Requirements Engineering

Requirements Engineering

- The process of establishing the services that the customer requires from a system.

- The constraints under which the system operates and is developed.

What Are Requirements?

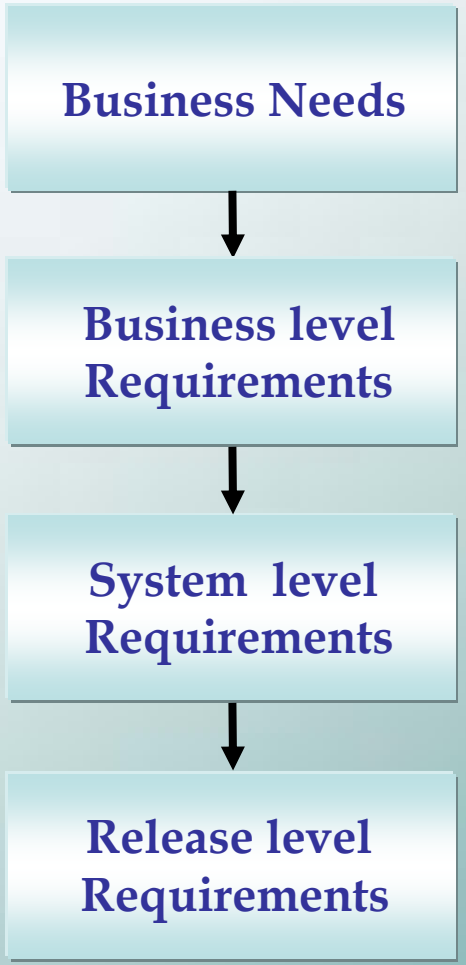
Requirements definition is a careful assessment of the needs that a system is to fulfill.

It Says

- **Why a system is needed?**
- **What system features will serve and satisfy?**
- **How the system is to be constructed?**

Requirements represent a specification for the new system.

Requirements are Defined at Different Levels



