

Software Engineering-BSCE-301L

Module 3:

Modeling Requirements

Dr . Saurabh Agrawal

Faculty Id: 20165

School of Computer Science and Engineering

VIT, Vellore-632014

Tamil Nadu, India

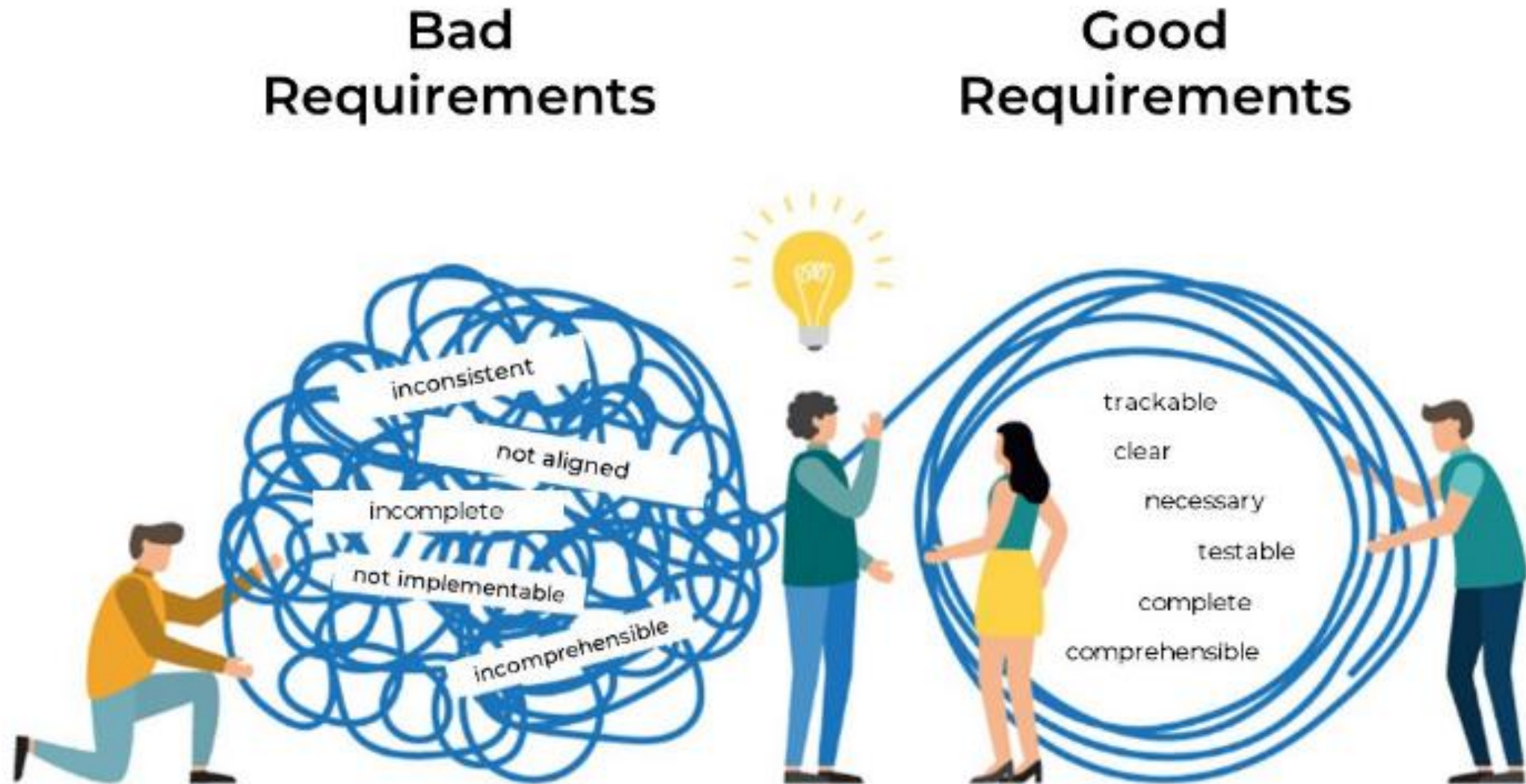
Outline

- ❑ Software Requirements and its Types
- ❑ Requirements Engineering Process
- ❑ Requirement Elicitation
- ❑ System Modeling – Requirements Specification and Requirement Validation
- ❑ Requirements Elicitation techniques
- ❑ Requirements management in Agile

