

Chapter 3:

Requirements Elicitation

OBJECTIVES

After the completion of this chapter students will be able to:

- ☐ Define Requirements
- ☐ Identify actors and use cases.

Requirement Engineering

2

What is requirement engineering?

- ♥ The software requirements are description of features and functionalities of the target system.
- ♥ Requirements convey the expectations of users from the software product.
- ♥ The requirements for a system are the descriptions of what the system should do the services that it provides and the constraints on its operation.
- ♥ Requirement engineering provides the appropriate mechanism to **understand what the customer desires, analyzing the need, and assessing feasibility, negotiating a reasonable solution**, specifying the solution clearly, validating the specifications and managing the requirements as they are transformed into a working system.

- ♥ Requirements are described in the system requirements specification.
- ♥ Requirements engineering is the process of:
 - ❖ finding out,
 - ❖ analysing,
 - ❖ documenting and
 - ❖ checking

services and constraints
- ♥ The system requirements document is created and maintained during requirements engineering.

Cont....

- ♥ The goal of requirement engineering is to develop and maintain sophisticated and descriptive 'System Requirements Specification' document.
- ♥ The **analyst** starts *requirement gathering and analysis* activity by collecting all information from the customer.
- ♥ then analyzes the collected information to obtain a clear and thorough understanding of the product to be developed, with a view to remove all ambiguities and inconsistencies from the initial customer perception of the problem.

Cont..

5

- ♥ Project should be clearly understood by the analyst in order to obtain a good grasp of the problem.
- ♥ User requirements and system requirements may be defined as follows in the requirement engineering:
- ♥ ***User requirements***: -high-level abstract 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:*** - detailed description of what the system should do.

- ❖ Functions,
- ❖ Services, and
- ❖ Operational constraints.

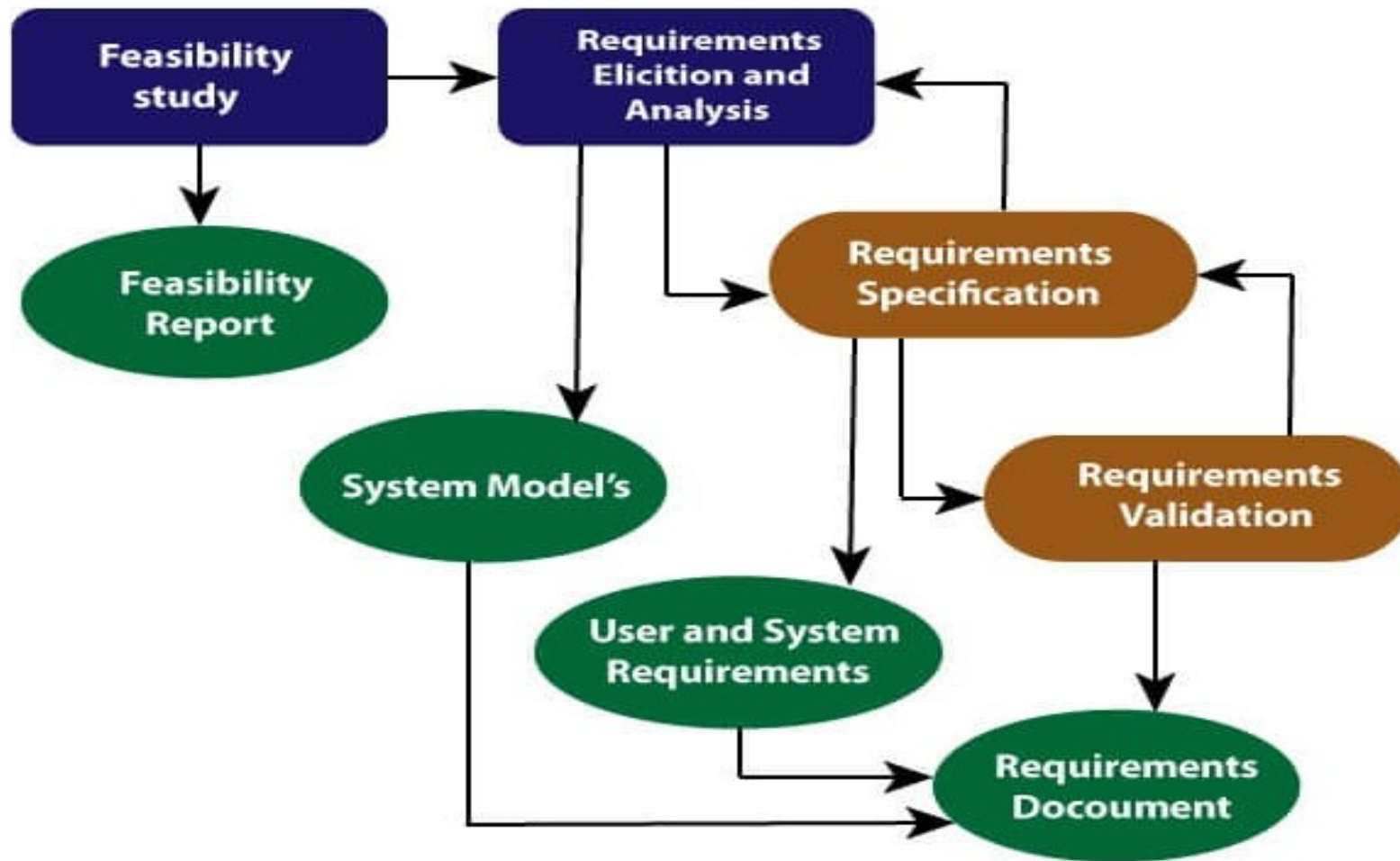
♥ The system requirements document (sometimes called a functional specification) should define exactly what is to be implemented.

