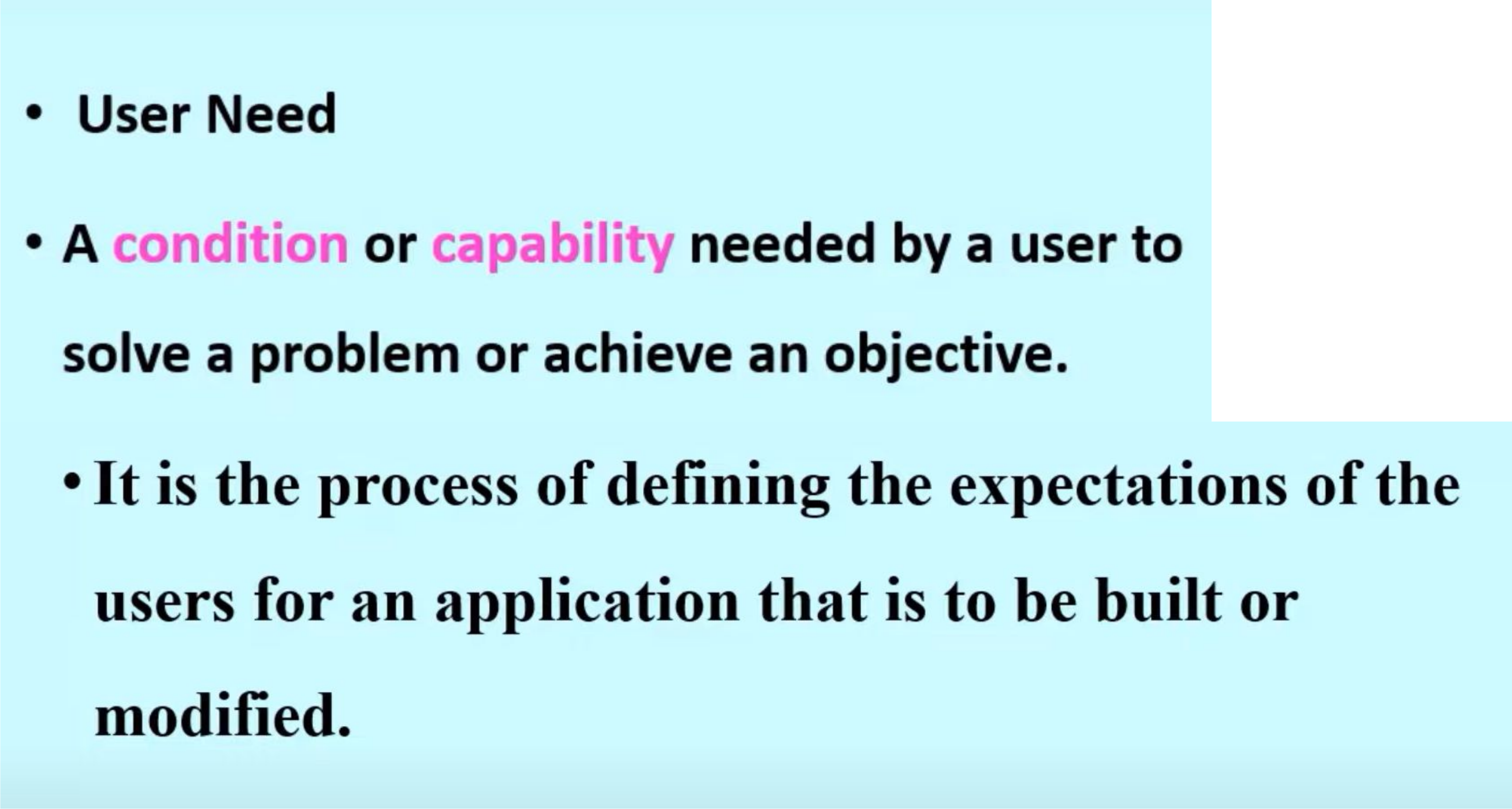
JGI JAIN
DEEMED-TO-BE UNIVERSITY

SOFTWARE ENGINEERING

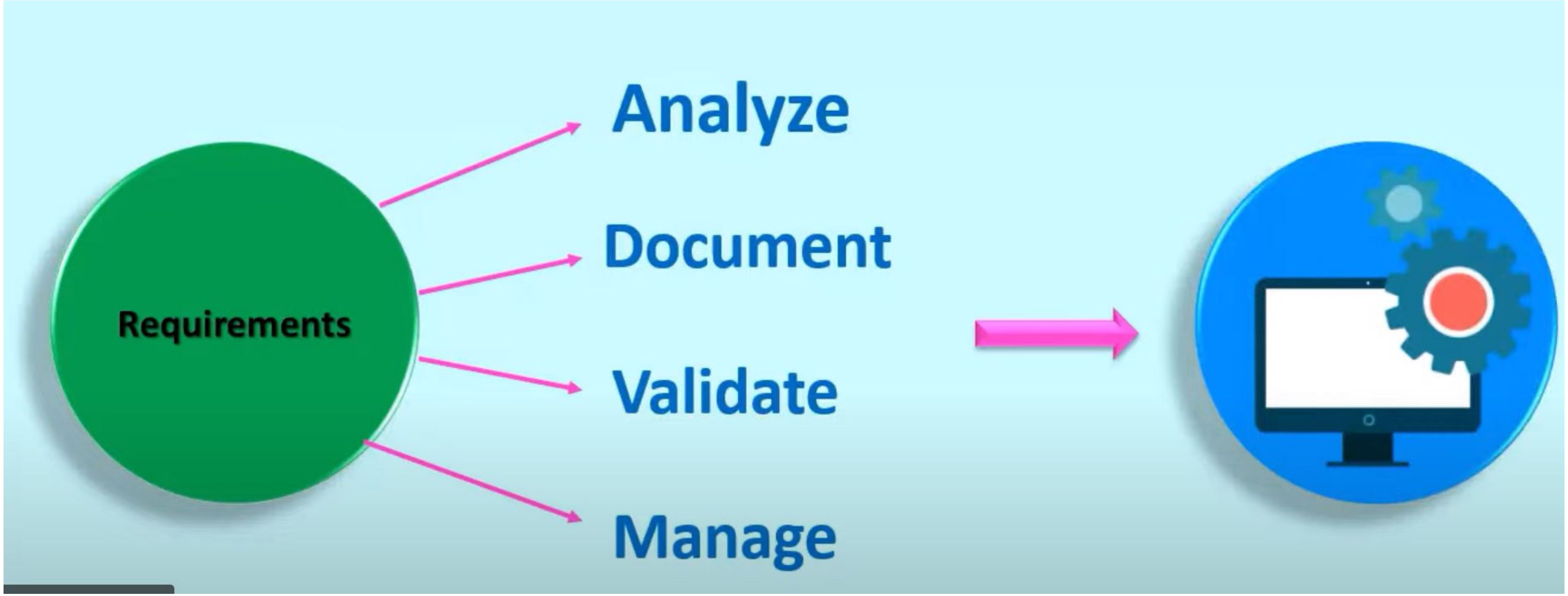
MODULE II

Dr.Preethi
Assistant Professor
School of CS and IT

Understanding Software Requirements: What are Software Requirements? Characteristics of Good Requirements, Functional and Non-Functional Requirements, Requirements Engineering Process: Feasibility Studies, Elicitation, Specification, Validation; Software Requirements Specification (SRS) document, IEEE Guidelines



- User Need
- A condition or capability needed by a user to solve a problem or achieve an objective.
- **It is the process of defining the expectations of the users for an application that is to be built or modified.**