Software Requirements and its Types

- ❑ The requirement can be defined as a high-level abstract statement or a detailed mathematical functional specification of a system's services, functions, and constraints.
- ❑ They are depictions of the characteristics and functionalities of the target system.
- ❑ Requirements denote the expectations of users from the software product.
- ❑ The requirement should be open to interpretation and detailed enough to understand.
- ❑ It is essential to know about software requirements because it minimizes the developer's time and effort and the development cost.
- ❑ We have also discussed requirement engineering and the process in one of our articles.

Software Requirements and its Types



Types of software requirements

Business requirements	User requirements	Software requirements
<p>Outline measurable goals for the business.</p> <hr/> <p>Define the <i>why</i> behind a software project.</p> <hr/> <p>Match project goals to stakeholder goals.</p> <hr/> <p>Maintain a BRD with requirements, updates or changes.</p>	<p>Reflect specific user needs or expectations.</p> <hr/> <p>Describe the <i>who</i> of a software project.</p> <hr/> <p>Highlight how users interact with it.</p> <hr/> <p>Create a URS, or make them part of the BRD.</p>	<p>Identify features, functions, non-functional requirements and use cases.</p> <hr/> <p>Delve into the <i>how</i> of a software project.</p> <hr/> <p>Describe software as functional modules and non-functional attributes.</p> <hr/> <p>Compose an SRS, and, optionally, an FRS.</p>

Software Requirements and its Types

❑ Business requirements

❑ Business needs drive many software projects.

❑ A business requirements document (BRD) outlines measurable project goals for the business, users and other stakeholders.

❑ Business analysts, leaders and other project sponsors create the BRD at the start of the project. This document defines the *why* behind the build.

❑ For software development contractors, the BRD also serves as the basis for more detailed document preparation with clients.

❑ A BRD is composed of one or more statements. No universally established format exists for BRD statements, but one common approach is to align goals:

❑ Write statements that match a project goal to a measurable stakeholder or business goal. The basic format of a BRD statement looks like:

"The [project name] software will [meet a business goal] in order to [realize a business benefit]."

