



---

# *Requirement Engineering Process*

---



# Requirements Engineering

- **Establishing** what the customer requires from a software system
  - **what is it?**
- An important life cycle phase
- Consists of two distinct activities
  - Requirements gathering and analysis
  - Requirements specification



# **Output of Requirements Analysis and Specification Stage**

- Software Requirements Specification (SRS) Document:
- serves as contract between the customer and the developers.



# What is a requirement?

- It may range from a **high-level** abstract statement of a service or of a system constraint to a **detailed** mathematical functional specification
- This is inevitable as requirements may serve a dual function
  - 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

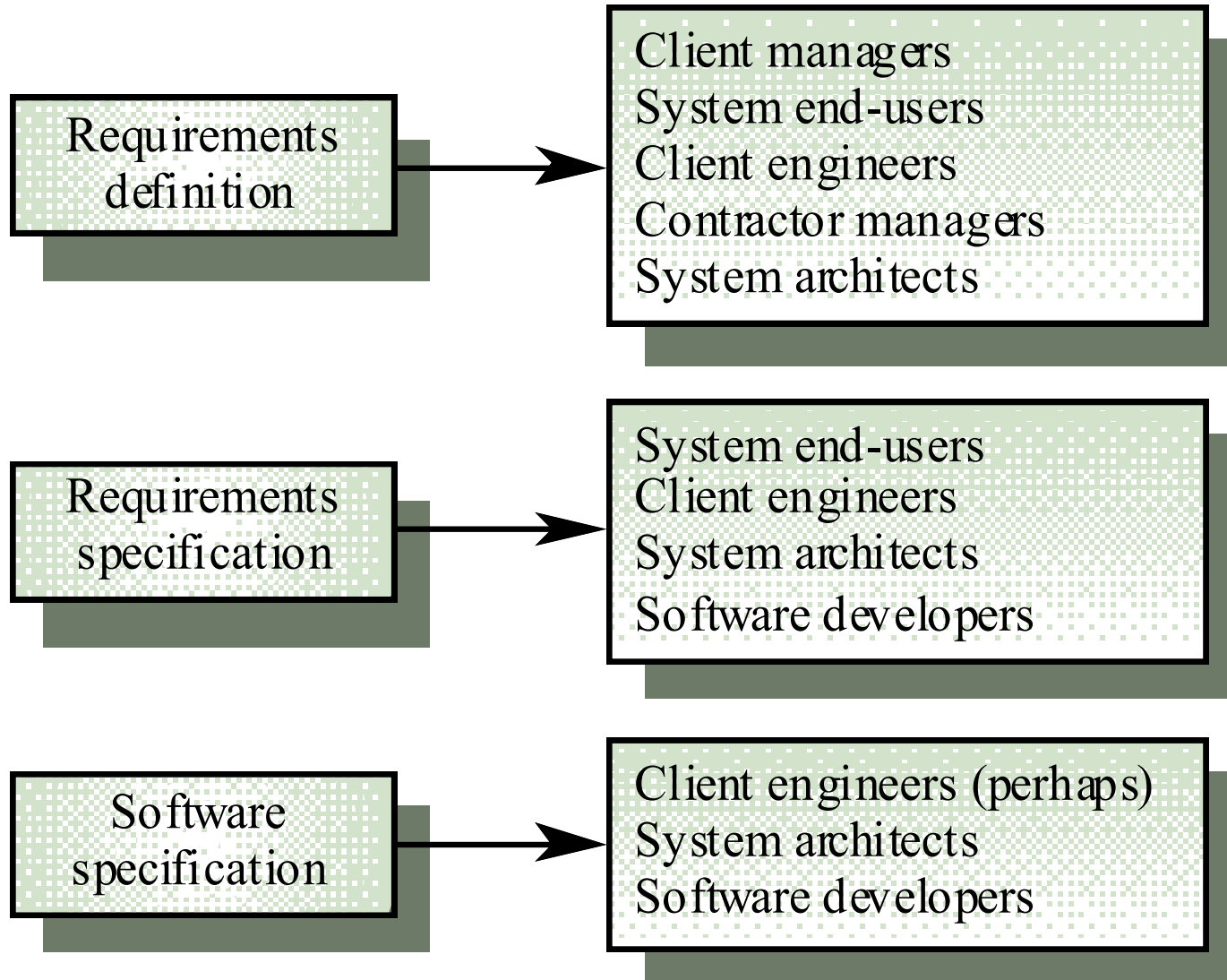


# Requirements definition/specification

- Requirements definition
  - A statement in natural language plus diagrams of the services the system provides and its operational constraints. Written for customers
- Requirements specification
  - A structured document setting out detailed descriptions of the system services. Written **as a contract** between client and contractor
- Software specification
  - A detailed software description which can serve as a basis for a design or implementation. Written for **developers**



# Requirements readers





# The requirements engineering process

- Feasibility study
  - Find out if the current user needs be satisfied given the available technology and budget?
- Requirements analysis
  - Find out what system stakeholders require from the system
- Requirements definition
  - Define the requirements in a form understandable to the customer
- Requirements specification
  - Define the requirements in detail