What are the objectives/goals of the business?

To meet these needs, what process, procedures etc. are required?

What does the system have to do to meet the business's requirements?

When and how will these requirements be met?

Why is Requirements Engineering Important?

- Identifies user needs to ensure that you are building the right system
- Help you manage the development process to ensure a quality system
- Identify defects early, reducing costs

Types of Requirement

User requirements

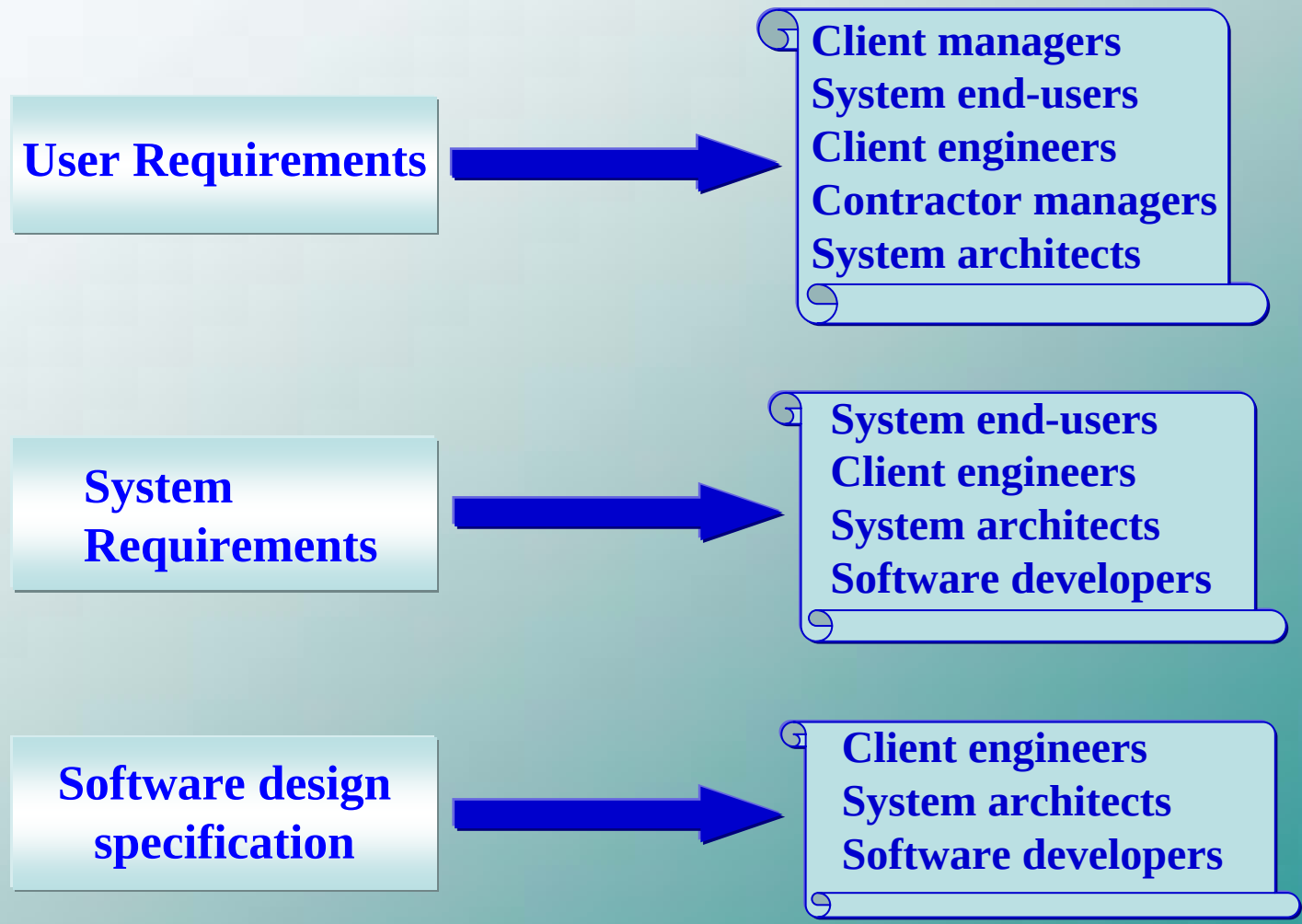
- State the services that the system should provide.
- State the operational constraints of the system.

