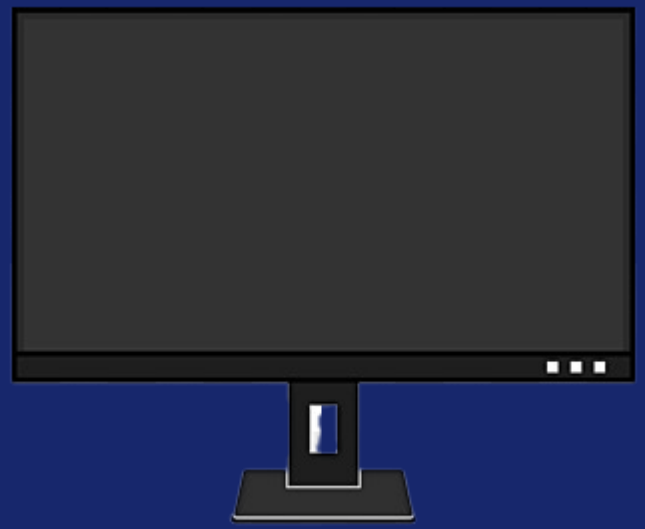
MONA YOUSEF

# QUALITY ASSURANCE MANUAL

MOUNA YOUSEF

# I WANT TO TEST



## Basic testing

1- Turn on the TV
2- Turn off the TV
3- Remote control
4 volume control
5 High level features

**Tester**

**Does it work as expected ?**

**Check everything** (complete testing)

**You need think mind set like user**

MOUNA YOUSEF

# VERIFICATION VS VALIDATION

**Verification:- check list (requirement expectation)**

****TV should do this**.**

**Validation:- test coverage (execution of the code )**

****user is satisfied**.**

## What is software ?

A **program** that performs a **specific task** by **analyzing** the input from the user and then **showing** the **output** to the user.

## What is software testing ?

process of **evaluating** software applications or systems to **ensure** that they meet the **specified requirements** and perform as expected

# VERIFICATION VS VALIDATION

| Verification | Validation |
|---|---|
| The verifying process includes checking documents, design, code, and program | It is a dynamic of testing and validating the actual product |
| It does not involve executing the code | It always involves executing the code |
| Verification uses methods like reviews, walkthroughs, inspections, checking ,etc. | It uses methods like Black Box Testing, White Box Testing, functional and non-functional testing |
| Whether the software conforms to specifications is checked | It checks the requirements and expectations |
| It finds bugs early in the development cycle | It can find bugs that the verification process can not catch |
| Target is an application and software architecture, specification, complete design, database design , etc. | Actual product |
| QA team verifies and ensures that the software matches the requirements in the SRS document. | Testing team validation is executed on software code. |
| It comes before the validation | It comes after the verification |

MOUNA YOUSEF

MOUNA YOUSEF

# SOFTWARE TESTER

- **The goal of the software tester is to find defects or bugs and improve the quality of the software.**

- If there was no tester:

1 Lower quality.

2 The business requirement will be lost.

3- A customer can be lost.

MOUNA YOUSEF

# HOW TO TEST

1- **understand** the application.

2- create a **test plan**.

3- create a **test scenario**

4- create a **test cases**.

5- **finding defects** and **reporting** the software as not working to expected .

MOUNA YOUSEF

# HOW TO TEST

1- Must start **domain knowledge**:

Ex : **e-commerce domain**

1- **P**roduct display page.

2- Search functionality.

3 Cart functionality.

4 Payment method.

https://magento.softwaretestingboard.com/

MOUNA YOUSEF

# DOMAIN

- E-commerce https://www.amazon.com/
- Banking Domain https://www.hdfcbank.com/
- Insurance Domain https://www.hdfcergo.com/
- Health-care Domain https://www.adroitinfosystems.com/products/ehospital-systems
- ERP Domain (Enterprise Resource Planning) https://www.orangehrm.com/
- Education Domain https://www.powerschool.com/classroom/schoology-learning/

And there are several other domains

MOUNA YOUSEF

# FOR UNDERSTANDING SOFTWARE TESTING BETTER

| Manual testing | Automation testing |
| --- | --- |
| A **real person** will be performing the testing **without code**. | the **tool** will be performing the testing .<br><br>**Automation tools** will **replace** the person in this **testing code** or program to **check** software.<br>EX:- Selenium |

