

# 22CSC51 - AGILE METHODOLOGIES

**Prepared By,**

**Mr.N.Aravindhraj,**

Assistant Professor

Department.of CSE

Kongu Engineering College

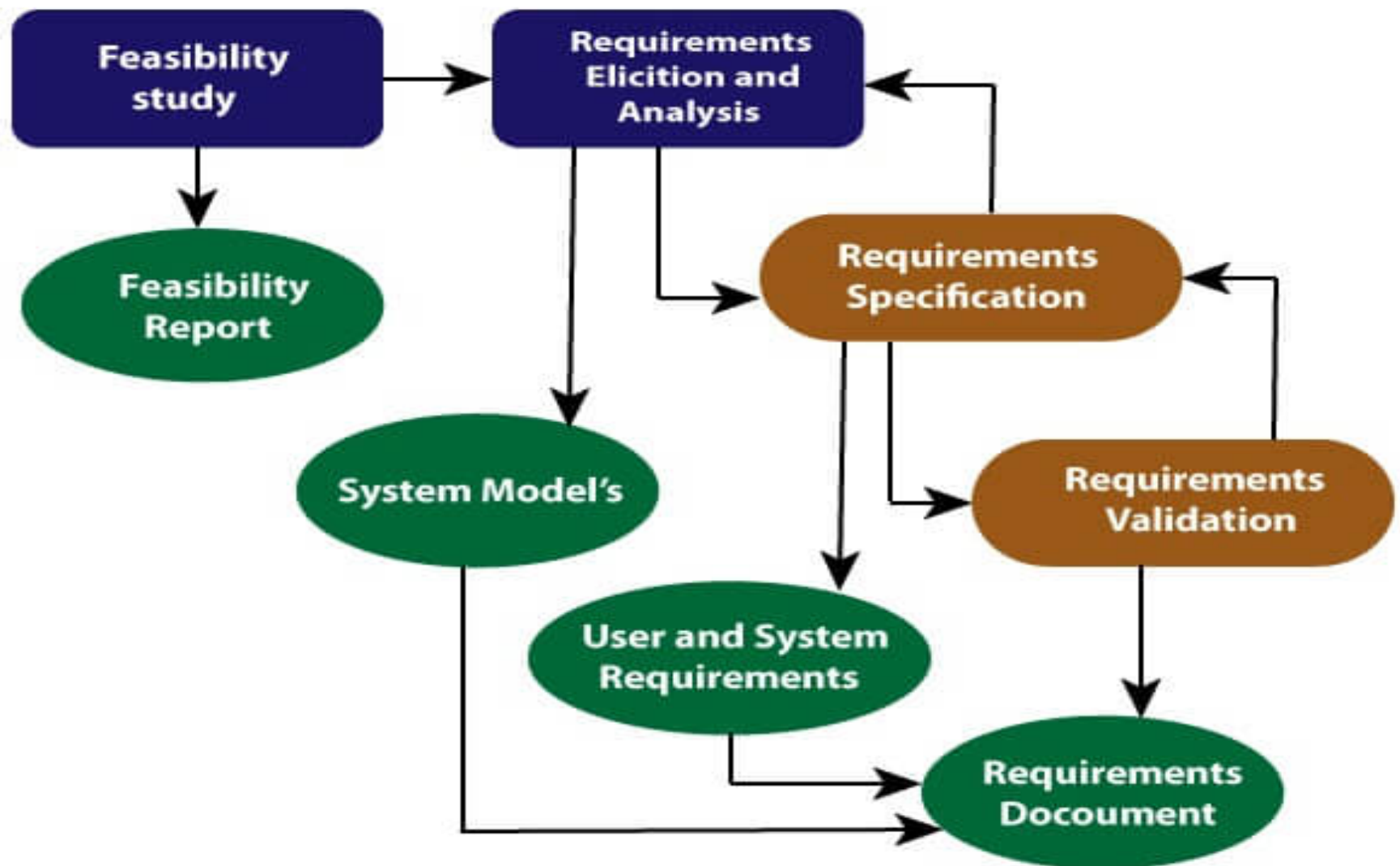
# Requirement Engineering

- Requirements Engineering (RE) refers to the process of **defining, documenting, and maintaining requirements** in the engineering design process.

