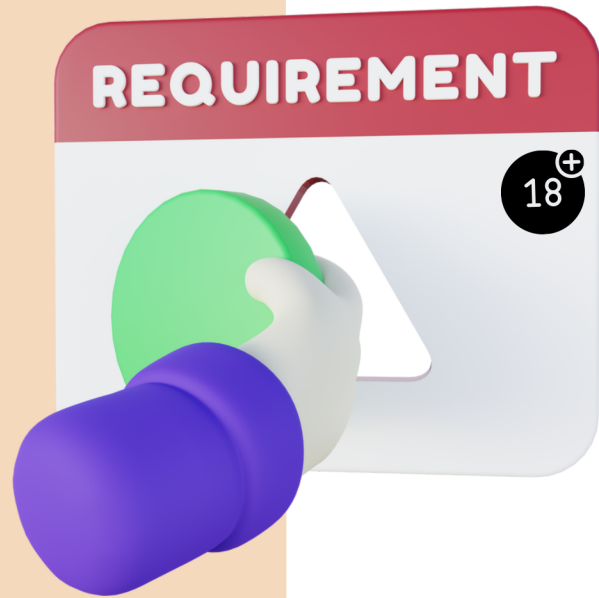




REQUIREMENTS



WHAT TO EXPECT



1

Definition

2

Qualities of a Good Requirement

3

Functional Requirements

4

Non-Functional Requirements

5

Requirement Analysis Techniques

Definition

A software requirement is a formal description of what a software system should do. It represents a specific function, feature, constraint, or quality that the software must possess to meet the needs of its users or stakeholders.

Generally, Requirements in context of QA are classified as -

- Functional Requirement
- Non-Functional Requirement

Qualities of a Good Requirement

Clear &
Unambiguous

Specific

Complete

Testable

Measurable

Consistent

Feasible

Traceable

Prioritized

Stakeholder-
Driven

Dont Stress ! Most of these come naturally

Functional Requirements

Functional requirements outline the specific functions, features, and interactions that the software must possess to meet the needs and expectations of its users or stakeholders.

Example -

As a user of BookStore app, The user should be able to purchase any book listed in the Store.



Non-Functional Requirements

Non-functional requirements (NFRs), also known as quality attributes or system qualities, describe the characteristics and attributes that define how a software system should perform rather than what it should do functionally.

Example -

As a user of BookStore app, The user should be presented with all the Book options in 5 seconds after Login.