MOUNA YOUSEF

# FOR UNDERSTANDING SOFTWARE TESTING BETTER

| Desktop Application Testing | Web Application Testing | Mobile Application Testing |
|---|---|---|
| Not **running** in the **browser**.<br>EX:- eclipse, word, PowerPoint. | **running** in the **browser**.<br>EX:- Amazon, google. | **running** in the **mobile**.<br>EX:- WhatsApp, Facebook. |

# APPLICATION LAYERS

| UI Layers | API Request | Database Layers |
|---|---|---|
| Elements in Page. | business logic layer. | Data stored in the Database (SQL request). |

# FUNCTIONAL VS NON-FUNCTIONAL

| FUNCTIONAL | NON-FUNCTIONAL(VULNERABILITIES ) |
| --- | --- |
| The software testing that validates the software system against the functional requirements/specifications( **Smoke Testing**, **Regression Testing** , **Sanity Testing**, **Re-testing**, **Exploratory Testing**, **Ad hoc Testing** , **Positive Testing** ,**Negative Testing** ,**End to End Testing** ). | The software testing to the handling of non-functional needs (**performance**, **usability**, **reliability**, **security**) of a software application. |

MOUNA YOUSEF

# QUALITY ASSURANCE VS QUALITY CONTROL

Quality assurance: include(**process-oriented activities**)it ensures the **prevention of defects** in the process used to make software applications so the **defects don't arise** when the software application is being developed.

Quality control: include in (**product**-oriented activities) it **executes** the **program** to identify the defects in the software application.