## Requirement Engineering Process

It is a four-step process, which includes -

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



**Requirement Engineering Process**

# 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.

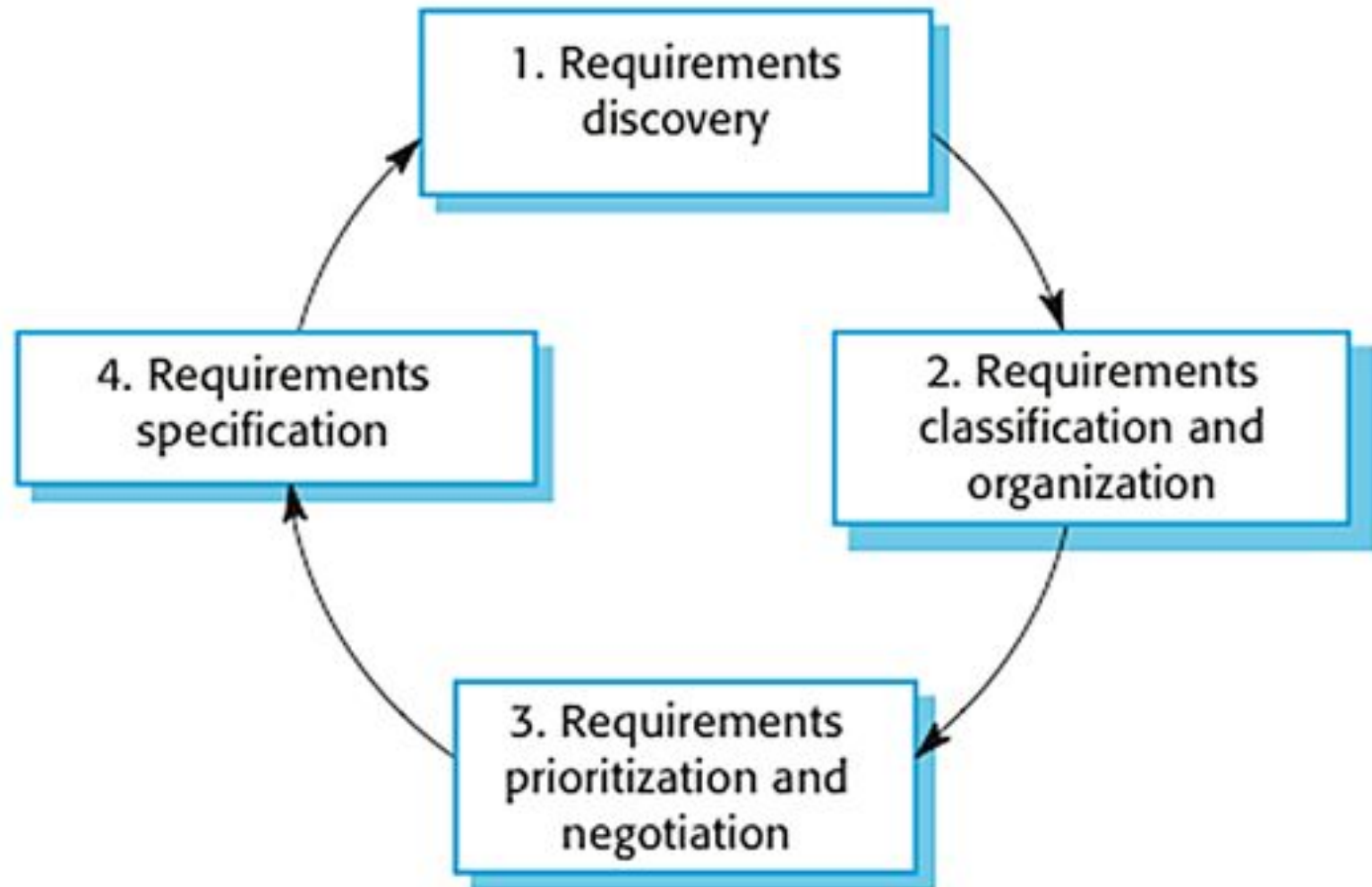
## Types of Feasibility:

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

# 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.**

# Requirement Elicitation and Analysis



# Software Requirements

Broadly software requirements should be categorized in two categories:

- Functional Requirements
- Non Functional Requirements

# Functional Requirements

Requirements, which are related to functional aspect of software fall into this category.

They define **functions and functionality** within and from the software system.

## Examples -

- Search option given to **user to search from various invoices**.
- User should be able to **mail any report** to management.
- Users can be divided into groups and groups can be given separate rights.
- Should comply business rules and administrative functions.
- Software is developed keeping **downward compatibility** intact.



# Non-Functional Requirements

Requirements, which are not related to functional aspect of software, fall into this category. They are **implicit or expected characteristics of software**, which users make assumption of.

