SOFTWARE ENGINEERING

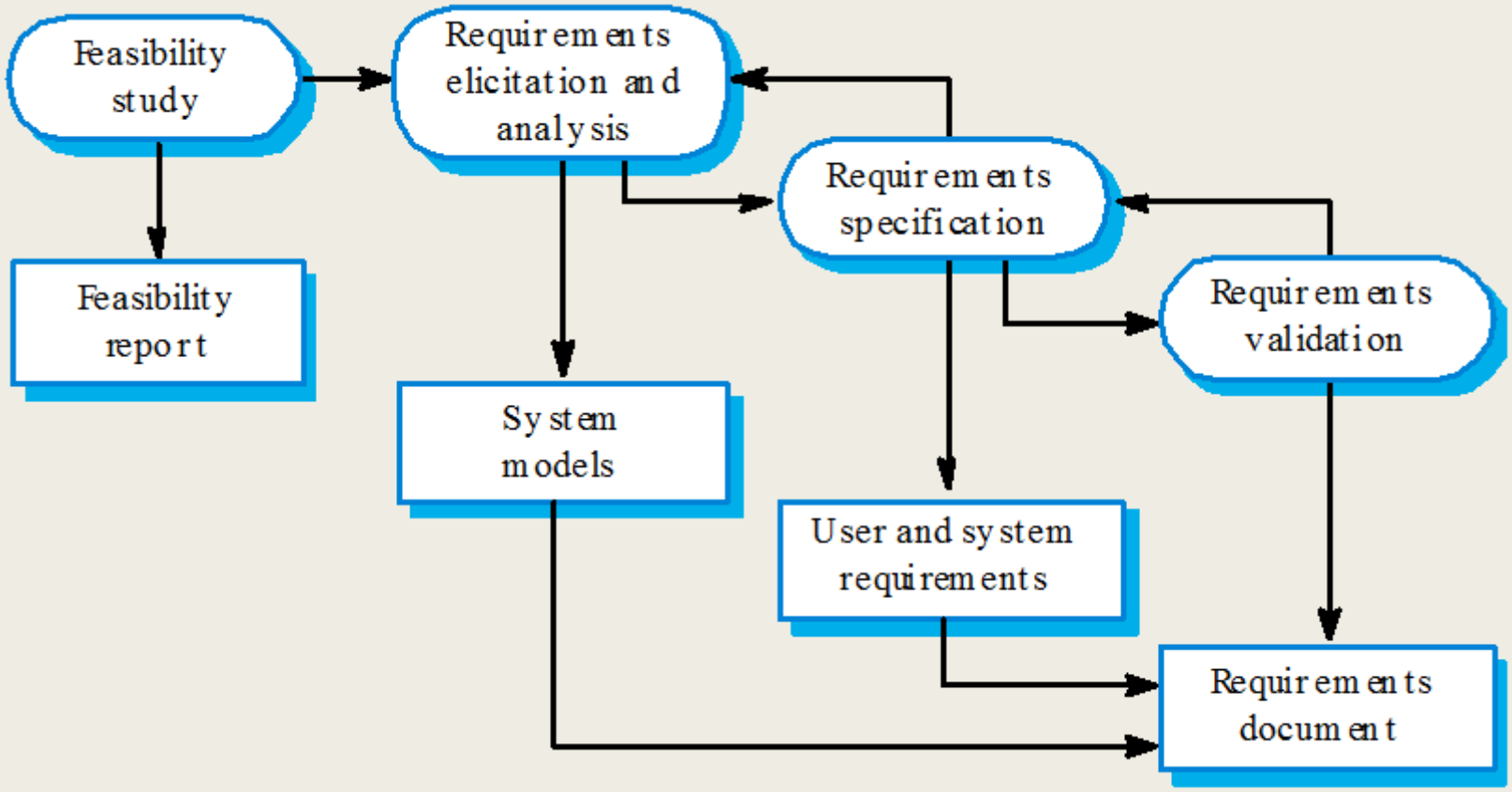
REQUIREMENT ENGINEERING

The Requirement Engineering Process

- The process of establishing what services are required and the constraints on the system's operation and development
- Requirements engineering help software engineers to better understand the problem they will work to solve. It encompasses the set of tasks that lead to an understanding of what the business impact of the software will be, what the customer wants and how end-users will interact with the software.