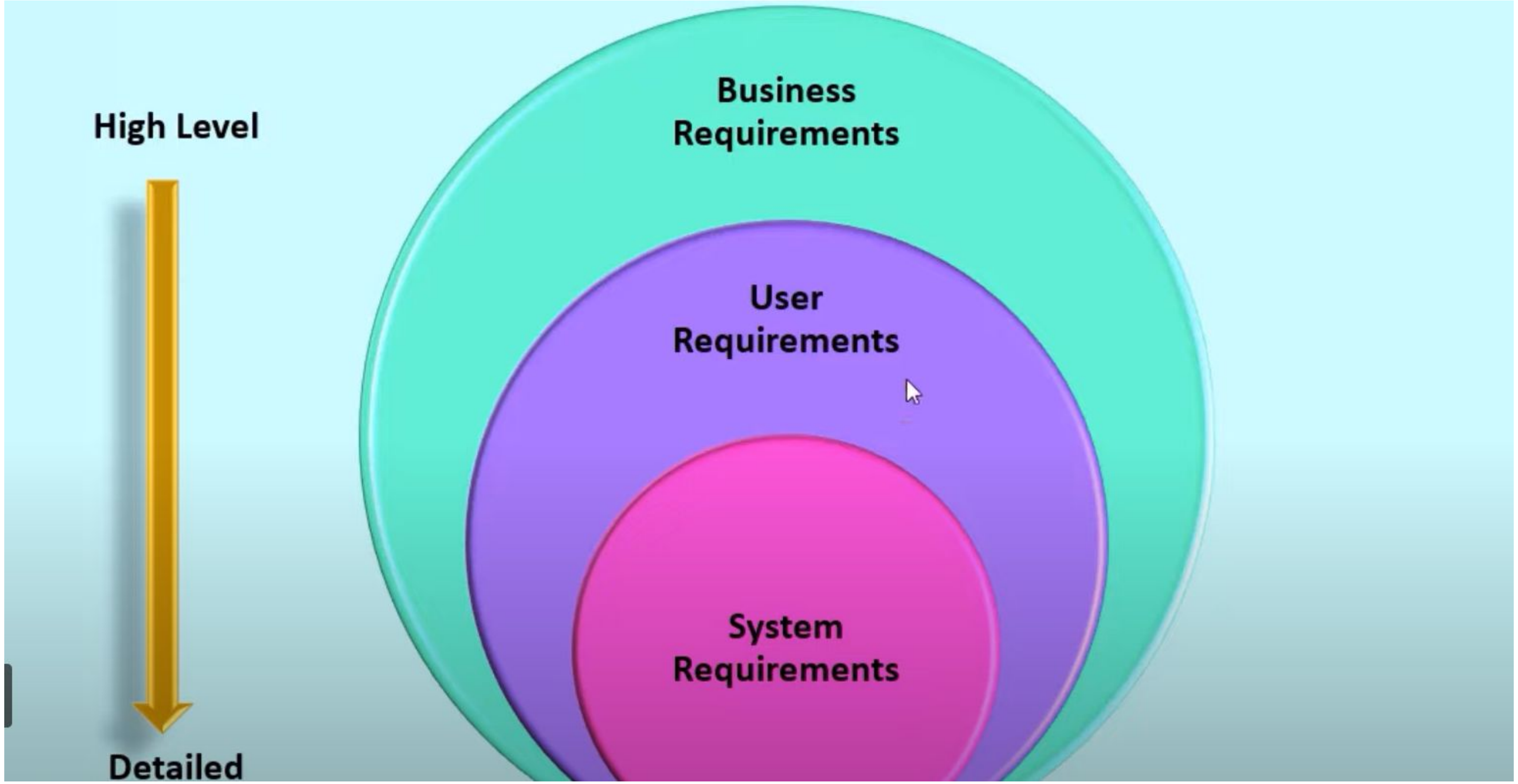
Requirements

Analyze

Document

Validate

Manage



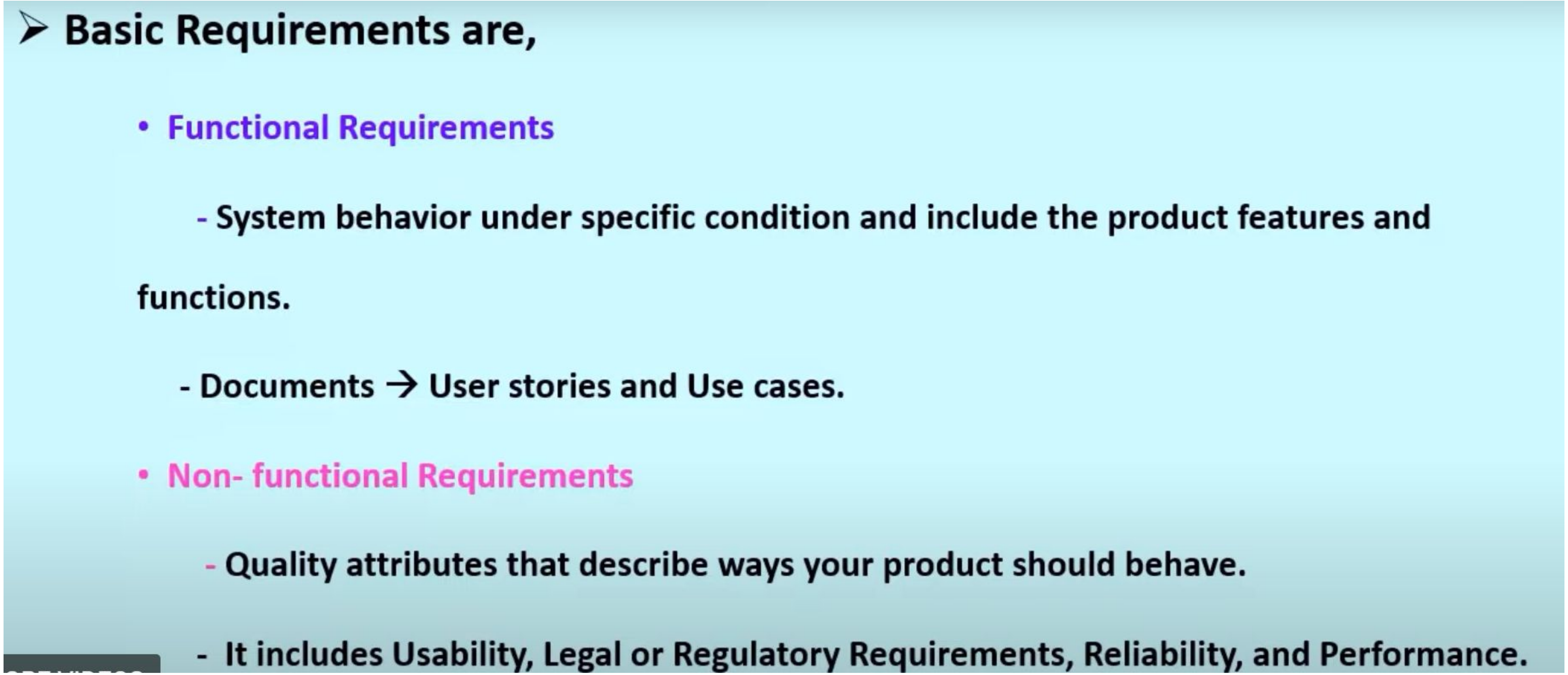
High Level

Detailed

Business Requirements

User Requirements