Requirement Engineering Process

It is a five-step process, which includes -

- ❖ Feasibility Study
- ❖ Requirement Elicitation and Analysis
- ❖ Software Requirement Specification
- ❖ Software Requirement Validation
- ❖ Software Requirement Management

Cont...



Feasibility study

- ❖ The organization for getting the desired product developed, it comes up with rough idea about what all functions the software must.
- ❖ Referencing to this information, the analysts does a detailed study about whether the desired system and its functionality are feasible to develop.
- ❖ Feasibility study analyzes whether the software product can be practically materialized in terms of implementation, contribution of project to organization, cost constraints and as per values and objectives of the organization.
- ❖ It explores technical aspects of the project and product such as usability, maintainability, productivity and integration ability.
- ❖ The output of this phase should be a feasibility study report that should contain adequate comments and recommendations for management about whether or not the project should be undertaken.

Requirement Gathering and SRS

- ❖ If the feasibility report is positive towards undertaking the project, next phase starts with gathering requirements from the user.
- ❖ Analysts and engineers communicate with the client and end-users to know their ideas on what the software should provide and which features they want the software to include.

Software Requirement Specification

- ❖ SRS is a document created by system analyst after the requirements are collected from various stakeholders.
- ❖ SRS defines how the intended software will interact with hardware, external interfaces, speed of operation, response time of system, portability of software across various platforms, maintainability, speed of recovery after crashing, Security, Quality, Limitations etc.
- ❖ The requirements received from client are written in natural language and it is the responsibility of system analyst to document the requirements in technical language.

Software Requirement Validation

- ❖ After requirement specifications are developed, the requirements mentioned in this document are validated. User might ask for illegal, impractical solution or experts may interpret the requirements incorrectly.

This results in huge increase in cost if not nipped in the bud.

Requirements can be checked against following conditions -

- ❖ If they can be practically implemented
- ❖ If they are valid and as per functionality and domain of software
- ❖ If there are any ambiguities
- ❖ If they are complete
- ❖ If they can be demonstrated

Requirement Elicitation Process

Requirements gathering - The developers discuss with the client and end users and know their expectations from the software.

Organizing Requirements - The developers prioritize and arrange the requirements in order of importance, urgency and convenience.

Negotiation & discussion - If requirements are ambiguous or there are some conflicts in requirements of various stakeholders, if they are, it is then negotiated and discussed with stakeholders.

The requirements come from various stakeholders. To remove the ambiguity and conflicts, they are discussed for clarity and correctness. Unrealistic requirements are compromised reasonably.

Documentation - All formal & informal, functional and non-functional requirements are documented and made available for next phase processing.

Types of Software requirement

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.
- ❖ 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 should describe the system services in detail.

Functional requirements for the CMS

- 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 imprecision

- ❖ Problems arise when requirements are not precisely 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.