Software Requirements and its Types

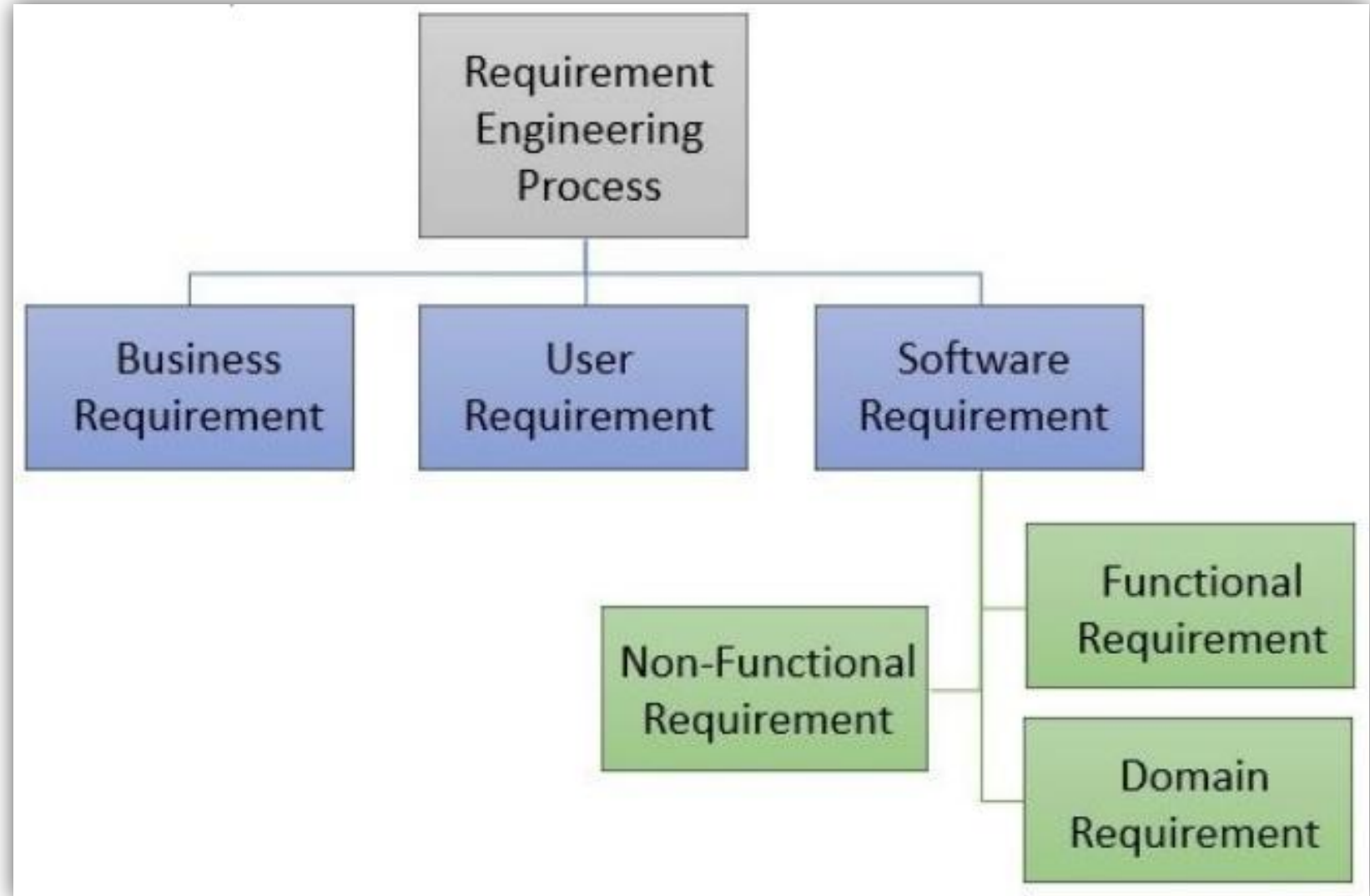
❑ User Requirements

- ❑ User requirements reflect the specific needs or expectations of the software's customers.
- ❑ Organizations sometimes incorporate these requirements into a BRD, but an application that poses extensive user functionality or complex UI issues might justify a separate document specific to the needs of the intended user.
- ❑ User requirements -- much like user stories -- highlight the ways in which customers interact with software.
- ❑ There is no universally accepted standard for user requirements statements, but here's one common format:
"The [user type] shall [interact with the software] in order to [meet a business goal or achieve a result]."

Software Requirements and its Types

❑ **Software Requirements:** There are three types of Software requirements as follows:

1. Functional requirements
2. Non-Functional requirements
3. Domain requirements



Software Requirements and its Types

❑ **Functional requirements** are such software requirements that are demanded explicitly as basic facilities of the system by the end-users.

❑ So, these requirements for functionalities should be necessarily incorporated into the system as a part of the contract.

❑ They describe system behavior under specific conditions, they are the functions that one can see directly in the final product, and it was the requirements of the users as well.

❑ It describes a software system or its components.

❑ These are represented as inputs to the software system, its behavior, and its output.

❑ It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform.

❑ A functional requirement can range from the high-level abstract statement of the sender's necessity to detailed mathematical functional requirement specifications.

1. Natural language

2. A structured or formatted language with no rigorous syntax and formal specification language with proper syntax.

Software Requirements and its Types

❑ **Non-functional Requirements (NFRs)** requirements are defined as the quality constraints that the system must satisfy to complete the project contract.

❑ But, the extent may vary to which implementation of these factors is done or get relaxed according to one project to another.

❑ They are also called non-behavioral requirements or quality requirements/attributes.