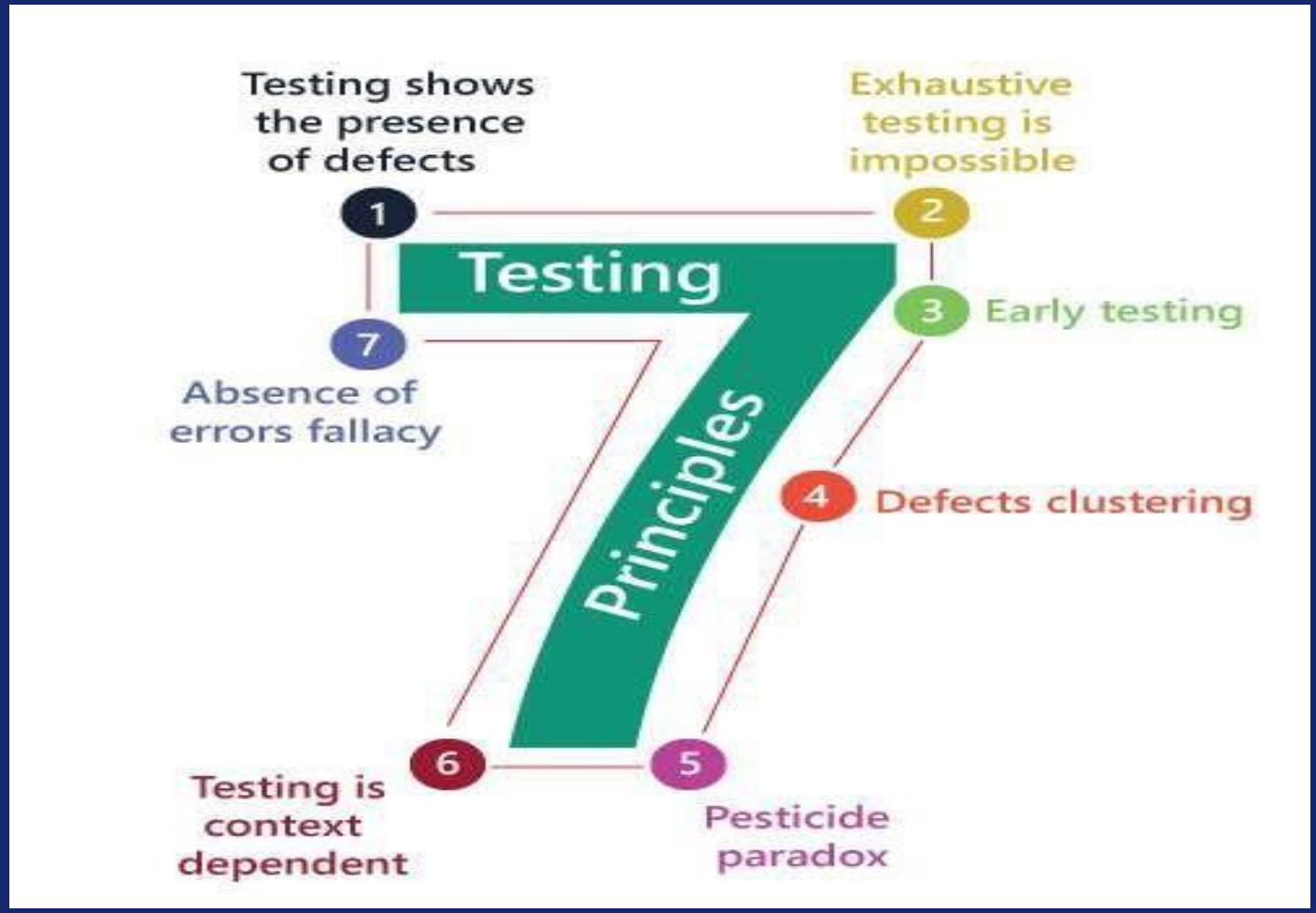
MOUNA YOUSEF

# WHAT EXACTLY DOES A SOFTWARE QA DO?

- **Speed** up the **development** process by identifying bugs at an **early** stage
- **Reduce** the **risk**
- **Maximize** the **value** of the software
- Ensure **successful release** of the product, **save money**, **time**, and the reputation of the company
- Promote **continual improvement**

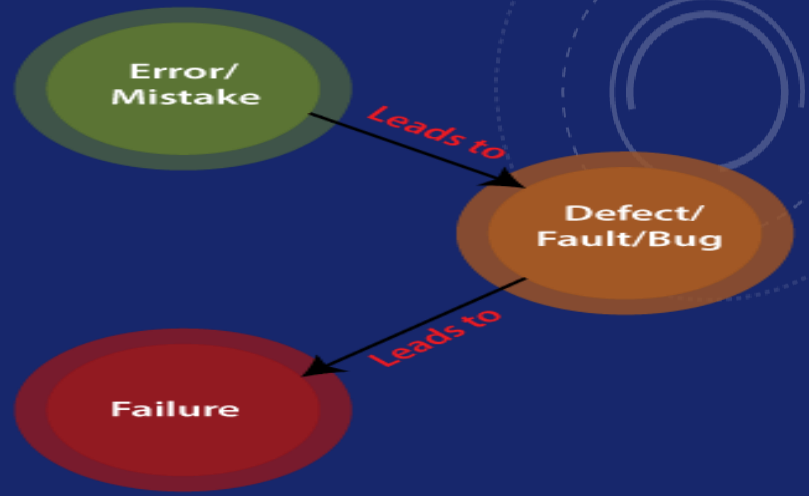
MOUNA YOUSEF

# SEVEN TESTING PRINCIPLES



# ERROR VS DEFECT VS DUG VS FAILURE

- Error: The Problem in code leads to errors.
- Mistake **:** Problem in the document is known as a mistake.
- Defect: When the application is not working as per the requirement is known as a **defect.**
- Issue **:**When the application is not meeting the business requirement.
- Dug: the difference between the actual and expected result of the software.
- Failure : when event a component or system does not perform a requirement, Lots of defect leads to failure of the software..




(This slide has been modified)

MOUNA YOUSEF

# WHAT IS BRS , FRS & SRS ?

- BRS: The BRS document stands for **Business Requirement Specification**. To create the BRS document, the Business analyst will interrelate with the customers.

- **FRS:** The FRS document stands for Functional requirement specification

  is the document which describes all the functions that software or product has to perform.

- SRS: The SRS document stands for **Software Requirement Specification**.

  In this document, the Business Analyst will collect the Customer Requirement Specifications (CRS) from the client and translate them into Software Requirement Specifications (SRS).