Customer

System requirements

- Define in 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 Readers



Functional and Non-functional Requirements

Functional (Behavioral) requirements

Specify

- The services that the system should provide.
- How the system should react to particular inputs.
- How the system should behave in particular situations.

Non Functional (Non-Behavioral) requirements

Describe

- Constraints on the services or functions offered by the system such as timing constraints.
- Constraints on the development process, standards, etc.

Domain requirements

Define project characteristics such as the schedule, the cost, required design documents, quality procedures etc.

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 user requirements may be high-level statements of what the system should do
- Functional system requirements should describe the system services in detail.

Examples of Functional Requirements

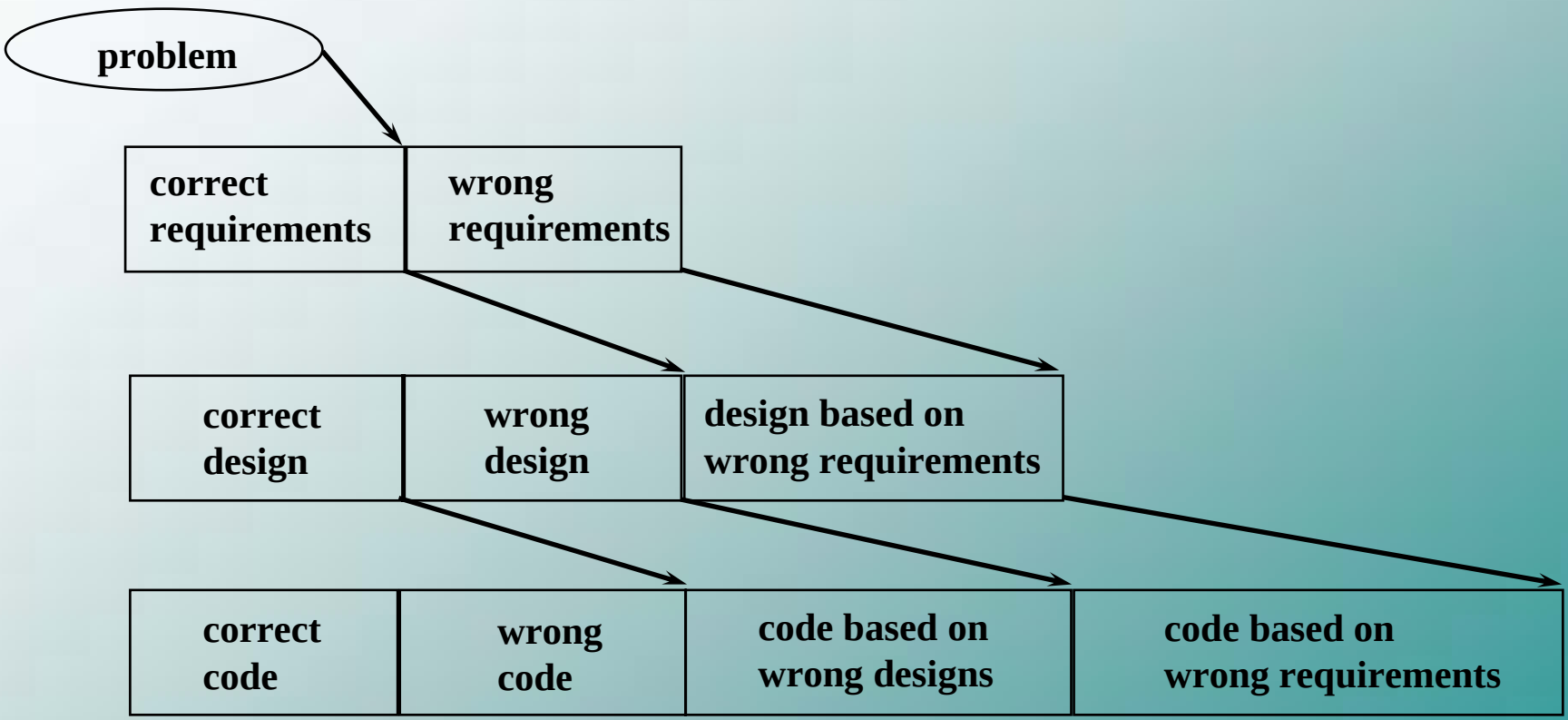
- The user shall be able to search [illegible] the initial set of databases o[illegible] set from it.

User Requirement

- Every order shall be allocated a unique identifier (ORDER_ID) w[illegible] be able to copy to the account's per[illegible]ent storage area.

System Requirement

Defects Propagate and Grow



Requirements Completeness and Consistency

Complete

Requirements 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

Non-functional Requirements

- Define system properties e.g. reliability, response time and storage requirements.
- Define constraints e.g. I/O device capability, system representations, etc.
- Specify CASE system, programming language or development method.