System Requirements



➢ Basic Requirements are,

- **Functional Requirements**

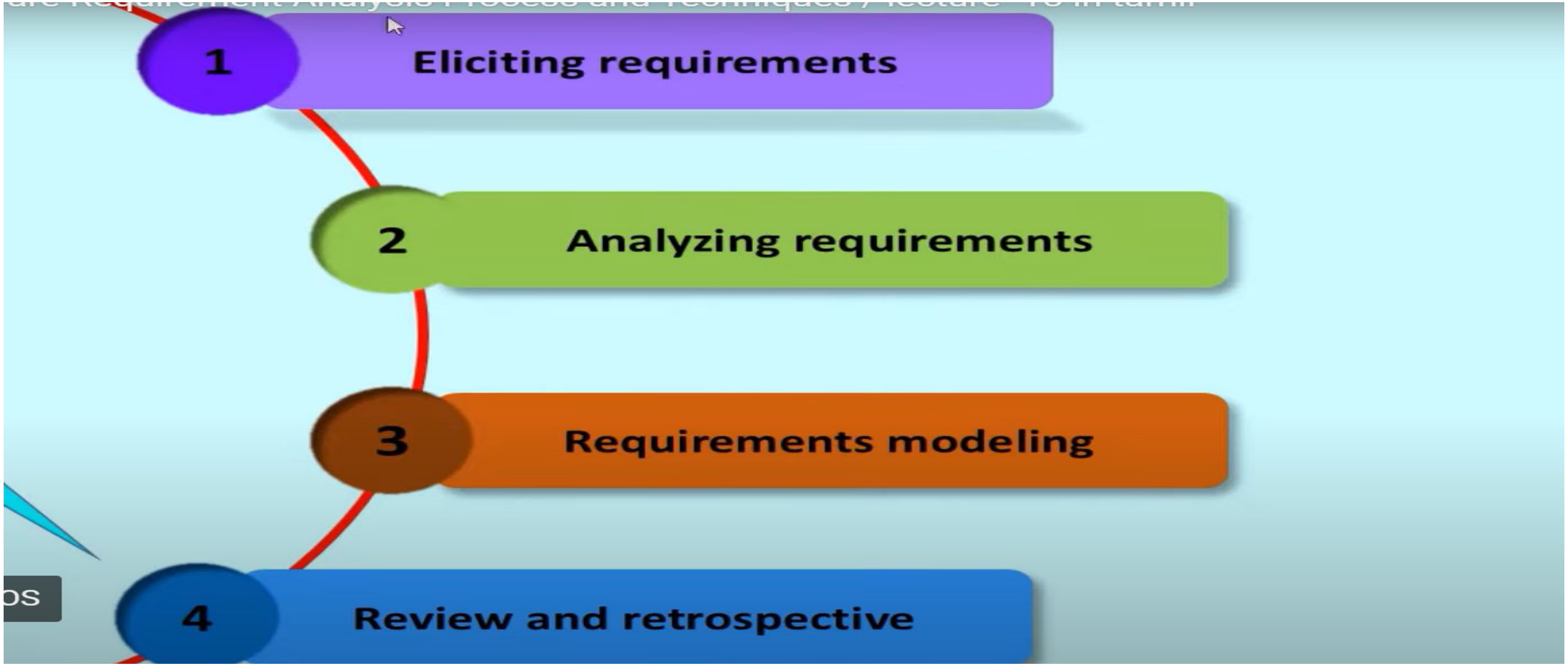
  - System behavior under specific condition and include the product features and functions.