MOUNA YOUSEF

# SRS(SOFTWARE REQUIREMENTS SPECIFICATION) DOCUMENTATION

1.Introduction: This section provides an overview of the software project, including its purpose, scope, and target audience.

2.Functional Requirements: including the features and capabilities that the system must provide.

3.Non-functional Requirements: including performance, security, reliability, and usability.

4.System Architecture:an overview of the system architecture, including the hardware and software components that make up the system.

MOUNA YOUSEF

# SRS(SOFTWARE REQUIREMENTS SPECIFICATION) DOCUMENTATION

5.User Interface: describes the user interface of the software system, including the design and layout of the user interface.

6.Use Cases: the use cases of the software system, including the various scenarios in which the system will be used.

7.Data Model: the data model of the software system, including the data structures and relationships between them.

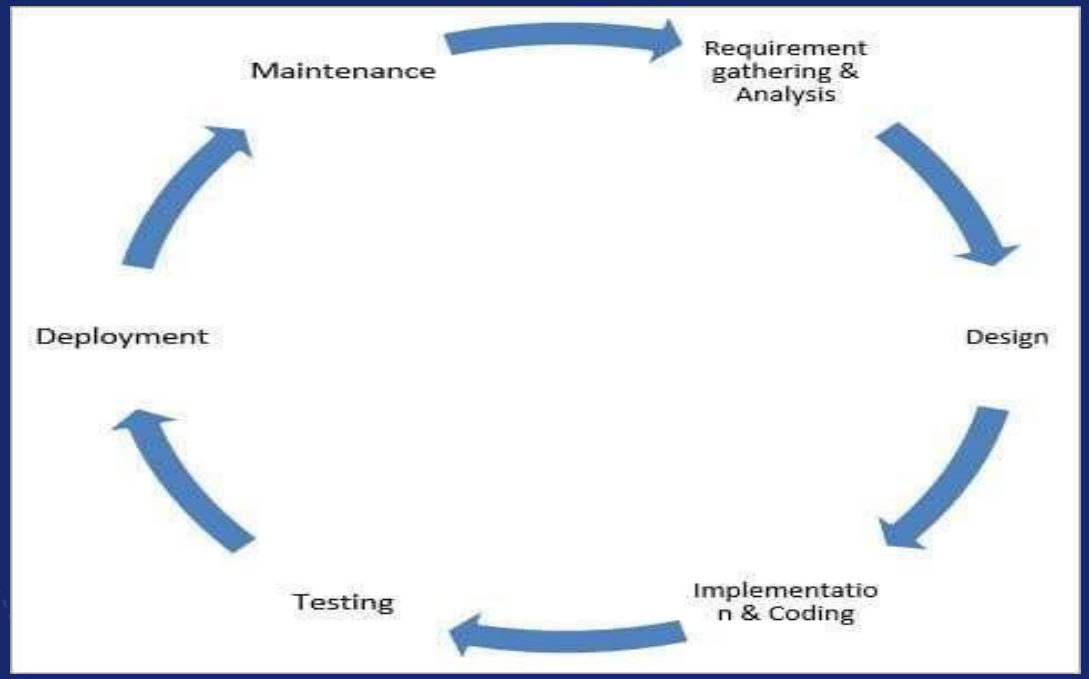
8.Assumptions and Constraints: outlines any assumptions or constraints that are placed on the software project.

9 Glossary: technical terms used in the SRS document.

MOUNA YOUSEF

# SOFTWARE DEVELOPMENT LIFE CYCLE(SDLC)

- **SDLC** : the process followed through Collection requirements, Planning and Requirement Analysis, Designing, Coding, Testing, Deployment, and Maintenance to ensure the product matches the client's requirements.



MOUNA YOUSEF

# SOFTWARE DEVELOPMENT LIFE CYCLE(SDLC)

1- Collection requirements :

- Done by the business analyst.
- BRS Document.
- Meeting with the customer to collect all information for the software.

2 Planning and Requirement Analysis: Analysis SRS Documentation.

3 Designing :

- the designer converts the requirements into a functional(FRS) Document.
- design the UI, the database, and the Architecture.