❑ They deal with issues like, Performance, Reusability, Flexibility, Reliability, Maintainability, Security, Portability

❑ **Non-Functional Requirements are classified into many types.**

- Interface Constraints
- Economic Constraints
- Operating Constraints
- Performance constraints: storage space, response time, security, etc.
- Life Cycle constraints: portability, maintainability, etc.

Software Requirements and its Types

❑ **Domain requirements** are the requirements related to a particular category like software, purpose or industry, or other domain of projects.

❑ Domain requirements can be functional or non-functional.

❑ These are essential functions that a system of specific domains must necessarily exhibit.

❑ The common factor for domain requirements is that they meet established standards or widely accepted feature sets for that category of the software project.

❑ Domain requirements typically arise in military, medical, and financial industry sectors.

❑ They are identified from that specific domain and are not user-specific.

1. **Software in medical equipment:** In medical equipment, software must be developed per IEC 60601 regarding medical electrical equipment's basic safety and performance. The software can be functional and usable but not acceptable for production because it fails to meet domain requirements.

2. **An Academic Software:** Such software must be developed to maintain records of an institute efficiently. Domain requirement of such software is the functionality of being able to access the list of faculty and list of students of each grade.

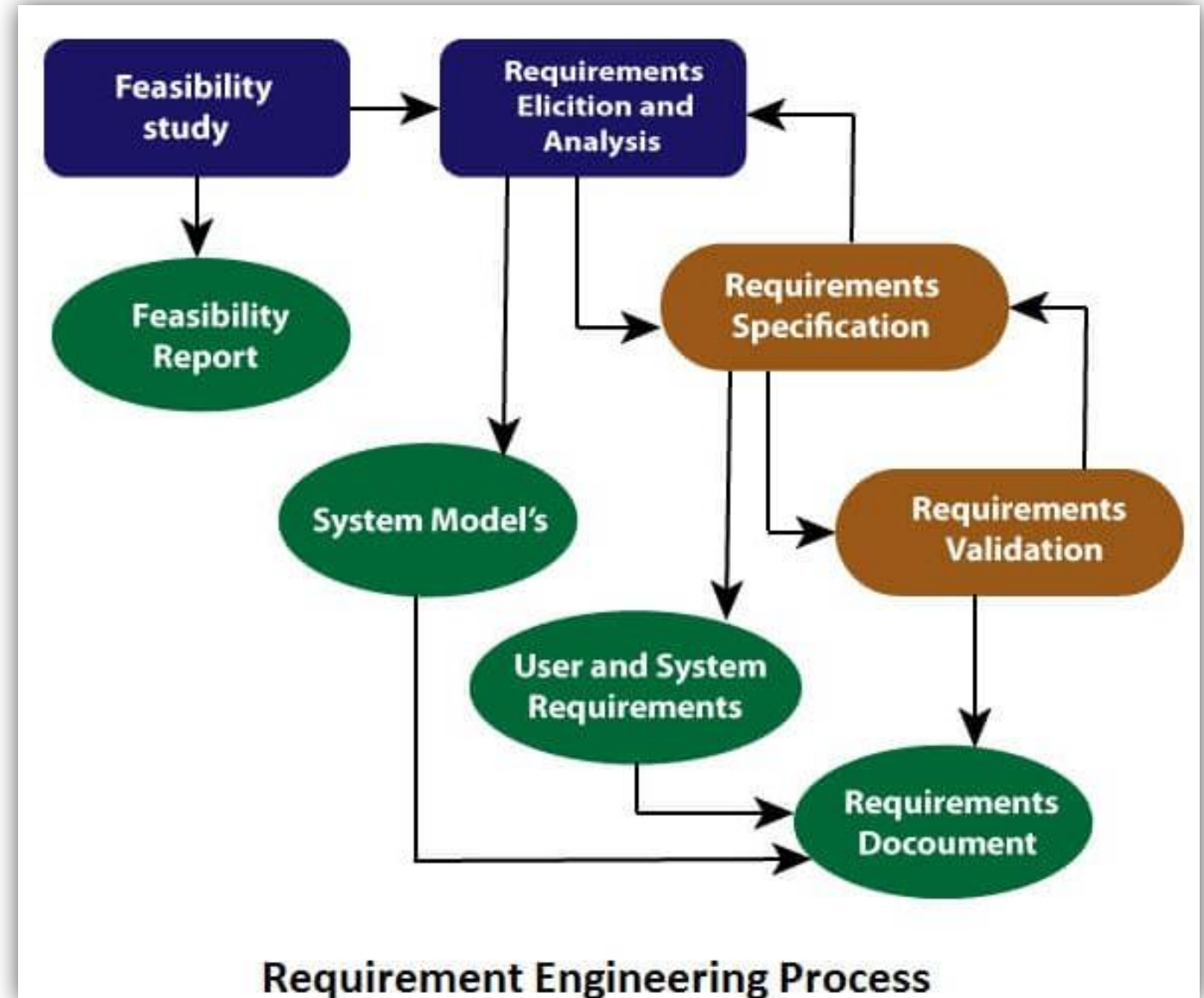
Requirement Engineering Process

- ❑ **Requirements engineering (RE)** refers to the process of defining, documenting, and maintaining requirements in the engineering design process.
- ❑ 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.
- ❑ Thus, requirement engineering is the disciplined application of proven principles, methods, tools, and notation to describe a proposed system's intended behavior and its associated constraints.

Requirement Engineering Process

□ **Requirements engineering (RE)** is a four-step process, which includes:

1. Feasibility Study
2. Requirement Elicitation and Analysis
3. Software Requirement Specification
4. Software Requirement Validation
5. Software Requirement Management



Requirement Engineering Process

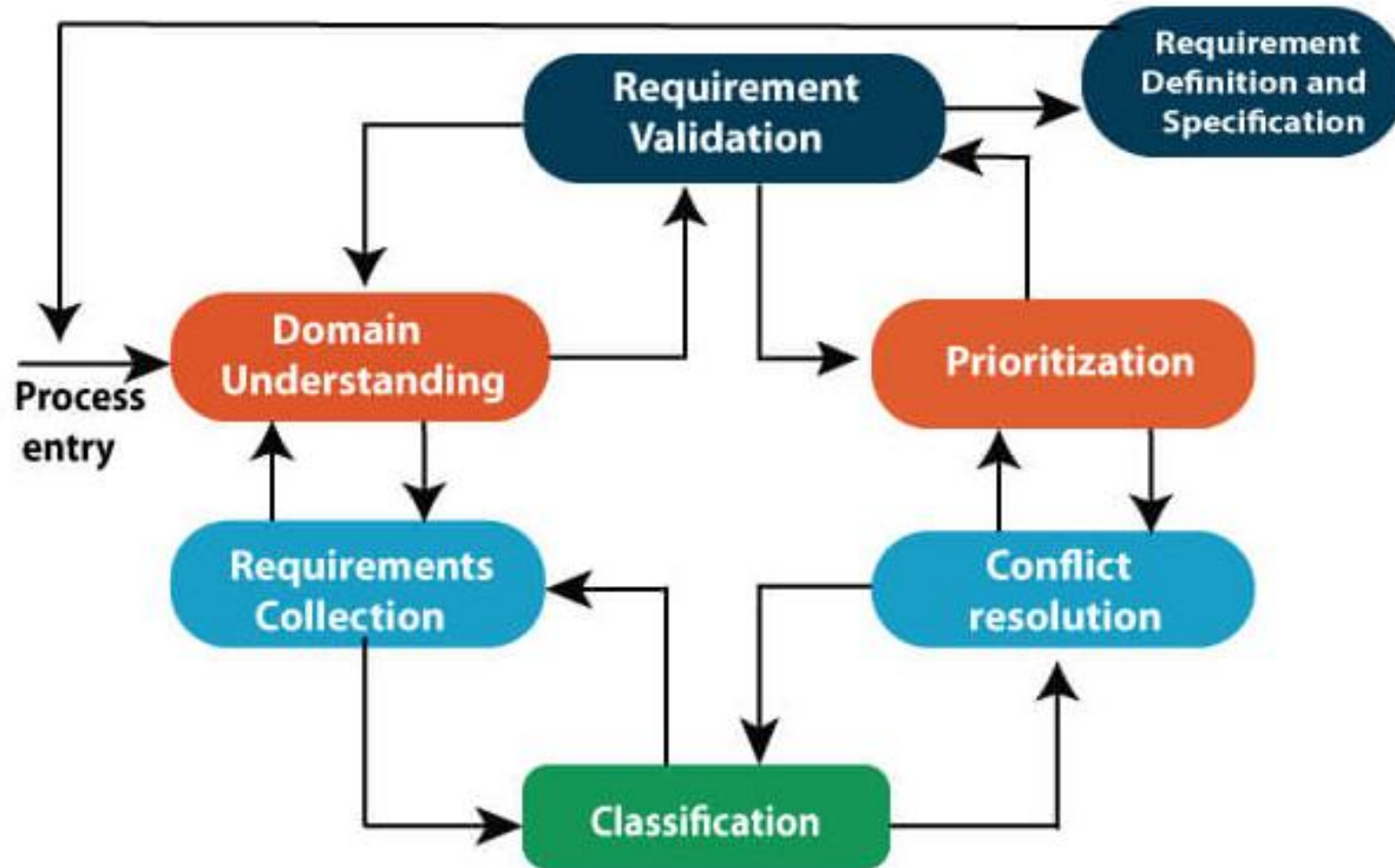
1. Feasibility Study: The objective behind the feasibility study is to create the reasons for developing the software that is acceptable to users, flexible to change and conformable to established standards.