  - Documents → User stories and Use cases.

- **Non- functional Requirements**

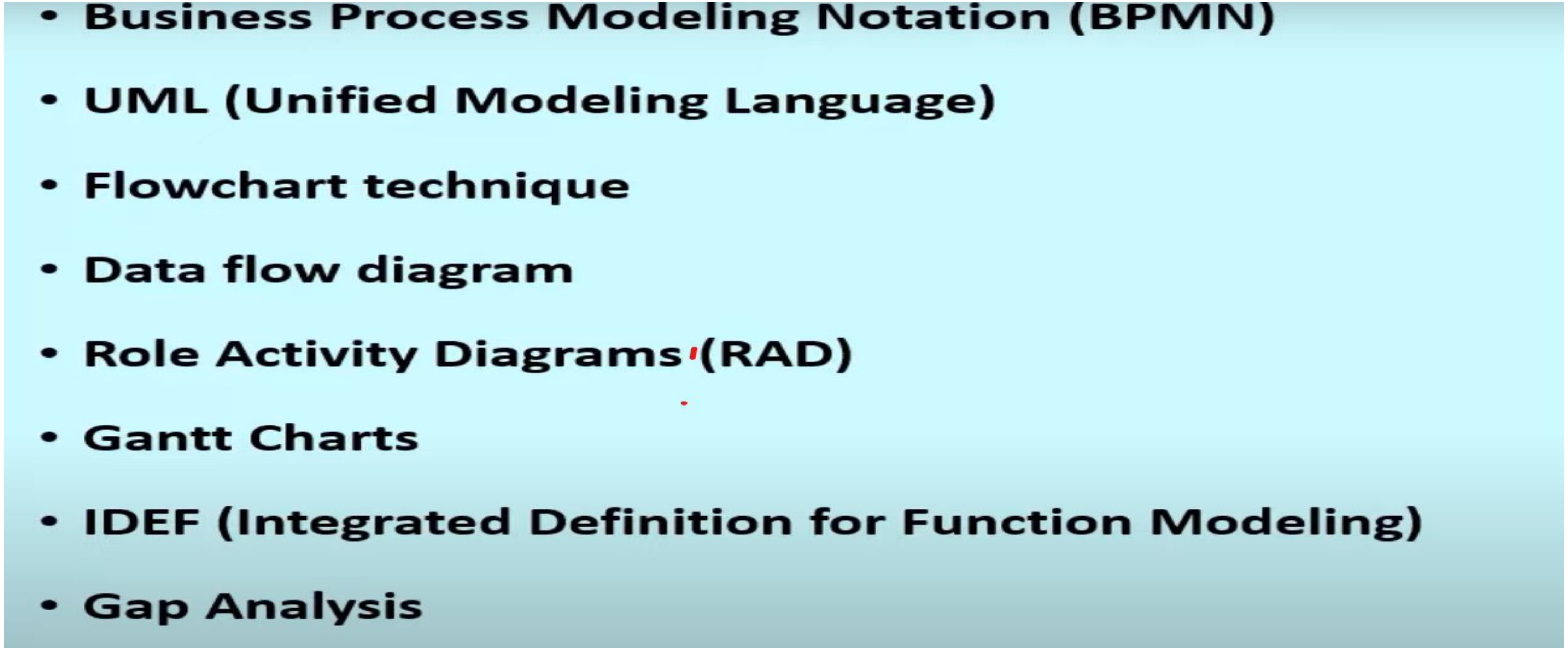
  - Quality attributes that describe ways your product should behave.

  - It includes Usability, Legal or Regulatory Requirements, Reliability, and Performance.



are Requirement Analysis Process and Techniques / lecture - 10 in tamil

1. Eliciting requirements
2. Analyzing requirements
3. Requirements modeling
4. Review and retrospective



- Business Process Modeling Notation (BPMN)
- UML (Unified Modeling Language)
- Flowchart technique
- Data flow diagram
- Role Activity Diagrams (RAD)
- Gantt Charts
- IDEF (Integrated Definition for Function Modeling)
- Gap Analysis

Software Characteristics

Different individuals judge software on different basis. This is because they are involved with the software in different ways. For example, users want the software to perform according to their requirements. Similarly, developers involved in designing, coding, and maintenance of the software evaluate the software by looking at its internal characteristics, before delivering it to the user. Software characteristics are classified into six major components.

- • **Functionality:** Refers to the degree of performance of the software against its intended purpose.
- • **Reliability:** Refers to the ability of the software to provide desired functionality under the given conditions.
- • **Usability:** Refers to the extent to which the software can be used with ease.
- • **Efficiency:** Refers to the ability of the software to use system resources in the most effective and efficient manner.
- • **Maintainability:** Refers to the ease with which the modifications can be made in a software system to extend its functionality, improve its performance, or correct errors.
- • **Portability:** Refers to the ease with which software developers can transfer software from one platform to another, without (or with minimum) changes. In simple terms, it refers to the ability of software to function properly on different hardware and software platforms without making any changes in it.
- In addition to the above mentioned characteristics, robustness and integrity are also important. **Robustness** refers to the degree to which the software can keep on functioning in spite of being provided with invalid data while **integrity** refers to the degree to which unauthorized access to the software or data can be prevented.

Requirements :

Services that the user expects from system –gathering the req- done by common with user.

Characteristics of Good Requirements:

1. Correctness : should cover over all requirements
2. Completeness: nothing should be left blank like page no; or any other information.
3. Consistency : no conflicts should be there for any set of requirements.
4. Unambiguous : no repetition of requirements should be there.
5. Modifiability : able to accept changes

6. Verifiability : some mechanism should be there for verify requirements.
7. Traceability : checking of requirements should be easy.
8. Design independent : option for selecting different design modules.
9. Testability : easy to generate test cases and test plans
10. Understandable by user : language should be easy

Functional and Non-Functional Requirements

Functional :

- Describe the functionality of system .

i.e. features provided by system to satisfy customer .

