   - Examples include operational process requirements that define how the system will be used; development process requirements that specify the programming language; the development environment or process standards to be used; and environmental requirements that specify the operating environment of the system.
3. **External requirements ❼** derived from factors external to the system and its development process.
   - Example include regulatory requirements that set out what must be done for the system to be approved for use by a regulator, such as a nuclear safety authority; legislative requirements that must be followed to ensure that the system operates within the law; and ethical requirements that ensure that the system will be acceptable to its users and the general public.

PRODUCT REQUIREMENT
The Mentcare system shall be available to all clinics during normal working hours (Mon–Fri, 08:30–17:30). Downtime within normal working hours shall not exceed 5 seconds in any one day.

ORGANIZATIONAL REQUIREMENT
Users of the Mentcare system shall identify themselves using their health authority identity card.

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

Property	Measure
Speed	Processed transactions/second User/event response time Screen refresh time
Size	Megabytes/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

Fig: Metrics for specifying non-functional requirements

Requirements Engineering Processes

- RE involves three key activities:
  1. Discovering requirements by interacting with stakeholders (elicitation and analysis)
  2. Converting these requirements into a standard form (specification)
  3. Checking that the requirements actually define the system that the customer wants (validation)
- The output of the RE process is a system requirements document.

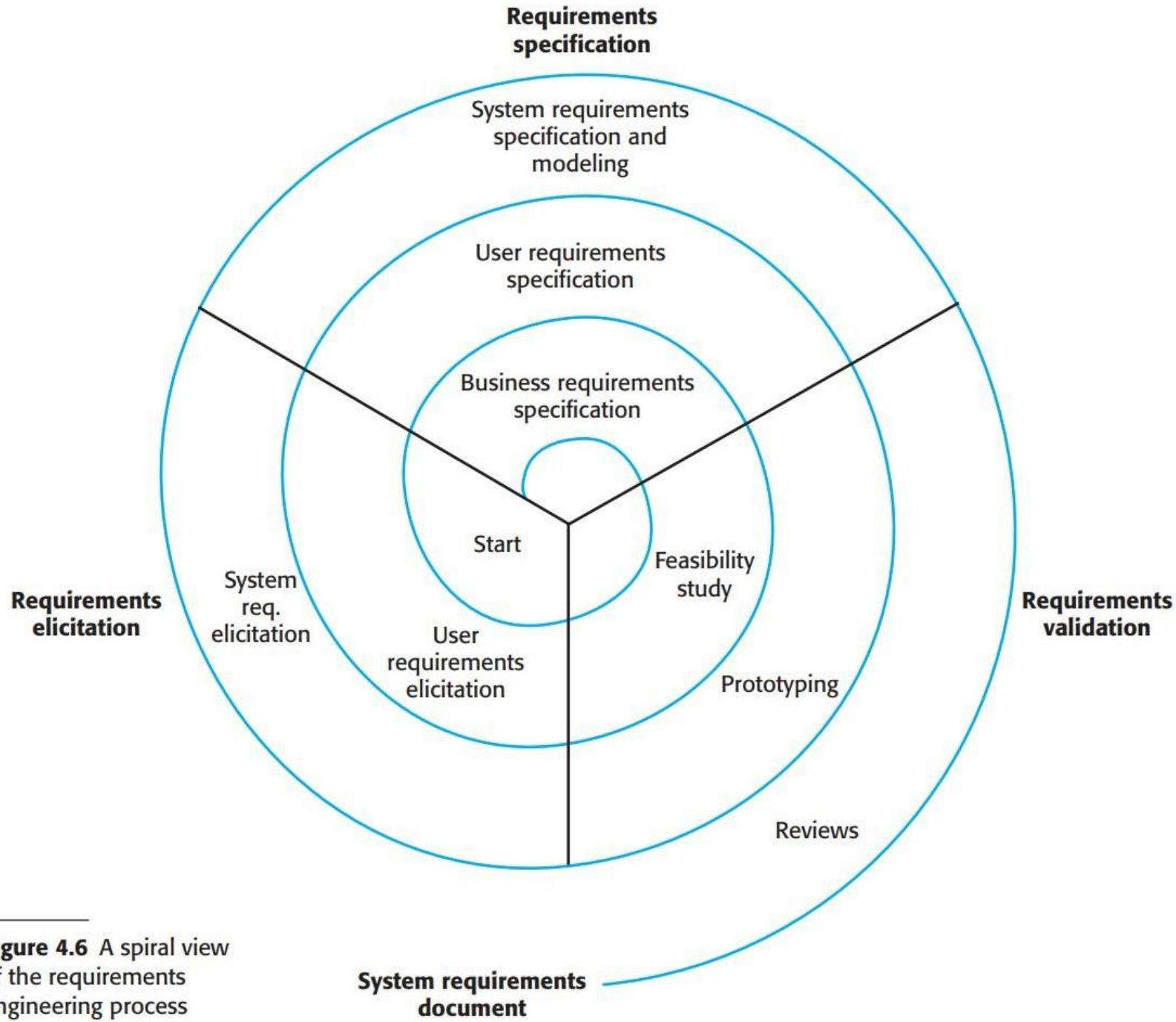


Figure 4.6 A spiral view of the requirements engineering process