Requirements completeness and consistency

In principle, requirements should be both **complete** and **consistent**

❖ Complete

- They should include descriptions of all facilities required

❖ Consistent

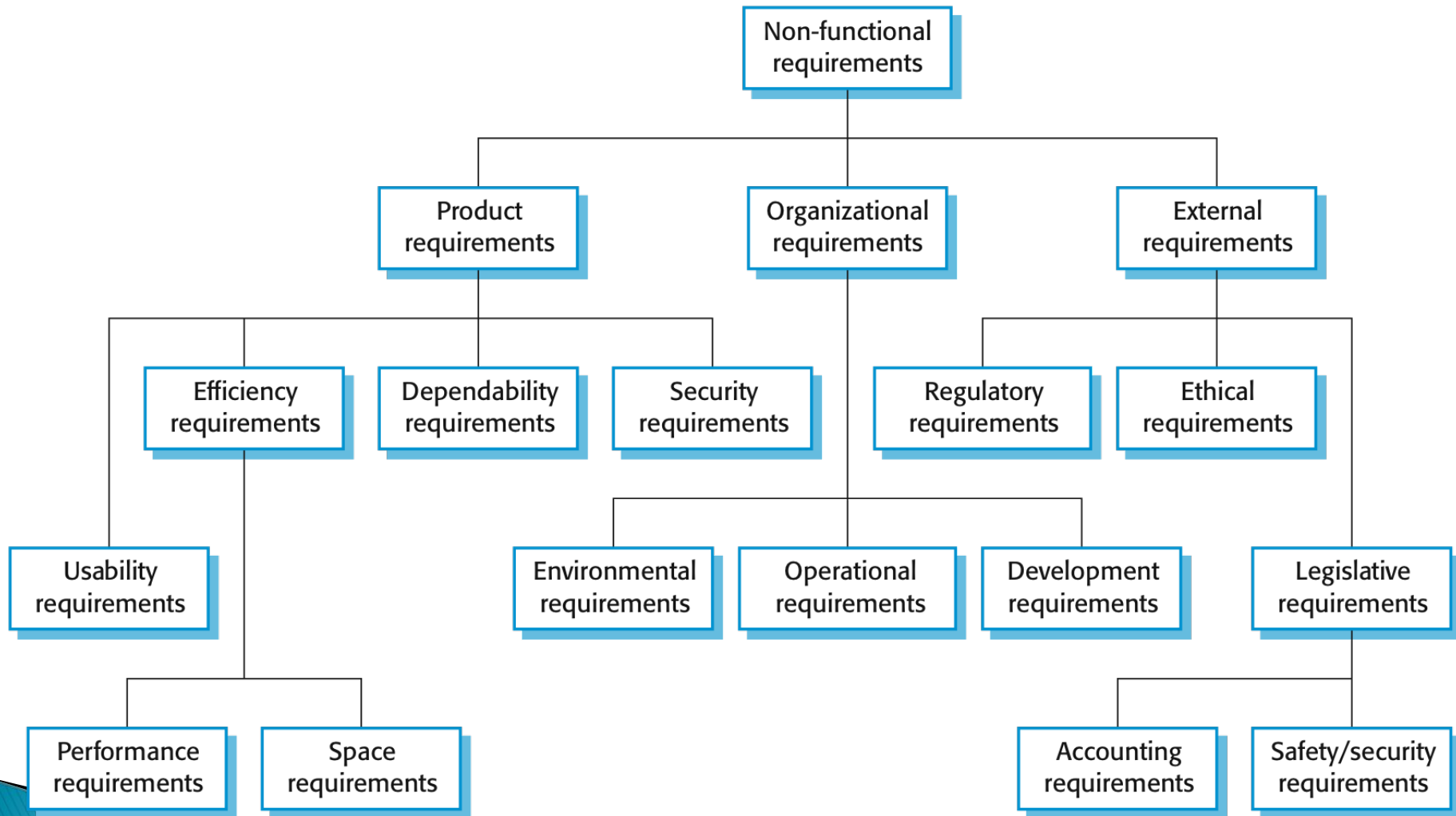
- There should be no conflicts or contradictions in the descriptions of the system facilities

In practice, it is impossible to produce a complete and consistent requirements document

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.
- ❖ Non-functional requirements may be more critical than functional requirements. If these are not met, the system may be useless.
 - Constraints on the services or functions offered by the system

Types of nonfunctional requirement



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 classifications

▶ Product requirements

- Requirements which specify that the delivered product must behave in a particular way e.g. execution speed, reliability, etc.

▶ Organizational requirements

- Requirements which are a consequence of organizational 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.

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 are helpful to developers as they convey the intentions of the system users.

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

The software requirements document

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

Users of a requirements document

Project Stakeholders:

- **Product Owners:** They provide input on business needs and priorities.
- **Customers/Clients:** They articulate their requirements and expectations for the final product.
- **Project Managers:** They use the requirements to plan, schedule, and manage resources effectively.