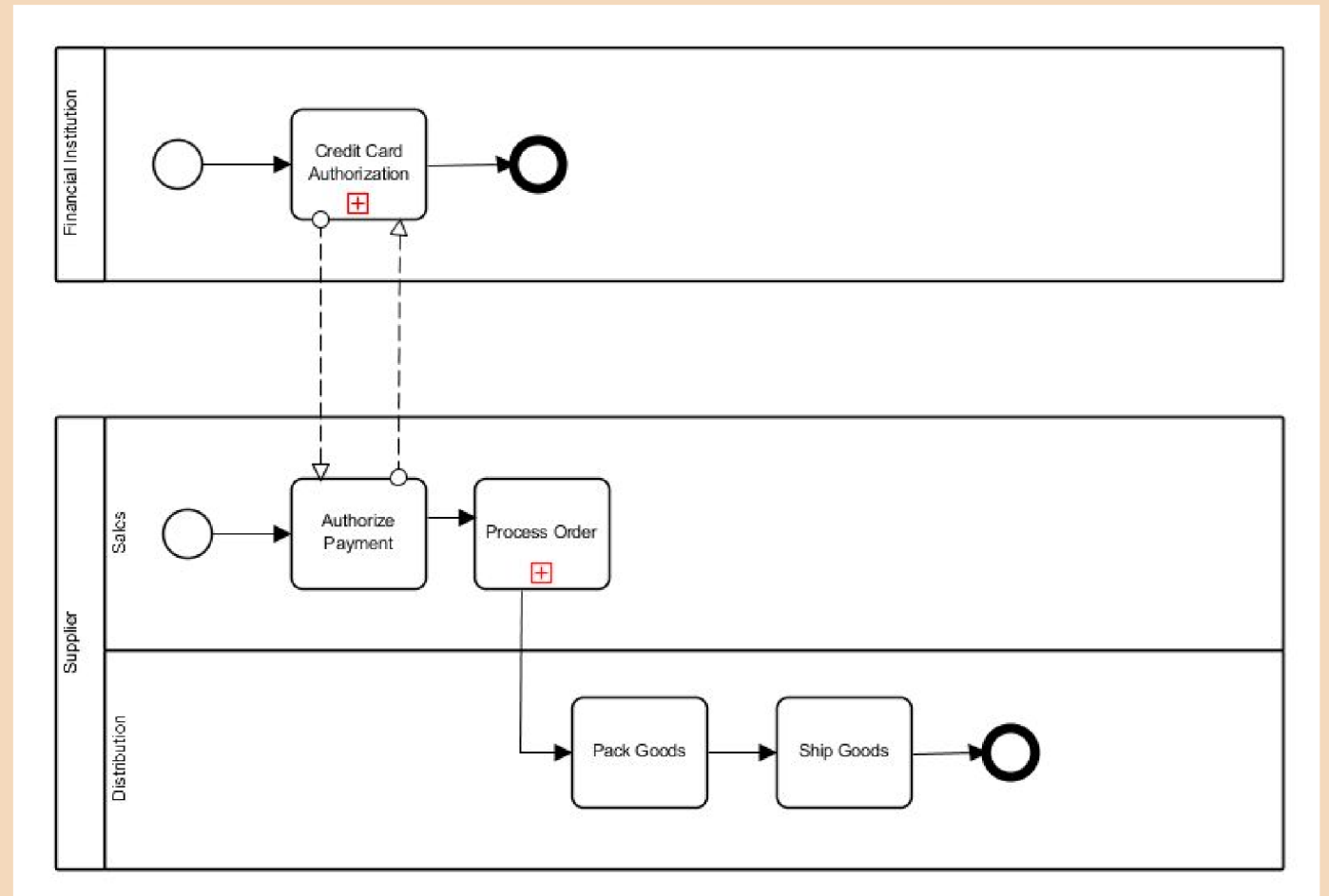


Requirement Analysis Techniques

- **Business Process Model and Notation (BPMN)**

Business Process Model and Notation (BPMN) is a standardized graphical notation used for representing business processes in a visual and easily understandable way. It provides a common language and framework for describing the activities, tasks, and interactions that make up business processes within organizations. BPMN is widely used in business process management (BPM) and workflow automation to improve process efficiency and communication.

<https://www.bpmn.org/>

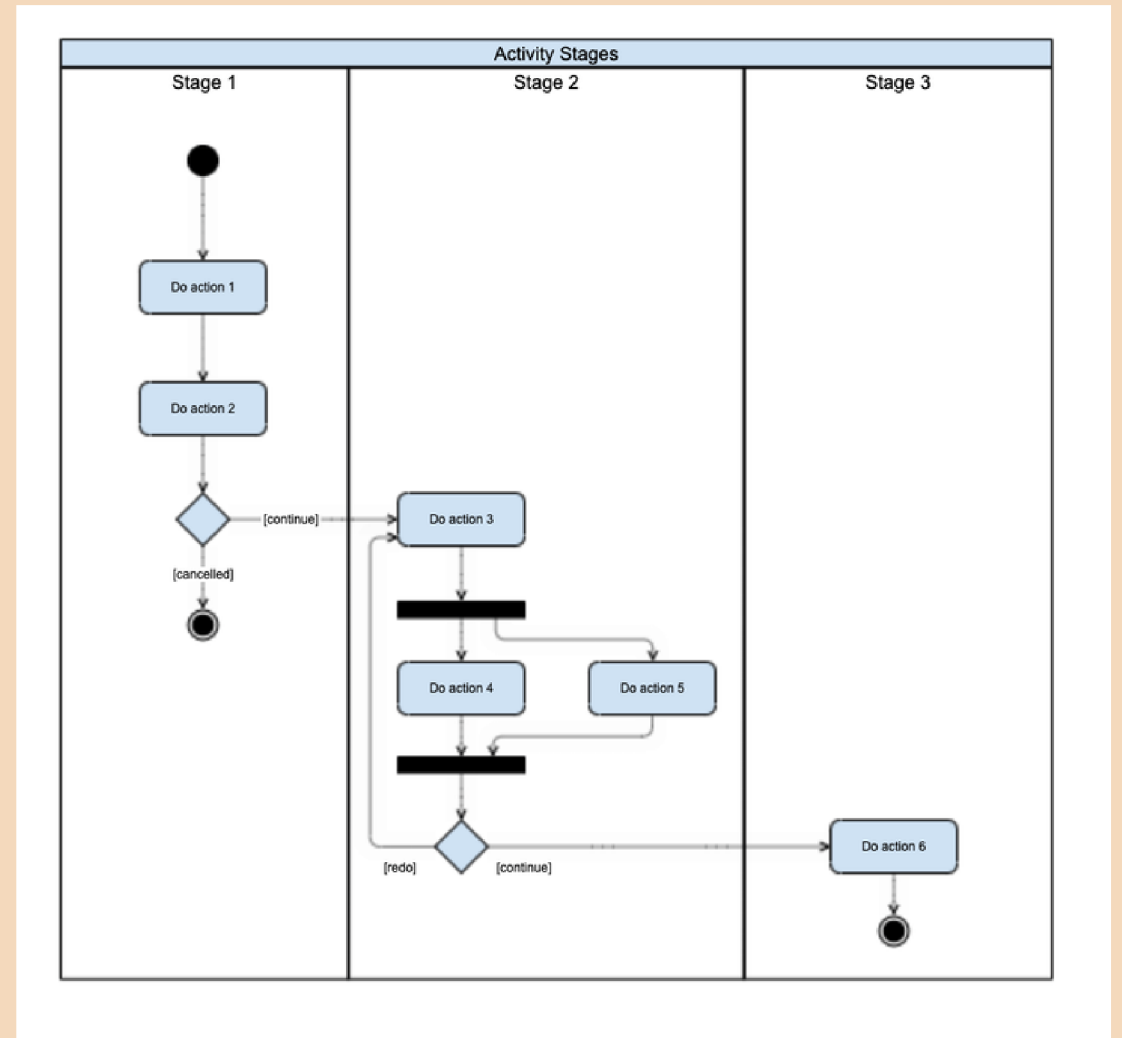


Requirement Analysis Techniques

- UML (Unified Modeling Language)