4- Coding :

- The developers build the application using a programming language (software or application).

MOUNA YOUSEF

# SOFTWARE DEVELOPMENT LIFE CYCLE(SDLC)

5- Testing :

- starts once the coding is complete and the modules are released for testing In this phase, the developed software is tested thoroughly, and identified defects .

6- Deployment :

- Once the product is tested, it is deployed in the production environment or the first UAT (User Acceptance testing) is done depending on the customer's expectation.

7- Maintenance :

- After the deployment of a product on the production environment, maintenance of the product.

MOUNA YOUSEF

# SOFTWARE TESTING LIFE CYCLE(STLC)

**STLC :** the process followed by the tester to ensure the product is of the best quality.

The STLC typically includes the following phases:

1.Requirement Analysis: In this phase, the requirements for the software are analyzed and documented. This phase helps to identify the scope of testing and the test cases that need to be developed.

2.Test Planning: In this phase, a detailed plan is developed for testing the software. The plan includes details such as the testing strategy, test cases, testing tools, and resources required for testing.

3.Test Case design: In this phase, the test cases are designed based on the requirements and the test plan. The test cases are designed to cover all possible scenarios.

MOUNA YOUSEF

# SOFTWARE TESTING LIFE CYCLE(STLC)

4. Test Environment Setup: In this phase, the test environment is set up with the necessary hardware, software, and network configurations required for testing.

5. Test Execution: In this phase, the test cases are executed and the results are recorded. The defects found during testing are reported for fixing.

6. Test Closure: the testing process is formally closed. A final report is prepared, and the testing team provides feedback on the overall testing process. This feedback is used to improve the testing process for future projects.

MOUNA YOUSEF

# WHAT TEST PLAN INCLUDE

1. Introduction: This section describes the purpose and scope of the test plan, and provides an overview of the software being tested.

2. Test strategy: This section outlines the overall testing approach, including the testing methods and techniques that will be used, the test environment and tools, and the roles and responsibilities of the testing team.

3. Test objectives: This section lists the specific objectives, including the quality attributes that will be tested (e.g., functionality, performance, security, usability), the acceptance criteria for each objective, and the risks that will be addressed.

MOUNA YOUSEF

# WHAT TEST PLAN INCLUDE

4. Test schedule: This section provides a timeline for the testing, including the testing phases, and the duration of each phase.

5. Test cases: This section describes the test cases that will be executed, including the input data, expected results, and the conditions under which the test case will be considered a pass or a fail.

6. Test data: This section describes the test data that will be used to execute the test cases, including the source of the data, how it will be prepared, and any special considerations or constraints.

7. Test environment: This section describes the test environment, including the hardware and software configurations, the network topology, and any resources required to execute the tests.

# WHAT TEST PLAN INCLUDE

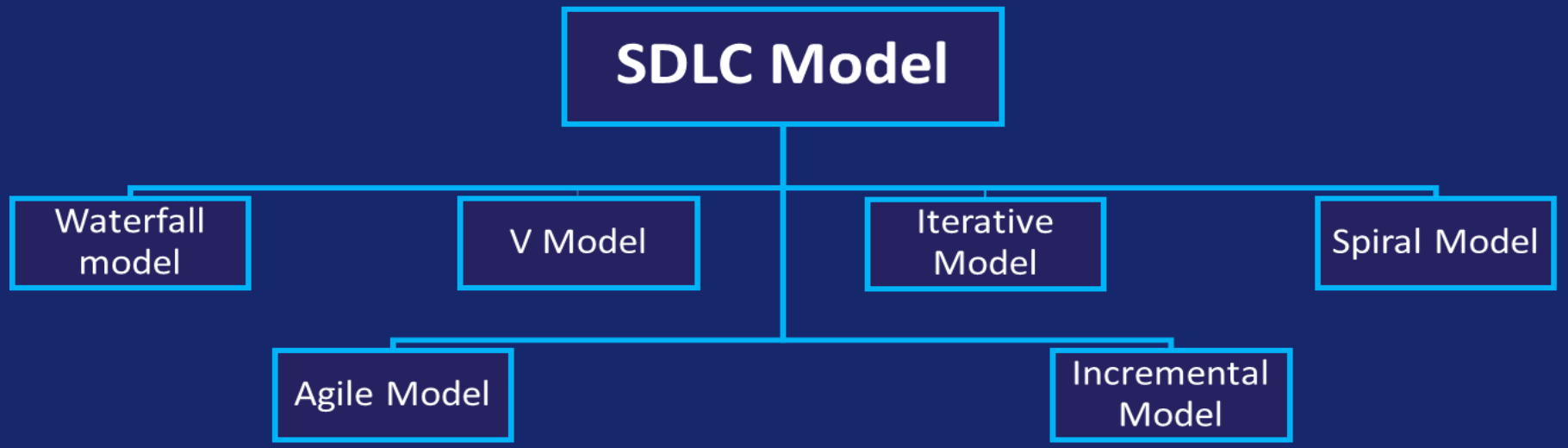
8. Test team: This section describes the roles and responsibilities of the testing team, including the testers, test lead, and any other stakeholders involved in the testing.