- **Requirement Engineering Process**
  - *Feasibility Study*
  - *Requirements elicitation and analysis*
  - *Requirements Specification*
  - *Requirements Validation*

The Requirements Engineering Process



Requirements Elicitation

- It is the practice of obtaining the requirements of a system from users, customers and other stakeholders. The practice is also sometimes referred to as **Requirement gathering**.
- Requirements elicitation practice include the following:
  - *Interviews*
  - *Questionnaires*
  - *User observation*
  - *Workshops*
  - *Brain storming*
  - *Use cases*
  - *Role playing*
  - *And prototyping*

Requirements Elicitation

- **Problems of Requirement Elicitation**
  - ***Problems of scope:*** *The boundary of system is ill-defined. Or unnecessary details are provided.*
  - ***Problems of understanding:*** *The users are not sure of what they need, and don't have full understanding of the problem domain.*
  - ***Problems of volatility:*** *the requirements change overtime.*

Requirements Elicitation

- **Guidelines of Requirements Elicitation**
  - *Assess the business and technical feasibility for the proposed system*
  - *Identify the people who will help specify requirements.*
  - *Define the technical environment (e.g. computing architecture, operating system, telecommunication needs) into which the system or product will be placed*
  - *Identify "domain constraints" (i.e. characteristics of the business environment specific to the application domain) that limit the functionality or performance of the system or product to build*
  - *Define one or more requirements elicitation methods (e.g. interviews, team meetings, ..etc)*
  - *Solicit participation from many people so that requirements are defined from different point of views.*
  - *Create usage scenarios of use cases to help customers/ users better identify key requirements.*

Requirements Analysis

- Requirements Analysis, determining whether the stated requirements are clear, complete, consistent and unambiguous.

Requirements Analysis

- **Stakeholder Identification**
  - *Stakeholders are people or organizations that have a valid interest in the system. They may be affected by it directly or indirectly.*
  - *Stake holders may include:*
    - Anyone who operates the system
    - Anyone who benefits from the system
    - Anyone involved in purchasing or procuring the system
    - People opposed to the system (negative stakeholders)
    - Organizations responsible for the system

Requirements Analysis

- **Stakeholder Interviews**
  - *Interviews are a common technique used in requirement analysis.*
  - *This technique can serve as a means of obtaining the highly focused knowledge from different stakeholders perspectives*

Requirements Analysis

- **Types of Requirements:**
  - *Customer Requirements:*
    - Operational distribution or deployment: Where will the system be used?
    - Mission profile or scenario: How will the system accomplish its mission objective?
    - Performance and related parameters: What are the critical system parameters to accomplish the mission?
    - Utilization environments: how are the various system components to be used?
    - Effectiveness requirements: How effective or efficient must the system be in performing its mission?
    - Operational life cycle: How long will the system be in use by the user?
    - Environment: what environments will the system be expected to operate in an effective manner?

Requirements Analysis

- **Types of Requirements:**
  - *Architectural Requirements:*
    - A formal description and representation of a system, organized in a way that support reasoning about the structure of the system which comprises system components, the externally visible properties of those components, the relationships and the behavior between them, and provides a plan from which products can be procured and systems developed, that will work together to implement the overall system.

Requirements Analysis

- **Types of Requirements:**
  - *Functional Requirements:*
    - Defines functions of a software system or its components. They may be calculations, technical details, data manipulation and processing and other specific functionality that define “what a system is supposed to accomplish?”
    - They describe particular results of a system.
    - Functional requirements are supported by Non-functional requirements.