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

Non-functional Classifications

Product requirements

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

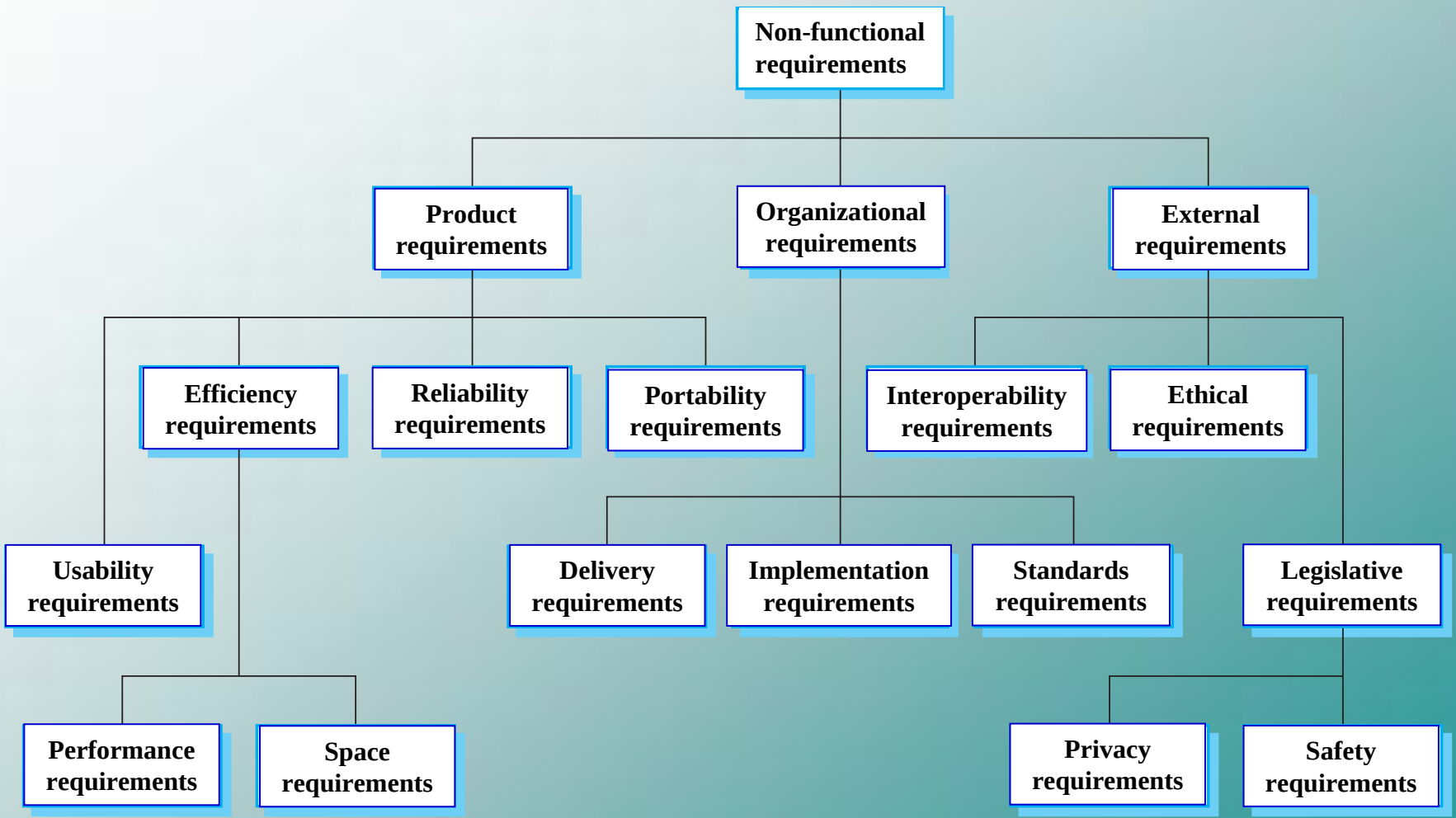
Organisational requirements

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 Requirement Types



Non-functional Requirements Examples

Product requirements

The user interface for LIBSYS shall be implemented as simple HTML without frames or Java applets.

Organisational requirements

The system development process and deliverable documents shall conform to the process and deliverables defined in XYZCo-SP-STAN-95.

External requirements

The system shall not disclose any personal information about customers apart from their name and reference number to the operators of the system.

Requirements Measures

Property	Measure
Speed	Processed transactions/second User/Event response time Screen refresh time
Size	M Bytes 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

Domain requirements

- Describe system characteristics and features that reflect the domain.

- Domain requirements be new functional requirements, constraints on existing requirements or define specific computations.

If domain requirements are not satisfied, the system may be unworkable.

User Requirements

- Describe functional and non-functional requirements in such a way that they are understandable by system users who don't have detailed technical knowledge.

- User requirements are defined using natural language, tables and diagrams as these can be understood by all users.

Guidelines for Writing Requirements

- Invent 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.

System Requirements

- More detailed specifications of system functions, services and constraints than user requirements.
- They are intended to be a basis for designing the system.
- They may be incorporated into the system contract.
- System requirements may be defined or illustrated using system models.

Requirement specifications

- Definition of the function or entity.
- Description of inputs and where they come from.
- Description of outputs and where they go to.
- Indication of other entities required.
- Pre and post conditions (if appropriate).
- The side effects (if any) of the function.