Development Team:

- **Developers:** Use the document to understand the features and functionalities they need to implement.
- **Architects:** Rely on requirements to design the system architecture.
- **Quality Assurance (QA) Team:** Use requirements to create test plans and ensure the final product meets specified criteria.

Cont...

Designers:

- **User Interface (UI) Designers:** Rely on functional requirements to design user interfaces.
- **System Designers:** Use requirements to guide the overall system architecture.

Documentation Team:

- **Technical Writers:** Use the requirements to create user manuals, technical documentation, and other relevant materials.

Regulatory Authorities and Compliance Teams:

- **Compliance Officers:** Use requirements to ensure that the project adheres to relevant regulations and standards.

Cont...

Marketing and Sales Teams:

- **Marketing Professionals:** Use the requirements to understand the features and benefits of the product for promotional purposes.
- **Sales Representatives:** Rely on the document to communicate product capabilities to potential customers.

End Users:

- **End Users or Customers:** Though not typically involved in the creation of the requirements document, they are the ultimate users of the product. The document reflects their needs and expectations.

Business Analysts:

- **Business Analysts:** They may be involved in the creation of the requirements document and use it to bridge the communication gap between stakeholders and the development team.

Cont...

Legal Teams:

- **Legal Professionals:** They may use the requirements document to ensure that contractual agreements and legal obligations are met.

► Maintenance and Support Teams:

- **Maintenance and Support Personnel:** Refer to the requirements to understand the intended functionality and make updates or fixes after the product is deployed.

Requirements specification

- ❖ Software requirement specification is a kind of document which is created by a software analyst after the requirements collected from the various sources.
- ❖ 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
- ❖ The requirements may be part of a contract for the system development
 - ✓ It is therefore important that these are as complete as possible

Cont...

- **Data Flow Diagrams:** Data Flow Diagrams (DFDs) are used widely for modeling the requirements. DFD shows the flow of data through a system. The system may be a company, an organization, a set of procedures, a computer hardware system, a software system, or any combination of the preceding.
- **Data Dictionaries:** Data Dictionaries are simply repositories to store information about all data items defined in DFDs. A data dictionary is a file or a set of files that contains a database's metadata.
- **Entity-Relationship Diagrams:** It is a detailed logical representation of the data for the organization and uses three main constructs i.e. data entities, relationships, and their associated attributes.

Requirements and design

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

- ▶ In practice, requirements and design are inseparable
 - A system architecture may be designed to structure the requirements;
 - The system may inter-operate with other systems that generate design requirements;
 - The use of a specific architecture to satisfy non-functional requirements may be a domain requirement.
 - This may be the consequence of a regulatory requirement.

Guidelines for writing requirements

- ❖ Create a **standard format** and use it for all requirements.
- ❖ Use language in a consistent way. Use **shall** for **mandatory** requirements, **should** for **desirable** requirements.
- ❖ Use text highlighting to identify key parts of the requirement.
- ❖ Avoid the use of computer jargon, abbreviations, acronyms(or technical SE language)
- ❖ Include an explanation (rationale) of why a requirement is necessary.