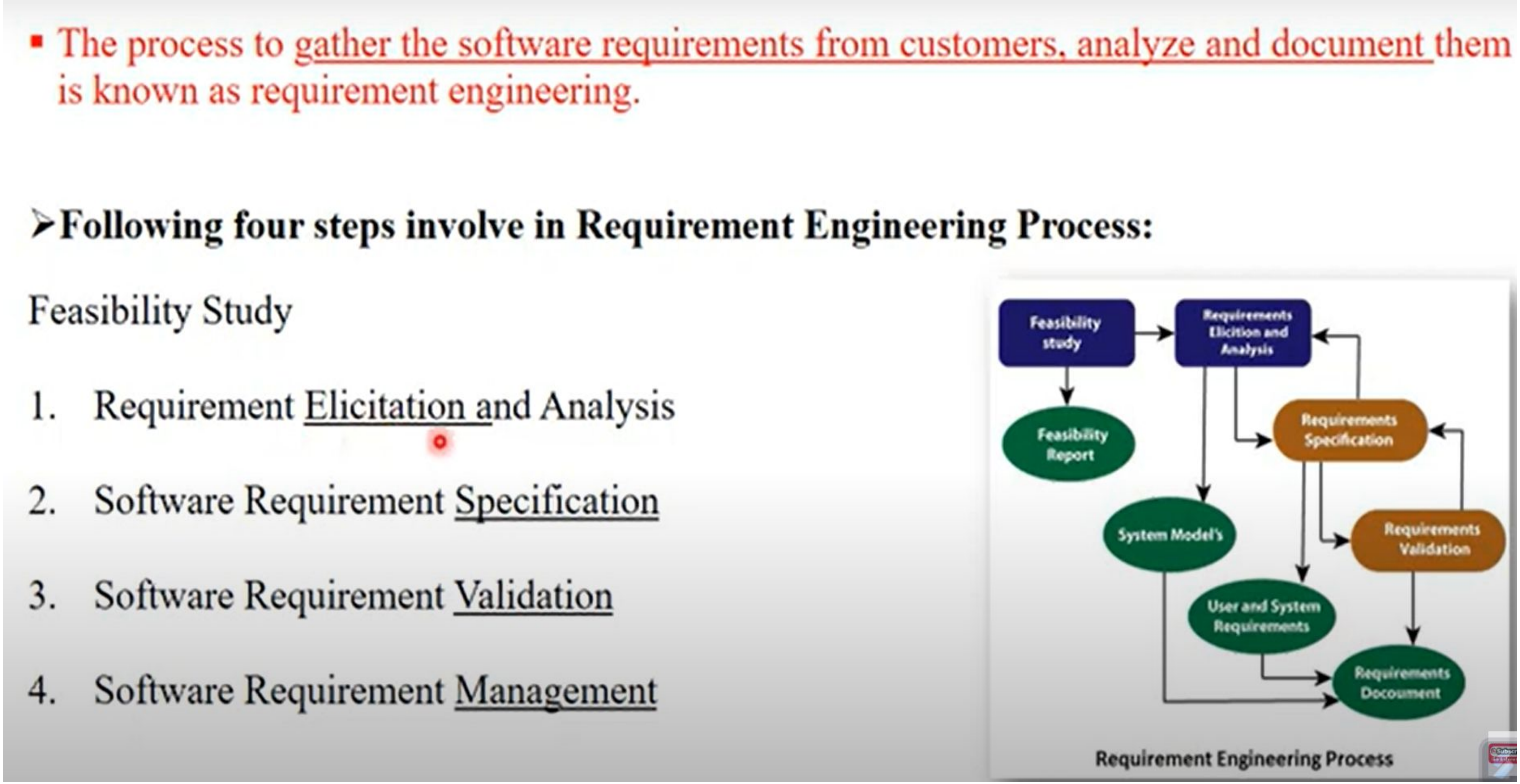
- Should be complete and consistent

Non – Functional requirements :

-non directly related to functionality of system .

-How features are provided.

- Not needed for user
- More important than functional



- The process to gather the software requirements from customers, analyze and document them is known as requirement engineering.

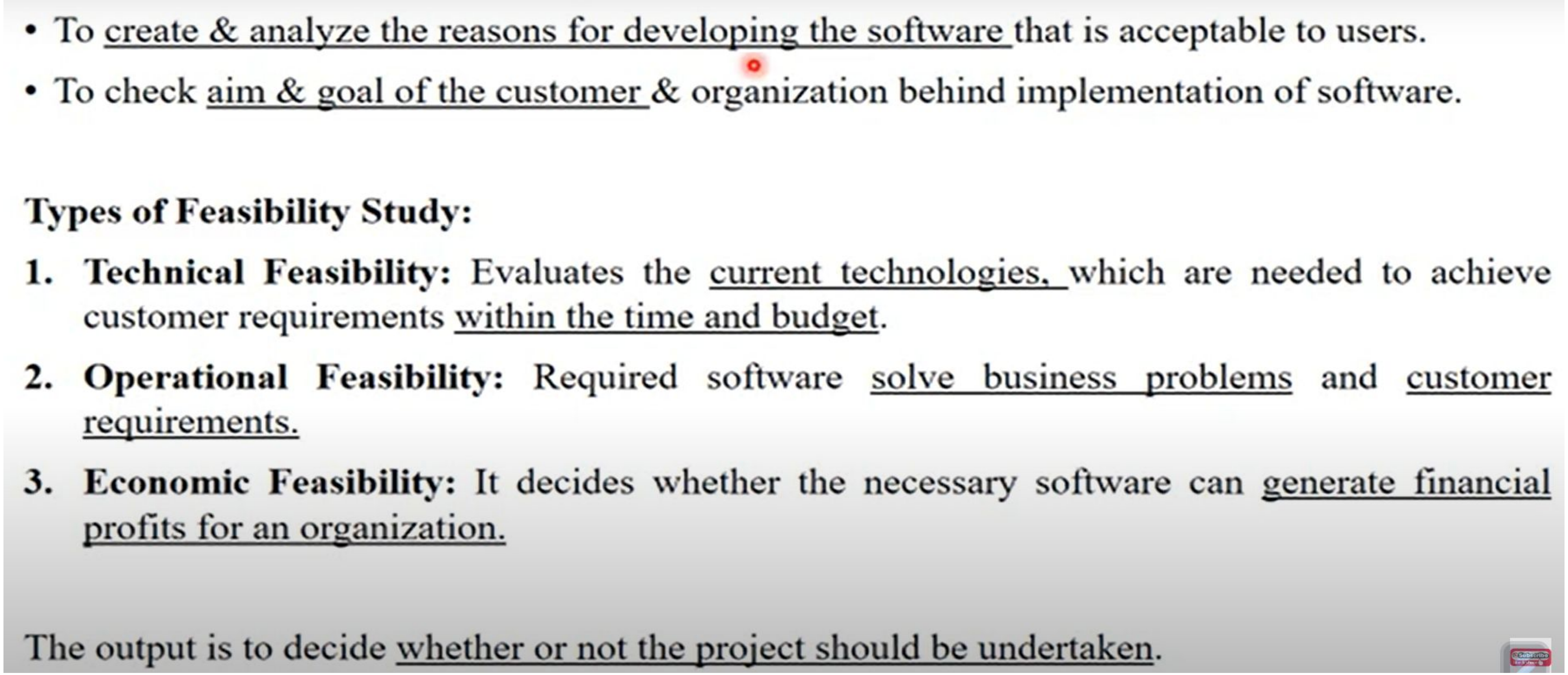
➢ **Following four steps involve in Requirement Engineering Process:**