Non-functional requirements include -

- **Security**
- **Logging**
- **Storage**
- **Configuration**
- **Performance**
- **Cost**
- **Interoperability**
- **Flexibility**
- **Disaster recovery**
- **Accessibility**

# Requirement Elicitation and Analysis

- Requirement Elicitation Techniques
- Interviews
- Surveys
- Questionnaires
- Task analysis
- Domain Analysis
- Brainstorming
- Prototyping
- Observation

# Requirements Analysis

- Requirements analysis or requirements engineering is a process used to **determine the needs and expectations of a new product**.
- It involves frequent **communication** with the **stakeholders and end-users** of the product to define expectations, resolve conflicts, and document all the key requirements.

# Requirements Analysis Process

- A requirements analysis process involves the following steps:
  - Identify Key Stakeholders and End-Users
  - Capture Requirements
  - Categorize Requirements
  - Interpret and Record Requirements
  - Sign off

# Requirements Analysis Process

## Identify Key Stakeholders and End-Users:

- ✓ The first step of the requirements analysis process is to **identify key stakeholders** who are the main sponsors of the project.
- ✓ They will have the final say on what should be included in the scope of the project.
- ✓ Next, identify the **end-users of the product**. Since the product is intended to satisfy their needs, their inputs are equally important.

# Requirements Analysis Process

## Capture Requirements:

Ask each of the stakeholders and end-users their requirements for the new product. some of the requirements analysis techniques are,

1. Hold One-on-One Interviews
2. Use Focus Groups
3. Utilize Use Cases
4. Build Prototypes

# Requirements Analysis Process

## Categorize Requirements:

Since requirements can be of various types, they should be grouped to avoid confusion. Requirements are usually divided into four categories:

**Functional Requirements** - Functions the product is required to perform.

**Technical Requirements** - Technical issues to be considered for the successful implementation of the product.

**Transitional Requirements** - Steps required to implement a new product smoothly.

**Operational Requirements** - Operations to be carried out in the backend for proper functioning of the product.

# Requirements Analysis Process

## Interpret and Record Requirements

Once the requirements are **categorized**, determine which requirements are actually achievable and document each one of them. some techniques to analyze and interpret requirements are,

**Define Requirements Precisely**

**Prioritize Requirements**

**Carry Out an Impact Analysis**

**Resolve Conflicts**

**Analyze Feasibility**



# Requirements Analysis Process

## Sign off

- Once a final decision is made on the requirements, **ensure that you get a signed agreement from the key stakeholders.**
- This is done to ensure that there are **no changes or uncontrolled growth** in the scope of the project.

# 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.

# Software Requirement Specification

The models used at this stage include

- **Data Flow Diagrams:** Data Flow Diagrams (DFDs) are used widely for **modeling the requirements**. DFD shows the flow of data through a system.
- **Data Dictionaries:** Data Dictionaries are **simply repositories** to store information about all data items defined in DFDs.
- **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.**

# Software Requirement Validation

- After requirement specifications developed, the requirements discussed in this **document are validated.**
- Requirements can be the check against the following conditions -
  - ✓ If they can practically implement
  - ✓ If they are correct and as per the functionality and specially of software
  - ✓ If there are any ambiguities
  - ✓ If they are full
  - ✓ If they can describe