- I. Technical Feasibility:** evaluates the current technologies, which are needed to accomplish customer requirements within the time and budget.
- II. Operational Feasibility:** assesses the range in which the required software performs a series of levels to solve business problems and customer requirements.
- III. Economic Feasibility:** decides whether the necessary software can generate financial profits for an organization.

2. Requirement Elicitation and Analysis: This is also known as the **gathering of requirements**.

- Here, requirements are identified with the help of customers and existing systems processes, if available.
- Analysis of requirements starts with requirement elicitation.
- The requirements are analyzed to identify inconsistencies, defects, omission, etc.
- We describe requirements in terms of relationships and also resolve conflicts if any.
- **Following are the problems of Elicitation and Analysis:**
 - I. Getting all, and only, the right people involved.
 - II. Stakeholders often don't know what they want
 - III. Stakeholders express requirements in their terms.
 - IV. Stakeholders may have conflicting requirements.
 - V. Requirement change during the analysis process.
 - VI. Organizational and political factors may influence system requirements.

Elicitation and Analysis Process



3. Software Requirement 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 requirement received by the customer written in ordinary language.
- It is the job of the analyst to write the requirement in technical language so that they can be understood and beneficial by the development team.
- The models used at this stage include ER diagrams, data flow diagrams (DFDs), function decomposition diagrams (FDDs), data dictionaries, etc.

3. Software Requirement Specification:

- I. **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. The DFD is also known as a data flow graph or bubble chart.
- II. **Data Dictionaries:** Data Dictionaries are simply repositories to store information about all data items defined in DFDs. At the requirements stage, the data dictionary should at least define customer data items, to ensure that the customer and developers use the same definition and terminologies.
- III. **Entity-Relationship Diagrams:** Another tool for requirement specification is the entity-relationship diagram, often called an "*E-R diagram*." 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.

4. Software Requirement Validation:

- After requirement specifications developed, the requirements discussed in this document are validated.
- The user might demand illegal, impossible solution or experts may misinterpret the needs.
- Requirements can be checked against the following conditions:
 - If they can practically implement
 - If they are correct and as per the functionality and specialty of software
 - If there are any ambiguities
 - If they are full
 - If they can describe

4. Software Requirement Validation:

■ Following are Requirements Validation Techniques

- I. **Requirements reviews/inspections:** systematic manual analysis of the requirements.
- II. **Prototyping:** Using an executable model of the system to check requirements.
- III. **Test-case generation:** Developing tests for requirements to check testability.
- IV. **Automated consistency analysis:** checking for the consistency of structured requirements descriptions.