Feasibility Study

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

Requirement Engineering Process

Feasibility study



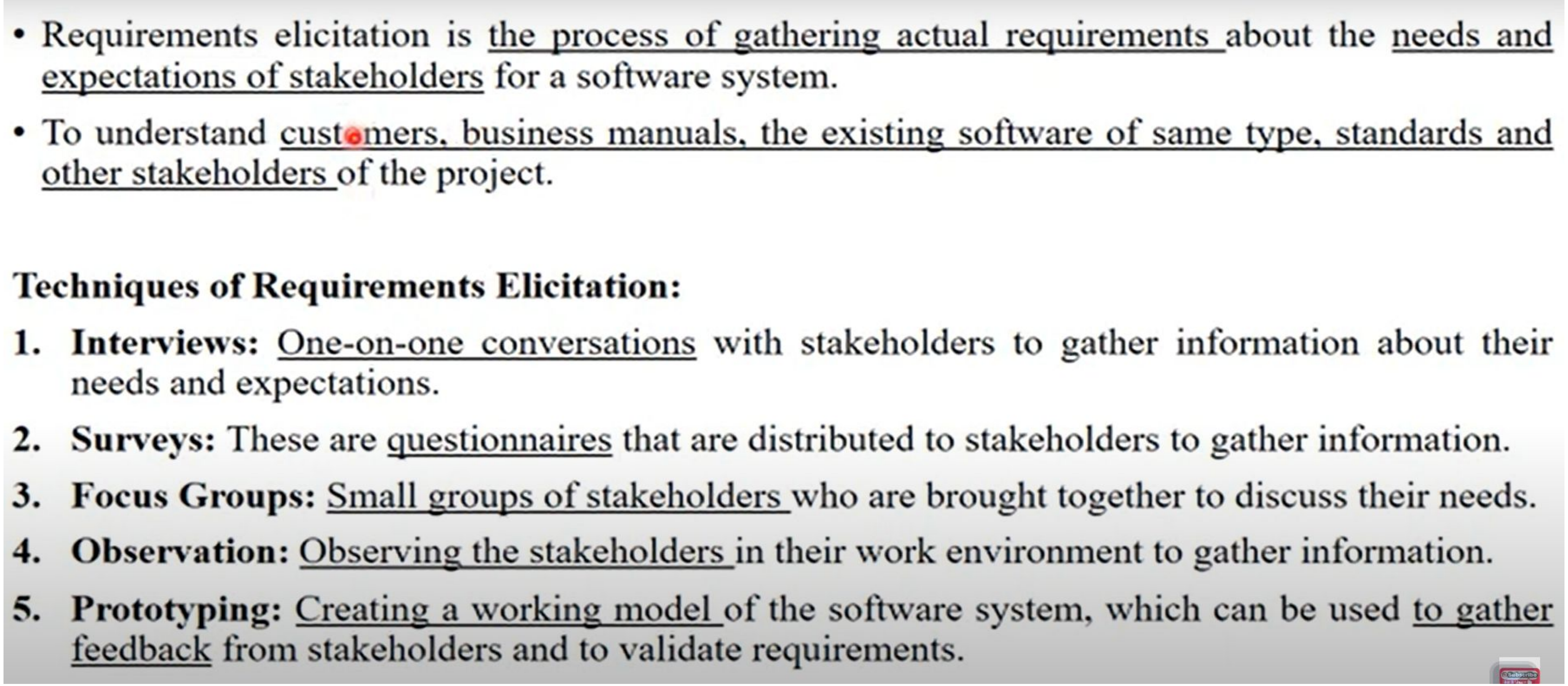
- To create & analyze the reasons for developing the software that is acceptable to users.
- To check aim & goal of the customer & organization behind implementation of software.

Types of Feasibility Study:

1. **Technical Feasibility:** Evaluates the current technologies, which are needed to achieve customer requirements within the time and budget.
2. **Operational Feasibility:** Required software solve business problems and customer requirements.
3. **Economic Feasibility:** It decides whether the necessary software can generate financial profits for an organization.

The output is to decide whether or not the project should be undertaken.