The requirements document

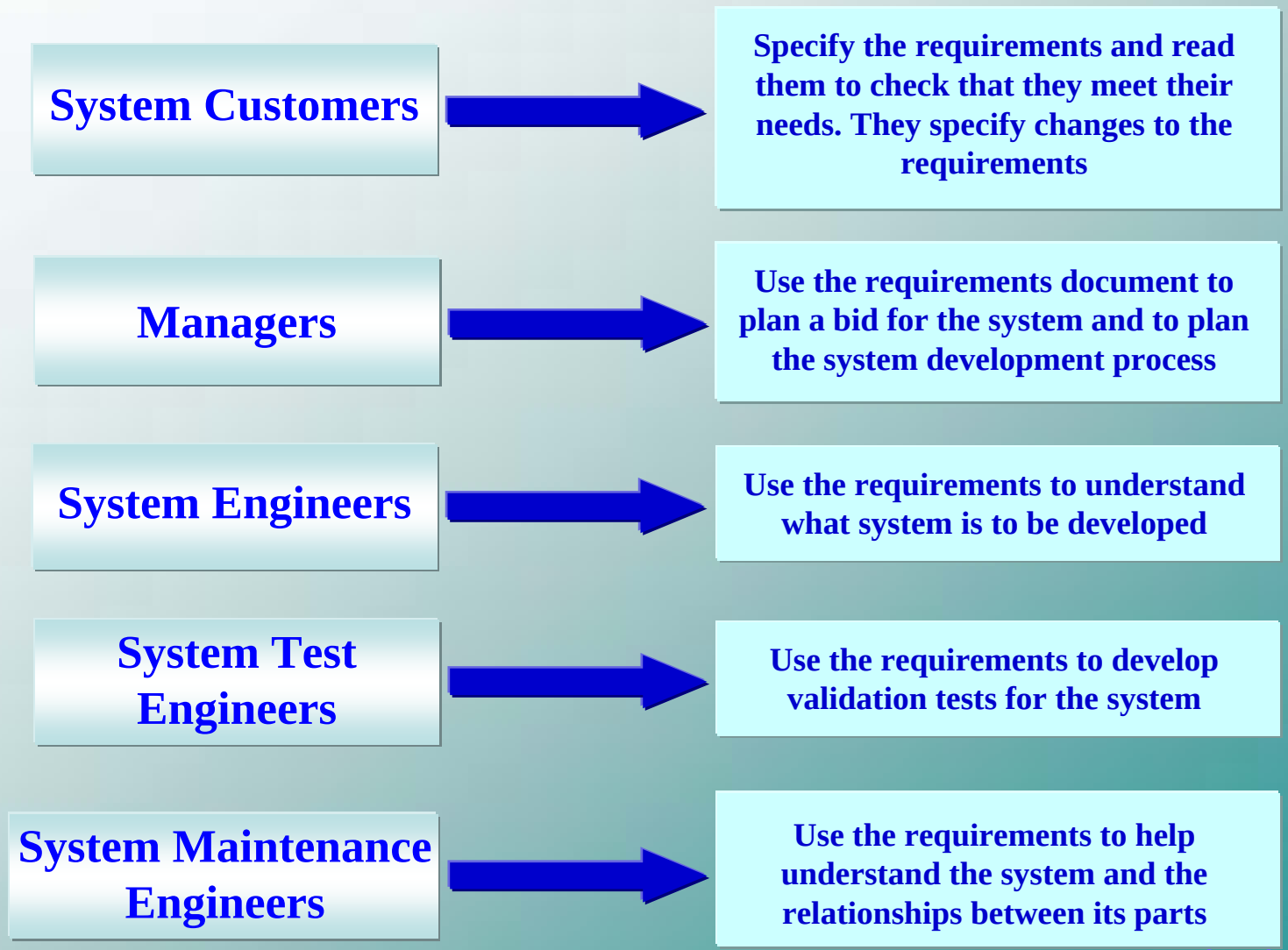
- The requirements document is the official statement of what is required of the system developers.
- Should include both a definition of user requirements and a specification of the system requirements.
- It is NOT a design document. As far as possible, it should set of WHAT the system should do rather than HOW it should do it