5. Software Requirement Management:

- Requirement management is the process of managing changing requirements during the requirements engineering process and system development.
- New requirements emerge during the process as business needs a change, and a better understanding of the system is developed.
- The priority of requirements from different viewpoints changes during development process.
- The business and technical environment of the system changes during the development.

Requirement Elicitation

❑ **Requirements elicitation** is the process of gathering and defining the requirements for a software system.

❑ The goal of requirements elicitation is to ensure that the software development process is based on a clear and comprehensive understanding of the customer's needs and requirements.

Requirement Elicitation

❑ **Requirement Elicitation:** Requirements elicitation is perhaps the most difficult, most error-prone, and most communication-intensive software development.

1. It can be successful only through an effective customer-developer partnership. It is needed to know what the users require.
2. Requirements elicitation involves the identification, collection, analysis, and refinement of the requirements for a software system.
3. It is a critical part of the software development life cycle and is typically performed at the beginning of the project.
4. Requirements elicitation involves stakeholders from different areas of the organization, including business owners, end-users, and technical experts.
5. The output of the requirements elicitation process is a set of clear, concise, and well-defined requirements that serve as the basis for the design and development of the software system.

□ Importance of Requirements Elicitation:

- 1. Compliance with Business Objectives:** The process of elicitation guarantees that the software development endeavours are in harmony with the wider company aims and objectives. Comprehending the business context facilitates the development of a solution that adds value for the company.
- 2. User Satisfaction:** It is easier to create software that fulfils end users needs and expectations when they are involved in the requirements elicitation process. Higher user pleasure and acceptance of the finished product are the results of this.
- 3. Time and Money Savings:** Having precise and well-defined specifications aids in preventing miscommunication and rework during the development phase. As a result, there will be cost savings and the project will be completed on time.

□ Importance of Requirements Elicitation:

- 4. Compliance and Regulation Requirements:** Requirements elicitation is crucial for projects in regulated industries to guarantee that the software conforms with applicable laws and norms. In industries like healthcare, finance, and aerospace, this is crucial.
- 5. Traceability and Documentation:** Throughout the software development process, traceability is based on requirements that are well-documented. Traceability helps with testing, validation, and maintenance by ensuring that every part of the software can be linked to a particular requirement.

Requirement Elicitation

❑ **Requirements Elicitation Activities:** Requirements elicitation includes the subsequent activities. A few of them are listed below:

1. Knowledge of the overall area where the systems are applied.
2. The details of the precise customer problem where the system is going to be applied must be understood.
3. Interaction of system with external requirements.
4. Detailed investigation of user needs.
5. Define the constraints for system development.

Requirement Elicitation

❑ **Requirements Elicitation Methods:** There are a number of requirements elicitation methods. Few of them are listed below:

1. Interviews
2. Brainstorming Sessions
3. Facilitated Application Specification Technique
4. Quality Function Deployment
5. Use Case Approach

□Interviews:

- Objective of conducting an interview is to understand the customer's expectations from the software.
- It is impossible to interview every stakeholder hence representatives from groups are selected based on their expertise and credibility. Interviews maybe be open-ended or structured.
 - I. In open-ended interviews there is no pre-set agenda. Context free questions may be asked to understand the problem.
 - II. In a structured interview, an agenda of fairly open questions is prepared. Sometimes a proper questionnaire is designed for the interview.

Requirement Elicitation

□ Brainstorming Sessions

- I. It is a group technique
- II. It is intended to generate lots of new ideas hence providing a platform to share views
- III. A highly trained facilitator is required to handle group bias and group conflicts.
- IV. Every idea is documented so that everyone can see it.
- V. Finally, a document is prepared which consists of the list of requirements and their priority if possible.

Requirement Elicitation

□ Facilitated Application Specification Technique

- Its objective is to bridge the expectation gap – the difference between what the developers think they are supposed to build and what customers think they are going to get.
- A team-oriented approach is developed for requirements gathering.
- Each attendee is asked to make a list of objects that are:
 - I. Part of the environment that surrounds the system.
 - II. Produced by the system.
 - III. Used by the system.
- Each participant prepares his/her list, different lists are then combined, redundant entries are eliminated, team is divided into smaller sub-teams to develop mini-specifications and finally a draft of specifications is written down using all the inputs from the meeting.