Requirement elicitation and analysis

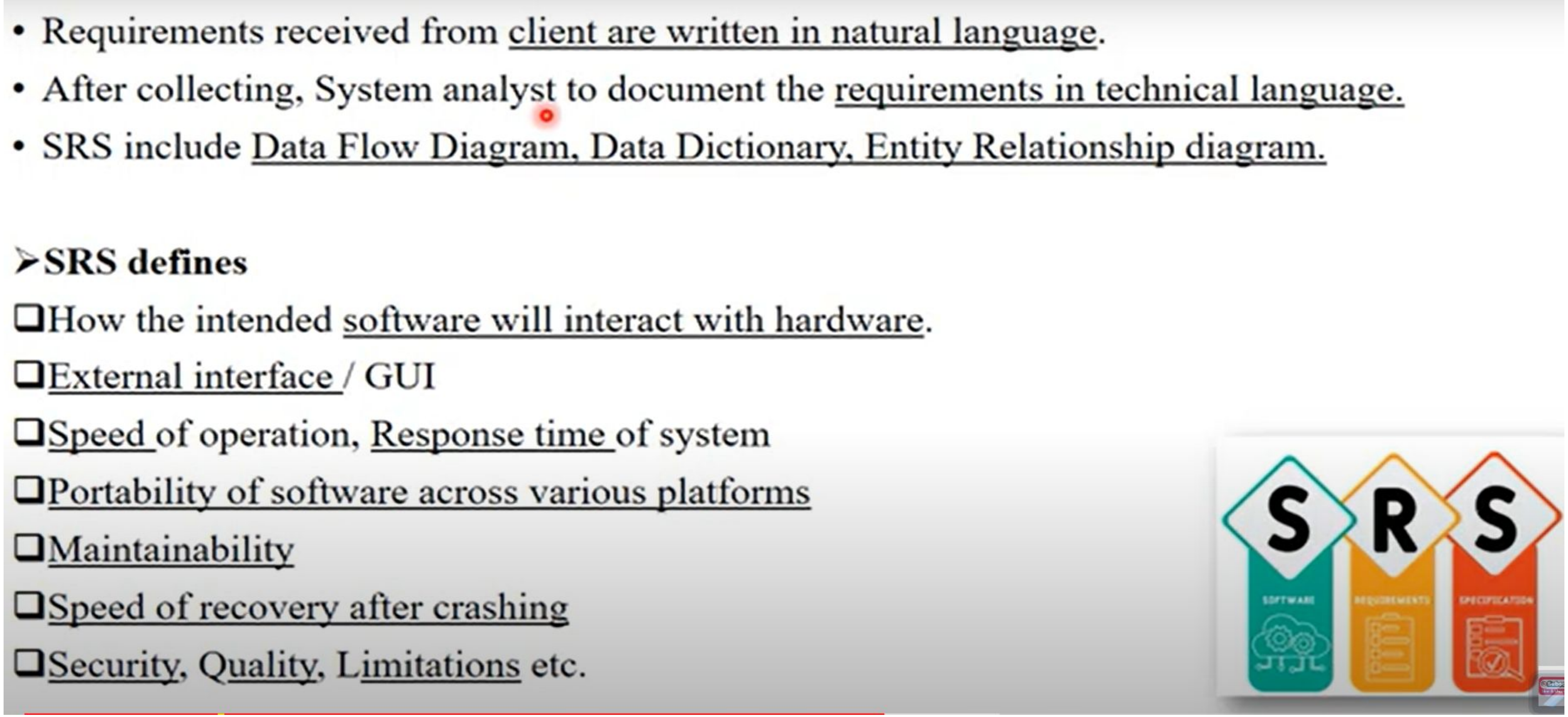


- Requirements elicitation is the process of gathering actual requirements about the needs and expectations of stakeholders for a software system.
- To understand customers, business manuals, the existing software of same type, standards and other stakeholders of the project.

Techniques of Requirements Elicitation:

1. **Interviews:** One-on-one conversations with stakeholders to gather information about their needs and expectations.
2. **Surveys:** These are questionnaires that are distributed to stakeholders to gather information.
3. **Focus Groups:** Small groups of stakeholders who are brought together to discuss their needs.
4. **Observation:** Observing the stakeholders in their work environment to gather information.
5. **Prototyping:** Creating a working model of the software system, which can be used to gather feedback from stakeholders and to validate requirements.

Software Requirement Specification



- Requirements received from client are written in natural language.
- After collecting, System analyst to document the requirements in technical language.
- SRS include Data Flow Diagram, Data Dictionary, Entity Relationship diagram.

➢ **SRS defines**

❑ How the intended software will interact with hardware.

❑ External interface / GUI

❑ Speed of operation, Response time of system

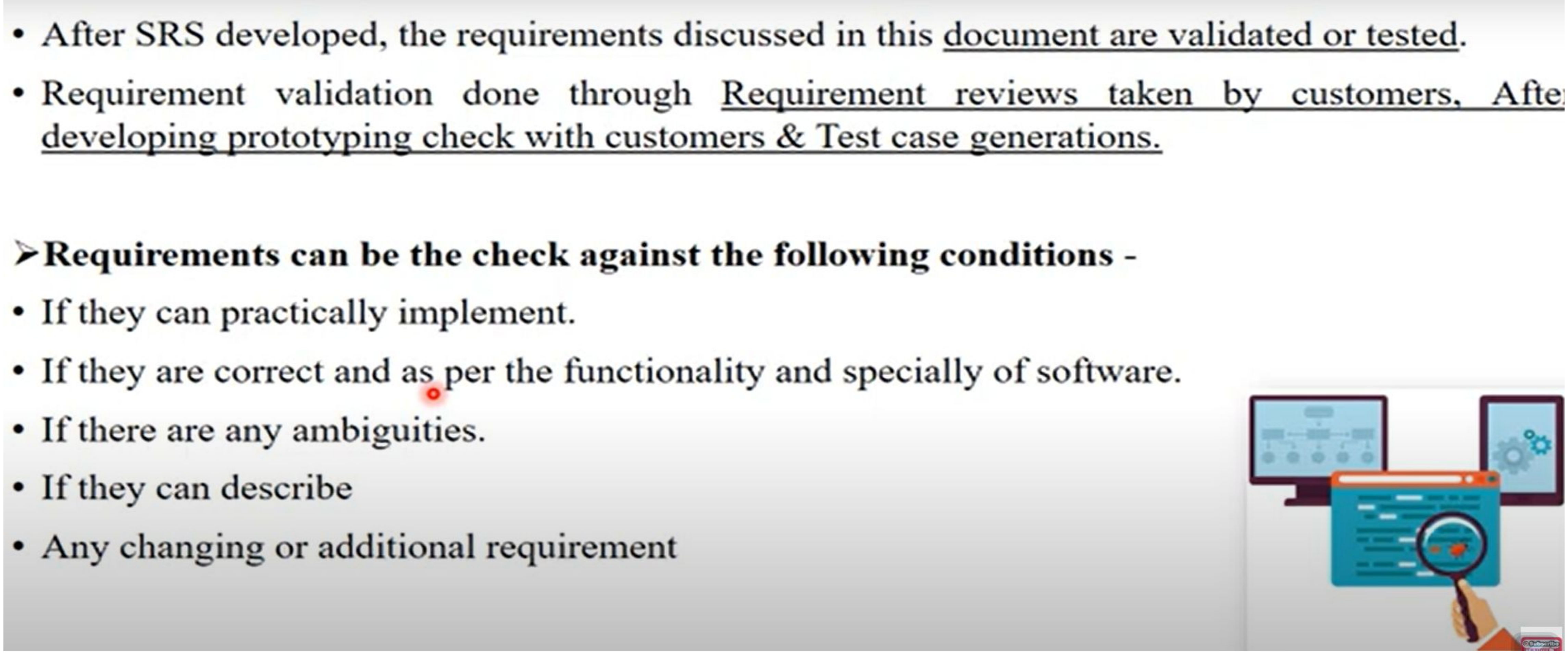
❑ Portability of software across various platforms