9. Test deliverables: This section lists the expected deliverables of the testing, including test reports, defect logs, and any other documentation required to support the testing process.

10. Test risks: This section identifies the risks of the testing, including the likelihood and effect of each risk, and outlines the risk reduction strategies that will be used to address them.

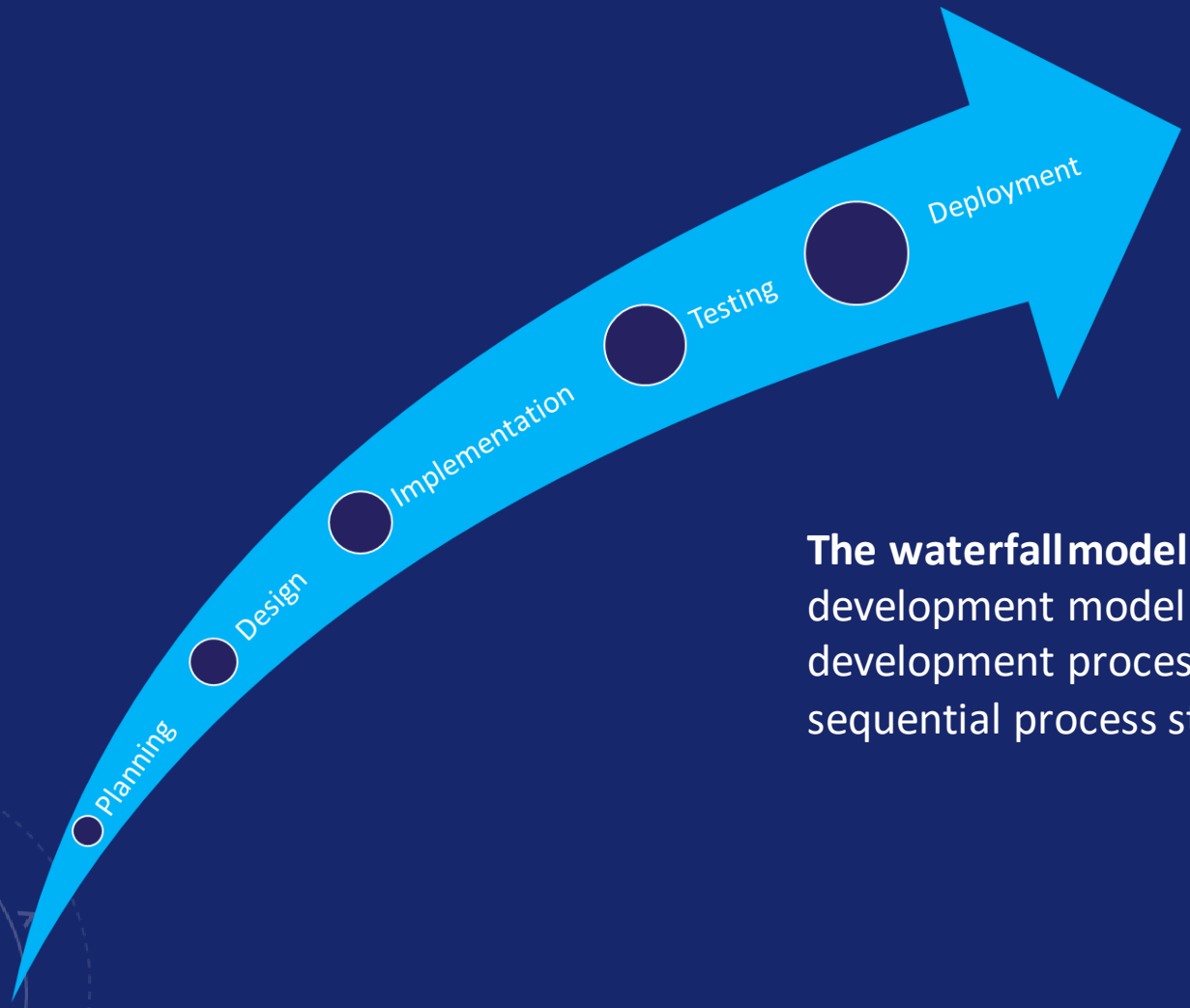
MOUNA YOUSEF

# SOFTWARE DEVELOPMENT MODELS



MOUNA YOUSEF

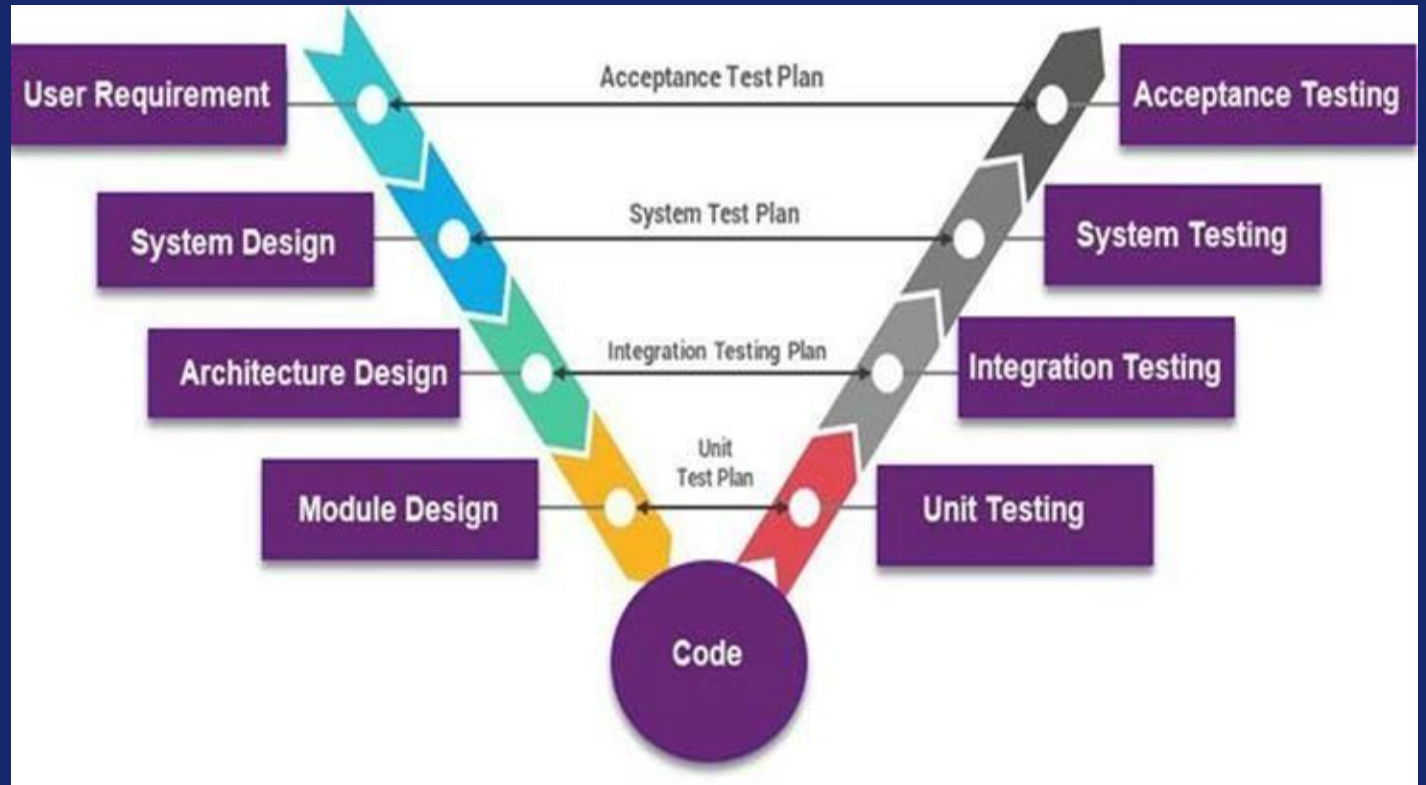
# WATERFALL MODEL




**The waterfall model :** a Sequential development model that presents the software development process as linear, sequential process steps.

# V-MODEL

V-model: This model is similar to the waterfall model, but with testing. It involves