# Requirement Engineering Process

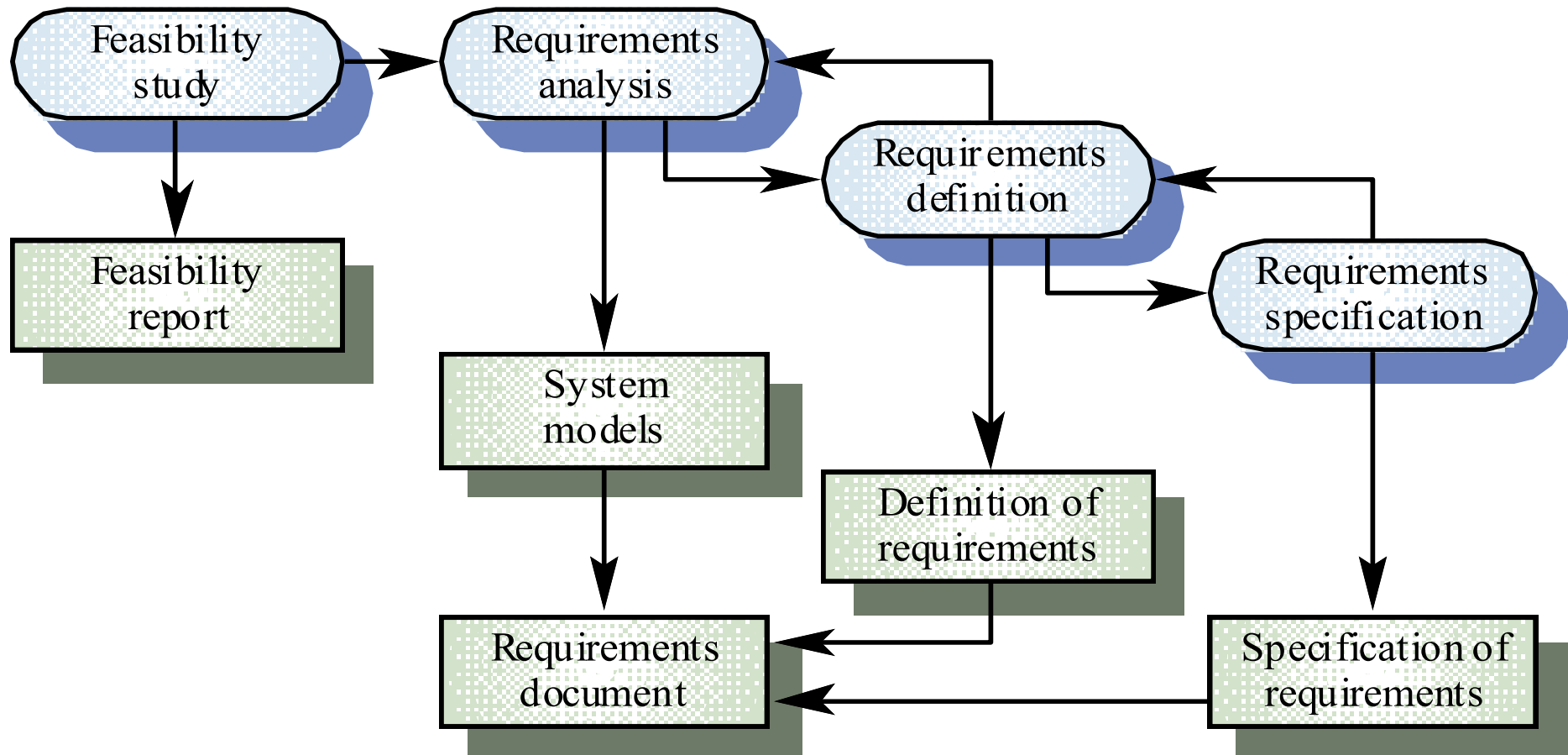
## **Feasibility Study:**

- Does the system contribute to the overall objective of the Organization ?
- Can the system be implemented with current technology, specified cost and schedule constraints ?
- Can the system be integrated with the other systems which are already in place ?
- Requirement Elicitation and Analysis
- Interaction with the customer to understand the domains, scenarios, viewpoints, ethnography etc.





# The RE process





# The Requirements Document

- The requirements document is the official statement of what is required of the system developers
- Should include both a definition and a specification of 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



# Requirements document requirements

- Specify external system behaviour
- Specify implementation constraints
- Easy to change
- Serve as reference tool for maintenance
- Record forethought about the life cycle of the system i.e. predict changes
- Characterise responses to unexpected events



# Requirements document structure

- Introduction
  - Describe need for the system and how it fits with business objectives
- Glossary
  - Define technical terms used
- System models
  - Define models showing system components and relationships
- Functional requirements definition
  - Describe the services to be provided



# Requirements document structure

- *Non-functional requirements definition*
  - Define constraints on the system and the development process
- System evolution
  - Define fundamental assumptions on which the system is based and anticipated changes
- Requirements specification
  - Detailed specification of functional requirements
- Appendices
  - System hardware platform description
  - Database requirements (as an ER model perhaps)
- Index



# 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
- Prototyping is an important technique of requirements validation



# Requirements 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



# Types of Requirements

- Functional requirements:
  - input/output
  - processing.
  - error handling.
- Non-functional requirements:
  - Physical environment (equipment locations, multiple sites, etc.).
  - Interfaces (data medium etc.).
  - User & human factors (who are the users, their skill level etc.).





## Types of Requirements (Cont.)

- Non-functional requirements (continued):
  - Performance (how well is system functioning).
  - Documentation.
  - Data (qualitative stuff).
  - Resources (finding, physical space).
  - Security (backup, firewall).
  - Quality assurance (max. down time, MTBF, etc.).



# Requirement Engineering Process

## **System Model:**

- Three types of system modeling exists.
- Data Requirements leads to Data Modeling and we get ERD ( Entity Relationship Diagram )
- Functional Requirements lead to Functional Modeling and we get DFD ( Data Flow Diagram ). It shows the information flow among various entities and functionalities.
- Behavioral Requirements lead to Behavioral Modeling and we get STD ( State Transition Diagram)