❑ Maintainability

❑ Speed of recovery after crashing

❑ Security, Quality, Limitations etc.

Software Requirement Validation

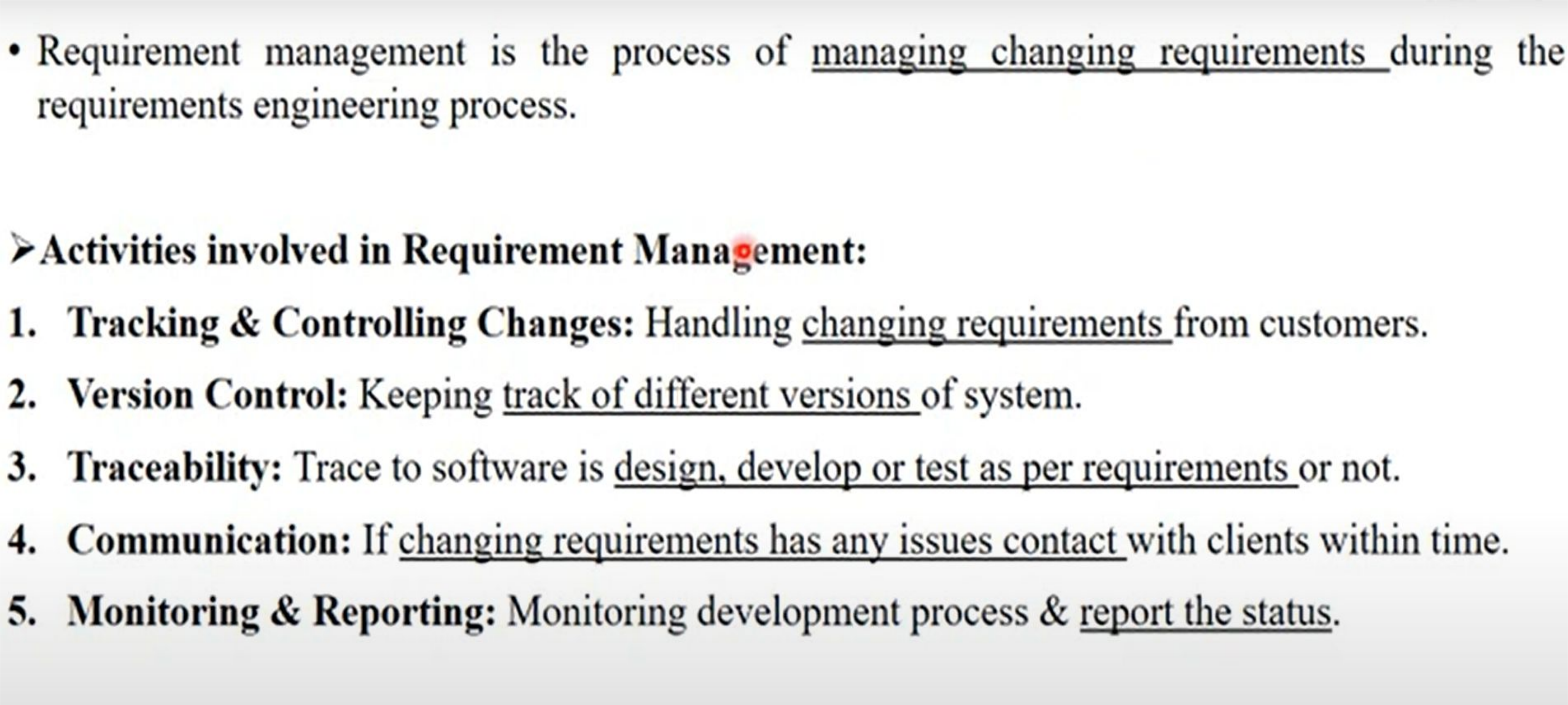


- After SRS developed, the requirements discussed in this document are validated or tested.
- Requirement validation done through Requirement reviews taken by customers, Afte developing prototyping check with customers & Test case generations.

➢**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 can describe
- Any changing or additional requirement

Software Requirement Management



- Requirement management is the process of managing changing requirements during the requirements engineering process.

➢ **Activities involved in Requirement Management:**

1. **Tracking & Controlling Changes:** Handling changing requirements from customers.
2. **Version Control:** Keeping track of different versions of system.
3. **Traceability:** Trace to software is design, develop or test as per requirements or not.
4. **Communication:** If changing requirements has any issues contact with clients within time.
5. **Monitoring & Reporting:** Monitoring development process & report the status.