Requirements elicitation and analysis

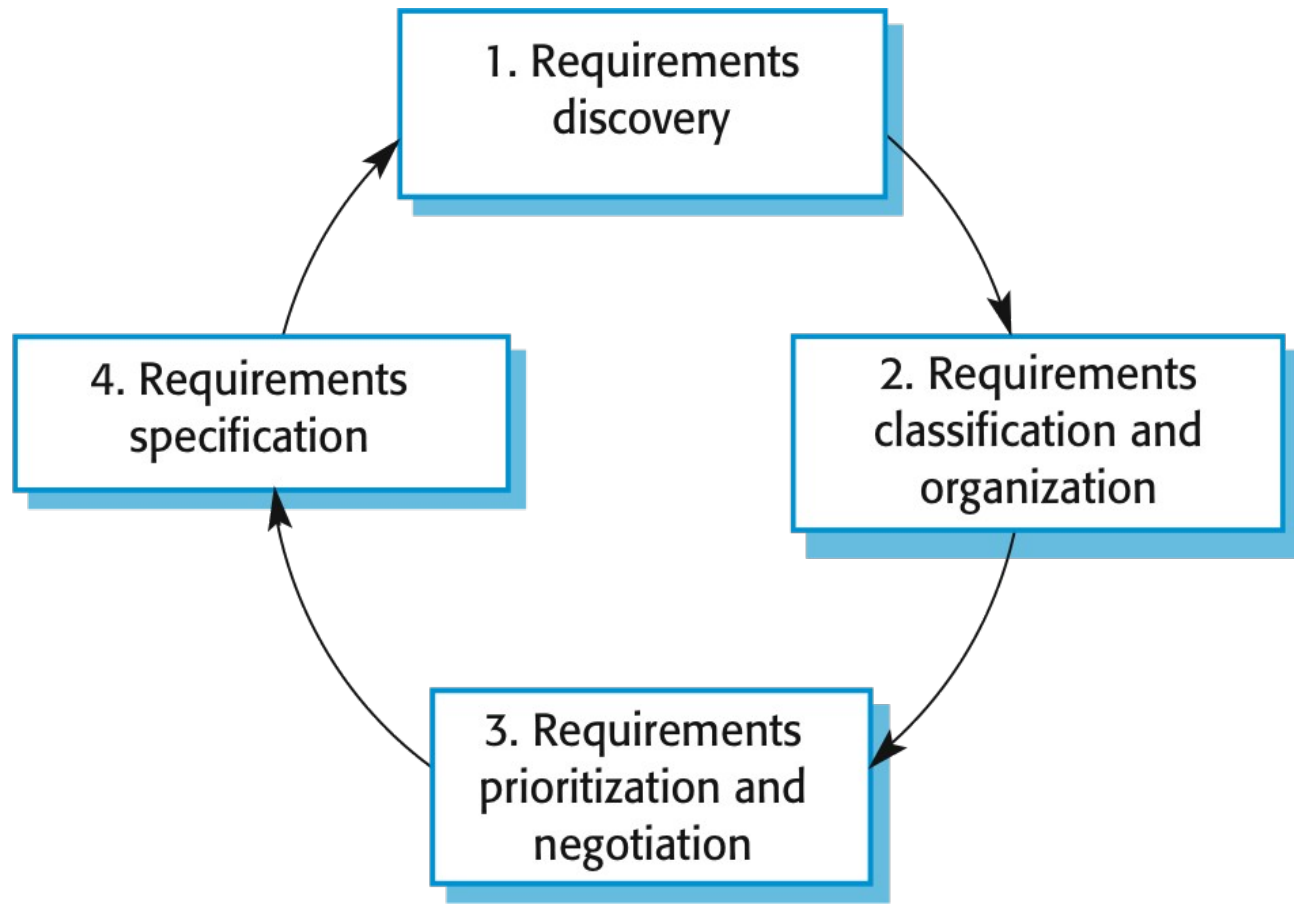
Sometimes called *requirements elicitation* or *requirements discovery*.

- ❖ 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*.

Problems of requirements analysis

- Stakeholders don't know what they really want.
- Stakeholders express requirements in their own terms.
- Different stakeholders may have conflicting requirements.
- Organisational and political factors may influence the system requirements.
- The requirements change during the analysis process. New stakeholders may emerge and the business environment may change.

The requirements elicitation and analysis process



Process activities

Requirements discovery

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

Requirements classification and organization

- Groups related requirements and organizes them into coherent clusters.

Prioritization and negotiation

- Prioritizing requirements and resolving requirements conflicts.

Requirements specification

- Requirements are documented and input into the next round of the spiral.

Requirements discovery

- ▶ The process of gathering information about the required and existing systems and distilling the user and system requirements from this information.
- ▶ Interaction is with system stakeholders from managers to external regulators.
- ▶ Systems normally have a range of stakeholders.

Requirement Gathering tools

Interview

- ❖ 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.
- ❖ 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 a springboard question, a requirements proposal, or by working together on a prototype system.