Stakeholders who participate in the Requirements Analysis Process

- Users
- Managers
- Analysts and Designers
- Sponsors/funders
- Developers

Each with different needs and perspectives

Users of a Requirements Document



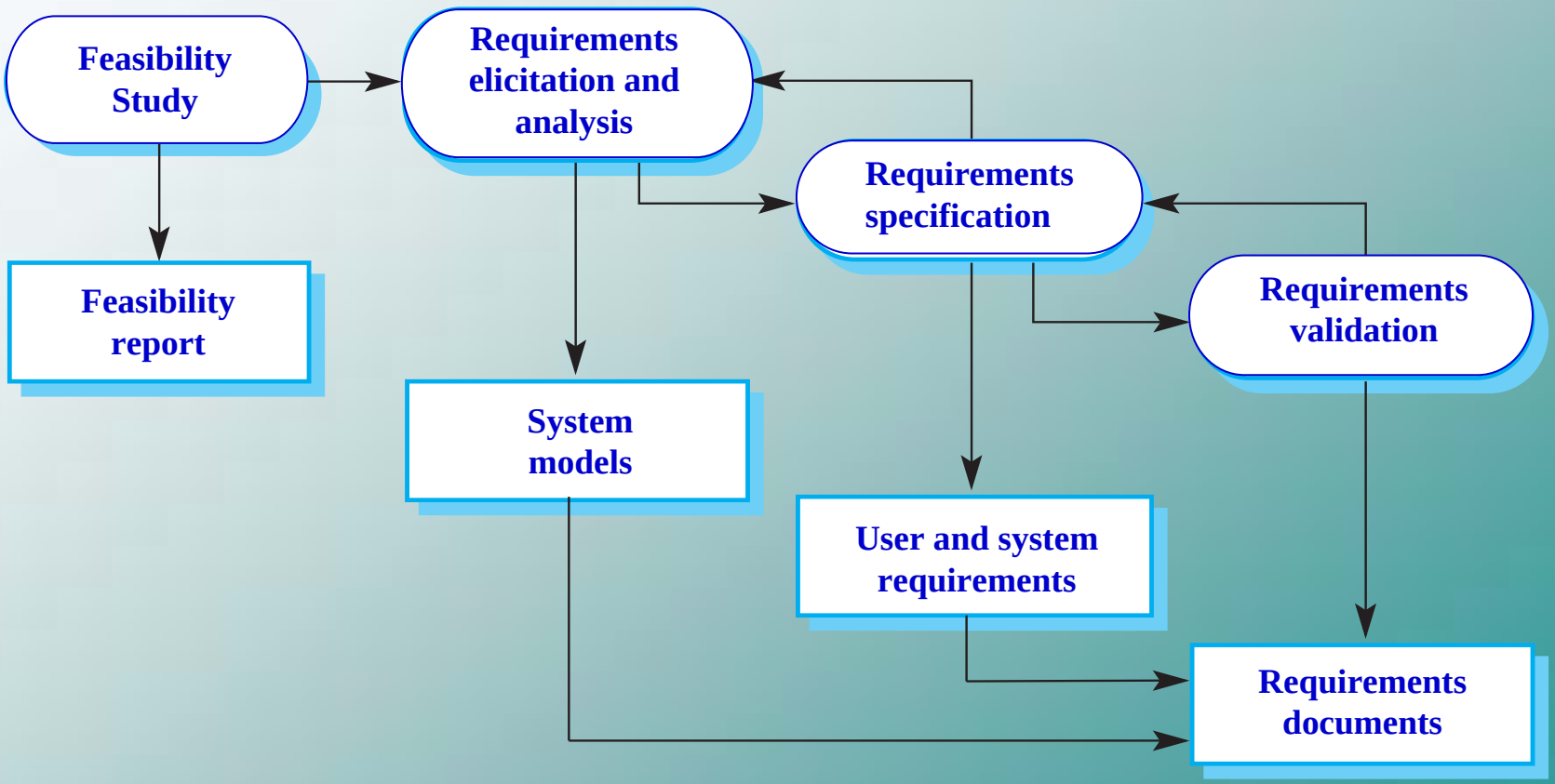
IEEE requirements standard

- **Defines a generic structure for a requirements document that must be instantiated for each specific system.**
  - **Introduction.**
  - **General description.**
  - **Specific requirements.**
  - **Appendices.**
  - **Index.**

Requirements Document Structure

- Preface
- Introduction
- Glossary
- System architecture
- System requirements specification
- System models
- System evolution
- Appendices
- Index

The Requirements Engineering Process



Activities in the requirements process

- **Requirements Elicitation**
  is a process of identifying stakeholders needs and bridging the disparities among the different stakeholders requirements.
- **Requirements Analysis and Organization**
  is the activity of structuring and synthesizing the information gathered during elicitation
- **Requirements Specification**
  specifies the requirements in a unambiguous and complete form as possible
- **Requirements Validation**
  is the activity of checking the quality of the requirements to ensure they are: Correct, Complete, Unambiguous, etc
- **Requirements Management**
  provides management and tracking of the requirements as the system is being developed

Social and Organisational Factors

- Software systems are used in a social and organisational context. This can influence or even dominate the system requirements.
- Social and organisational factors are not a single viewpoint but are influences on all viewpoints.
- Good analysts must be sensitive to these factors but currently no systematic way to tackle their analysis.

Requirements Validation

- Software systems are used in a social and organizational context. This can influence or even dominate the system requirements.