Requirement Elicitation

□ **Quality Function Deployment:** In this technique customer satisfaction is of prime concern, hence it emphasizes on the requirements which are valuable to the customer.

3 types of requirements are identified:

- I. **Normal requirements:** In this the objective and goals of the proposed software are discussed with the customer. Example – normal requirements for a result management system may be entry of marks, calculation of results, etc.
- II. **Expected requirements:** These requirements are so obvious that the customer need not explicitly state them. Example – protection from unauthorized access.
- III. **Exciting requirements:** It includes features that are beyond customer's expectations and prove to be very satisfying when present. Example – when unauthorized access is detected, it should backup and shutdown all processes.

Requirement Elicitation

❑ **Use Case Approach:** This technique combines text and pictures to provide a better understanding of the requirements.

❑ The use cases describe the 'what', of a system and not 'how'.

❑ Hence, they only give a functional view of the system.

❑ The components of the use case design include three major things – Actor, use cases, use case diagram.

I. **Actor:** It is the external agent that lies outside the system but interacts with it in some way. An actor maybe a person, machine etc. It is represented as a stick figure. Actors can be primary actors or secondary actors.

- **Primary actors:** It requires assistance from the system to achieve a goal.
- **Secondary actor:** It is an actor from which the system needs assistance.

II. **Use cases:** They describe the sequence of interactions between actors and the system. They capture who(actors) do what(interaction) with the system. A complete set of use cases specifies all possible ways to use the system.

III. **Use case diagram:** A use case diagram graphically represents what happens when an actor interacts with a system. It captures the functional aspect of the system.

- A stick figure is used to represent an actor.
- An oval is used to represent a use case.
- A line is used to represent a relationship between an actor and a use case.

Requirement Elicitation

□ Steps Of Requirements Elicitation:

- I. Identify all the stakeholders, e.g., Users, developers, customers etc.
- II. List out all requirements from customer.
- III. A value indicating degree of importance is assigned to each requirement.
- IV. In the end the final list of requirements is categorized as:
 - It is possible to achieve.
 - It should be deferred and the reason for it.
 - It is impossible to achieve and should be dropped off.

Requirement Elicitation

□ Steps Of Requirements Elicitation:

- I. Identify all the stakeholders, e.g., Users, developers, customers etc.
- II. List out all requirements from customer.
- III. A value indicating degree of importance is assigned to each requirement.
- IV. In the end the final list of requirements is categorized as:
 - It is possible to achieve.
 - It should be deferred and the reason for it.
 - It is impossible to achieve and should be dropped off.

Requirement Elicitation

□ Advantages of Requirements Elicitation:

1. **Clear requirements:** Helps to clarify and refine customer requirements.
2. **Improves communication:** Improves communication and collaboration between stakeholders.
3. **Results in good quality software:** Increases the chances of developing a software system that meets customer needs.
4. **Avoids misunderstandings:** Avoids misunderstandings and helps to manage expectations.
5. **Supports the identification of potential risks:** Supports the identification of potential risks and problems early in the development cycle.
6. **Facilitates development of accurate plan:** Facilitates the development of a comprehensive and accurate project plan.
7. **Increases user confidence:** Increases user and stakeholder confidence in the software development process.

❑ Disadvantages of Requirements Elicitation:

1. **Time consuming:** Can be time-consuming and expensive.
2. **Skills required:** Requires specialized skills and expertise.
3. **Impacted by changing requirements:** May be impacted by changing business needs and requirements.
4. **Impacted by other factors:** Can be impacted by political and organizational factors.
5. **Lack of commitment from stakeholders:** Can result in a lack of buy-in and commitment from stakeholders.
6. **Impacted by conflicting priorities:** Can be impacted by conflicting priorities and competing interests.
7. **Sometimes inaccurate requirements:** May result in incomplete or inaccurate requirements if not properly managed.

Note for Students

□ This power point presentation is for lecture, therefore it is suggested that also utilize the text books and lecture notes.