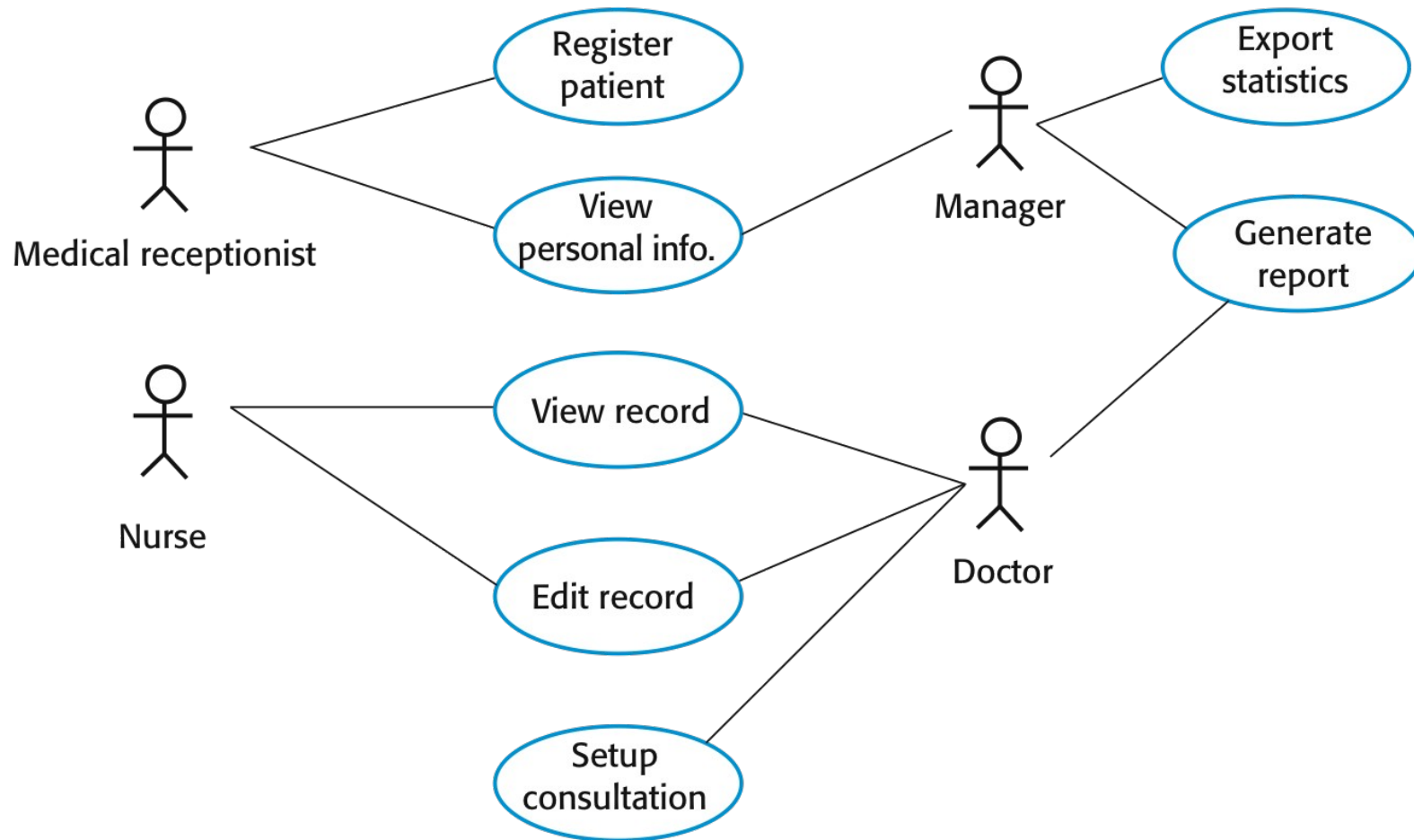
Interviews in practice

- ❖ Normally a mix of closed and open-ended interviewing.
- ❖ Interviews are good for getting an overall understanding of what stakeholders do and how they might interact with the system.
- ❖ Interviews are not good for understanding domain requirements
 - Requirements engineers cannot understand specific domain terminology;
 - Some domain knowledge is so familiar that people find it hard to articulate or think that it isn't worth articulating.

Use cases

- ❖ Use-cases are a scenario based technique in the UML which identify the actors in an interaction and which describe the interaction itself.
- ▶ A set of use cases should describe all possible interactions with the system.
- ❖ High-level graphical model supplemented by more detailed tabular description.
- ❖ Sequence diagrams may be used to add detail to use-cases by showing the sequence of event processing in the system.

Use cases for the CMS



Requirements validation

- ❖ 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 validation 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.

Requirements reviews

- ❖ Regular reviews should be held while the requirements definition is being formulated.
- ❖ Both client and contractor staff should be involved in reviews.
- ❖ Reviews may be formal (with completed documents) or informal. Good communications between developers, customers and users can resolve problems at an early stage.

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?

Requirements management

- ▶ *Requirements management* is the process of managing changing requirements during the requirements engineering process and system development.
- ▶ New requirements emerge as a system is being developed and after it has gone into use.
- ▶ You need to keep track of individual requirements and maintain links between dependent requirements so that you can assess the impact of requirements changes. You need to establish a formal process for making change proposals and linking these to system requirements.

Requirements evolution