Requirements Analysis

- **Types of Requirements:**
  - *Non-Functional Requirements:*
    - They are requirements that specify criteria that can be used to judge the operation of a system, rather than specific behavior.
    - Functional requirements define what the system is supposed to ***do***, whereas non-functional requirements define how a system is supposed to ***be***.
    - Non-functional requirements can be divided into two main categories:
      - *Execution qualities, such as security and usability, which are observable at runtime.*
      - *Evolution qualities, such as testability, maintainability and scalability.*

Requirements Specifications

- Requirements Specification is the direct result of a requirement analysis and can refer to:
  - *Software Requirements Specification*
  - *Hardware Requirements Specification*

Understand and specifying requirements

- A problem of scale
  - ***For small scale:*** *understand and specifying requirements is easy*
  - ***For large scale:*** *very hard; probably the hardest, most problematic and error prone*

- The requirements task:
  - ***Input:*** *User needs in minds of people (hopefully)*
  - ***Output****: precise statement of what the future system will do*

Challenges

- Identifying and specifying requirements
  - *Necessarily involves people interaction*
  - *Cannot be automated*

Background..

- What is a Requirement?
  - *A condition or capability that must be possessed by a system (IEEE)*
- What is the work product of the Requirement phase ?
  - *A software requirements specification (SRS) document*
- What is an SRS ?
  - *A complete specification of what the proposed system should do!*

Challenges.

- Requirements understanding is hard
  - *Visualizing a future system is difficult*
  - *Capability of the future system not clear, hence needs not clear*
  - *Requirements change with time*
  - *…*
- Essential to do a proper analysis and specification of requirements

Purpose of SRS document?

- SRS establishes basis of agreement between the user and the supplier.
  - *Users needs have to be satisfied, but user may not understand software*
  - *Developers will develop the system, but may not know about problem domain*

- SRS is
  - *the medium to bridge the communications gap, and*
  - *specifies user needs in a manner both can understand*

Need for SRS…

- Helps user understand his needs.
  - *users do not always know their needs*
  - *must analyze and understand the potential*
  - *The requirement process helps clarify needs*

- SRS provides a reference for validation of the final product
  - *Clear understanding about what is expected.*
  - *Validation - “ SW satisfies the SRS “*

Need for SRS…

- High quality SRS essential for high Quality SW
  - *Requirement errors get manifested in final sw*
  - *To satisfy the quality objective, must begin with high quality SRS*
  - *Requirements defects cause later problems*
    - 25% of all defects in one study; 54% of all defects found after user testing
    - defects often found in previously approved SRS.

Need for SRS…

- Good SRS reduces the development cost
  - *SRS errors are expensive to fix later*
  - *Req. changes can cost a lot (up to 40%)*
  - *Good SRS can minimize changes and errors*
  - *Substantial savings; extra effort spent during req. saves multiple times that effort*

- An Example
  - *Cost of fixing errors in req. , design , coding , acceptance testing and operation are 2 , 5 , 15 , 50 , 150 person-months*

Definition of SRS

- A software requirements specification (SRS) is a document that captures complete description about how the system is expected to perform. It is usually signed off at the end of requirements engineering phase.

Qualities of SRS

- Correct
- Unambiguous
- Complete
- Consistent
- Ranked for importance and/or stability
- Verifiable
- Modifiable
- Traceable

Types of Requirements

Requirements Validation and Verification

- Validation (& Verification), is the process of checking whether the requirements, as identified, do not contradict the expectations about the system of various stakeholders and do not contradict each other.
- It is Requirements Quality Control

Validation Vs. Verification

- Validation: “Am I building the right product?” checking a work product against higher-level work products or authorities that frame this particular product.
  - *Requirements are validated by stakeholders*

- Verification: “Am I building the product right?” checking a work product against some standards and conditions imposed on this type of product and the process of its development.
  - *Requirements are verified by the analysts mainly*