UML (Unified Modeling Language) is a powerful tool for requirements analysis, design, documentation, and communication within the software development process. It helps stakeholders visualize and understand complex systems, leading to better decision-making, improved collaboration among team members, and ultimately the successful development of software systems.

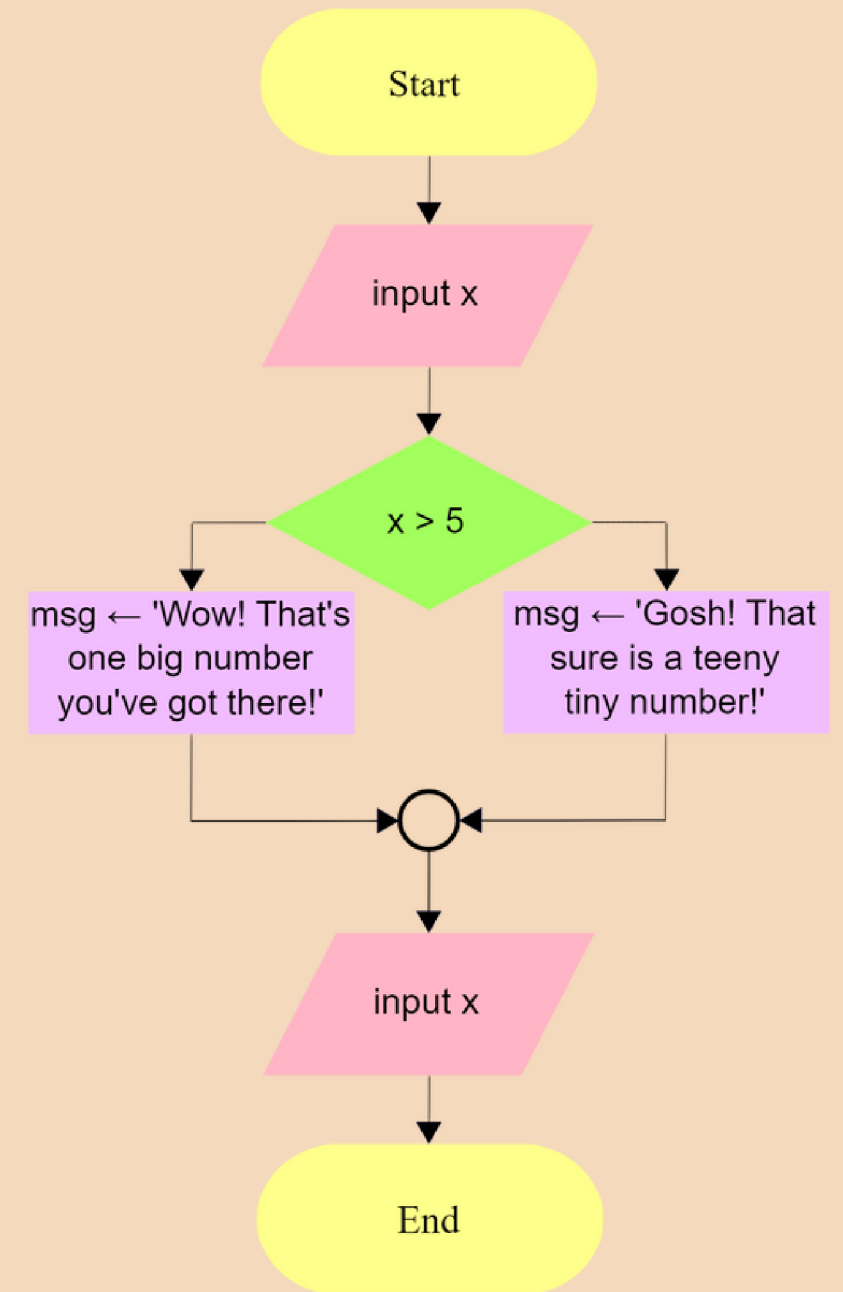
<https://www.uml.org/>



Requirement Analysis Techniques

- **Flowcharts**

Flowcharts depict sequential flow and control logic of a related set of activities. They are useful for both technical and non-technical members.



Requirement Analysis Techniques

Few More -

- **Use Cases:** It helps identify, clarify, and organize requirements by outlining how systems and users interact in a specific context and for a specific purpose.
- Discuss with team and identify testing scope
- Break down requirements in tasks and user stories



**HOT
TIP**

**You can either use 1 or combination
of the different techniques to come
up with Requirements**



66

See you in Next Video