# Software Requirement Validation

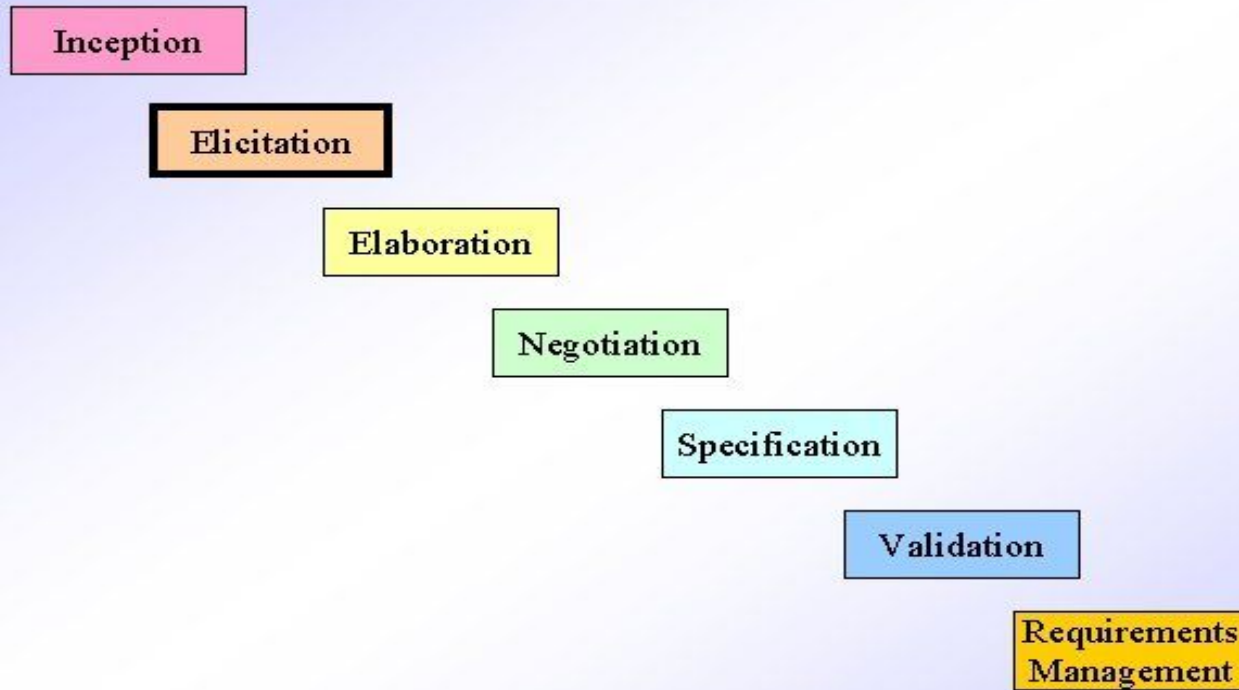
## Requirements Validation Techniques

- **Requirements reviews/inspections:** 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.
- **Automated consistency analysis:** checking for the consistency of structured requirements descriptions.

# Software Requirement Management

- Requirement management is the process of **managing changing requirements** during the requirements engineering process and system development.

# Requirement Engineering task



# Requirement Engineering task

- ❖ ***Inception*** —Establish a basic understanding of the problem and the nature of the solution.
- ❖ ***Elicitation*** —Draw out the requirements from stakeholders.
- ❖ ***Elaboration*** —Create an analysis model that represents information, functional, and behavioral aspects of the requirements.
- ❖ ***Negotiation*** —Agree on a deliverable system that is realistic for developers and customers.
- ❖ ***Specification*** —Describe the requirements formally or informally.
- ❖ ***Validation***—*Review* the requirement specification for errors, ambiguities, omissions, and conflicts.
- ❖ ***Requirements management***—Manage changing requirements.



# Inception Task

The requirement engineer *ask a set of question* to establish

- ❖ basic understanding of the problem
- ❖ the people who want a solution
- ❖ the nature of the solution that is desired, and
- ❖ the effectiveness of preliminary communication and collaboration
- ❖ between the customer and the developer

***Through out the question , requirement engineer needs to***

- ❖ Identify the stakeholder
- ❖ Recognize multiple view points
- ❖ Work towards collaboration
- ❖ Break the ice and initiate the communication

# Elicitation task

*Ask the customer, the users and others*

- ❖ what the objectives for the system or product are,
- ❖ what is to be accomplished,
- ❖ how the system or product fits into the needs of the business.
- ❖ How the system or product to be used on day to day basis

*Following are the problems that are encountered during elicitation*

- ❖ **Problem of scope**
- ❖ **Problem of understanding**
- ❖ **Problems of volatility**

To overcome the above problem , we must approach the *requirement gathering in an organized way*

# Elaboration

- ❖ The information obtained from the customer during inception and elicitation is expanded and refined it
- ❖ Elaboration focuses on developing a refined technical model of software functions , features, and constraints
- ❖ It is an analysis modeling task
  - Use cases are developed
  - Domain classes are identified
  - State machine diagrams are used

# Negotiation

- ❖ Customers and users are ask for more than can be achieved ,given limited business resources
- ❖ It is common for different customers or users to propose conflicting requirements
- ❖ Reconciling the conflict through a process of negotiation
- ❖ Customers, users and other stakeholders are asked

**To rank/prioritizes the requirement**

**Assesses their cost**

**Risk**

**Addresses internal conflicts**

- ❖ So that requirements are eliminated, combined / modified both(Developer and customer) achieve some measure of satisfaction

# Specification

***Specification***— “*different things to different people*” can be any one (or more) of the following:

- A written document
- A set of models
- A formal mathematical model
- A collection of user scenarios (use-cases)
- A prototype

# Validation

- ❖ Product produced are assessed for quality during validation
- ❖ Requirement validation examines the specification to ensure all the SW requirements stated clearly, that inconsistencies, omissions and error have been detected and corrected
- ❖ The work product conform to the standards established for the process

***Validation*—a review mechanism that looks for**

- ❖ errors in content or interpretation
- ❖ areas where clarification may be required
- ❖ missing information
- ❖ inconsistencies (a major problem when large products or systems are engineered)
- ❖ conflicting or unrealistic (unachievable) requirements

# Requirement management

It is a **set of activities** that help the **project team identify, control and track the requirements** and changes to requirement at any time as the project proceeds