- 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 Techniques

- **Requirements reviews**

  Systematic manual analysis of the requirements.

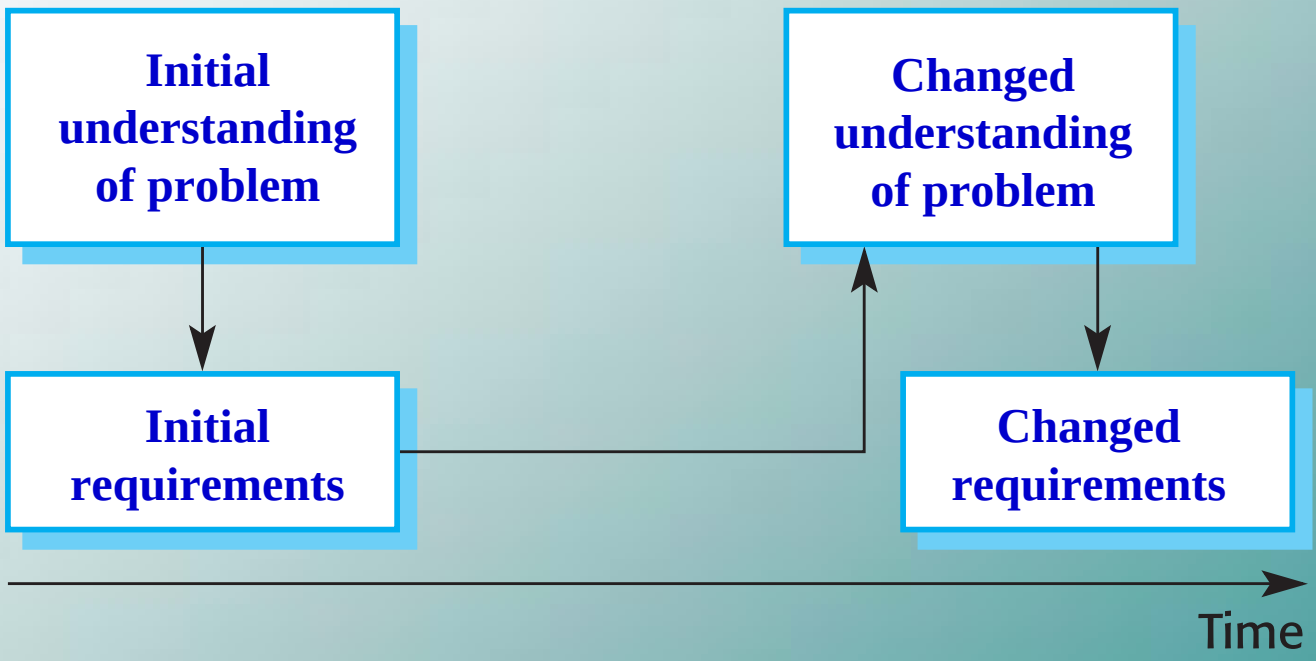
- **Prototyping**

  Using an executable model of the system to check requirements.

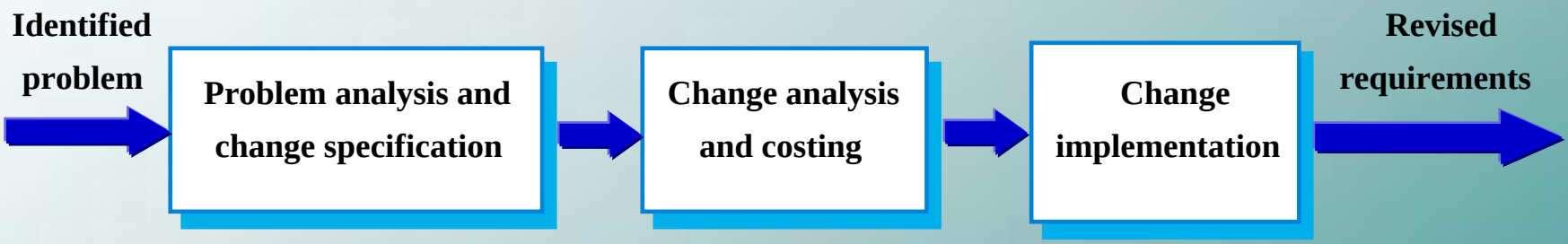
- **Test-case generation**

  Developing tests for requirements to check testability.

Requirements Evolution



Requirements Change Management



Questions