developing test plans and test cases before the implementation phase, with testing occurring at each stage of development.



# SPIRAL MODEL

Spiral model: This model combines elements of the waterfall model and the iterative model, with a focus on risk management. It includes repeated cycles of planning, risk analysis, development, and testing.

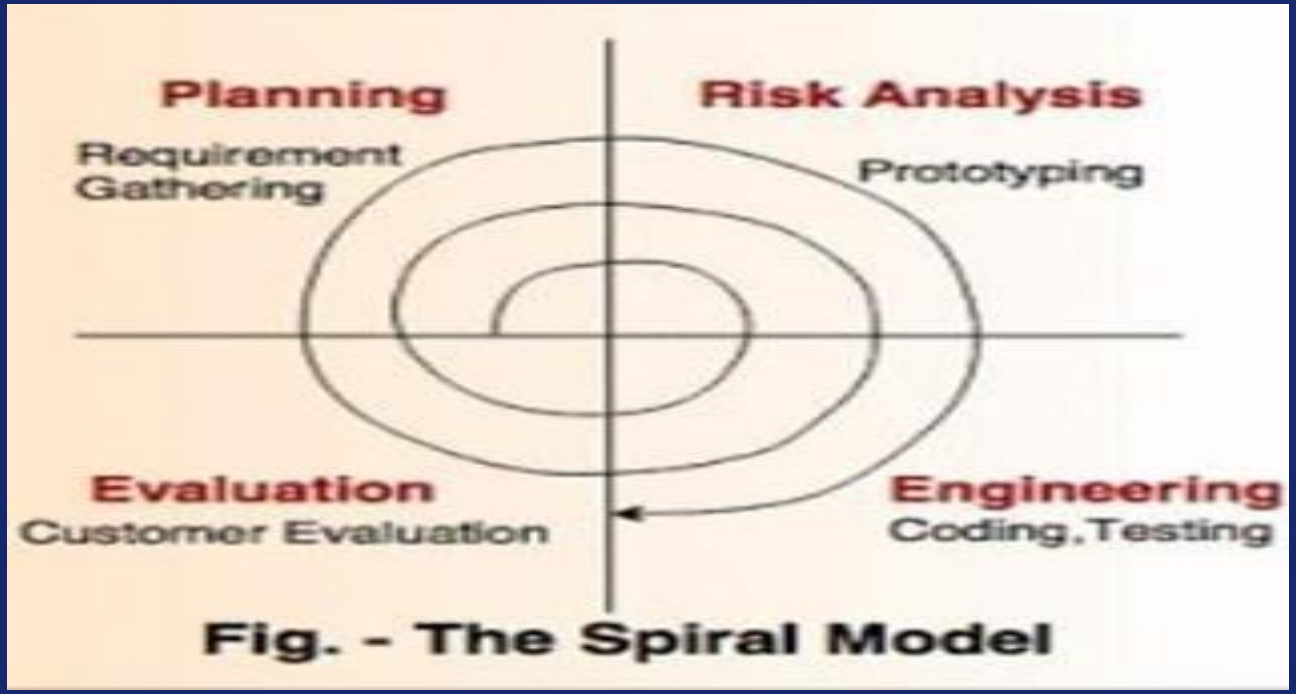
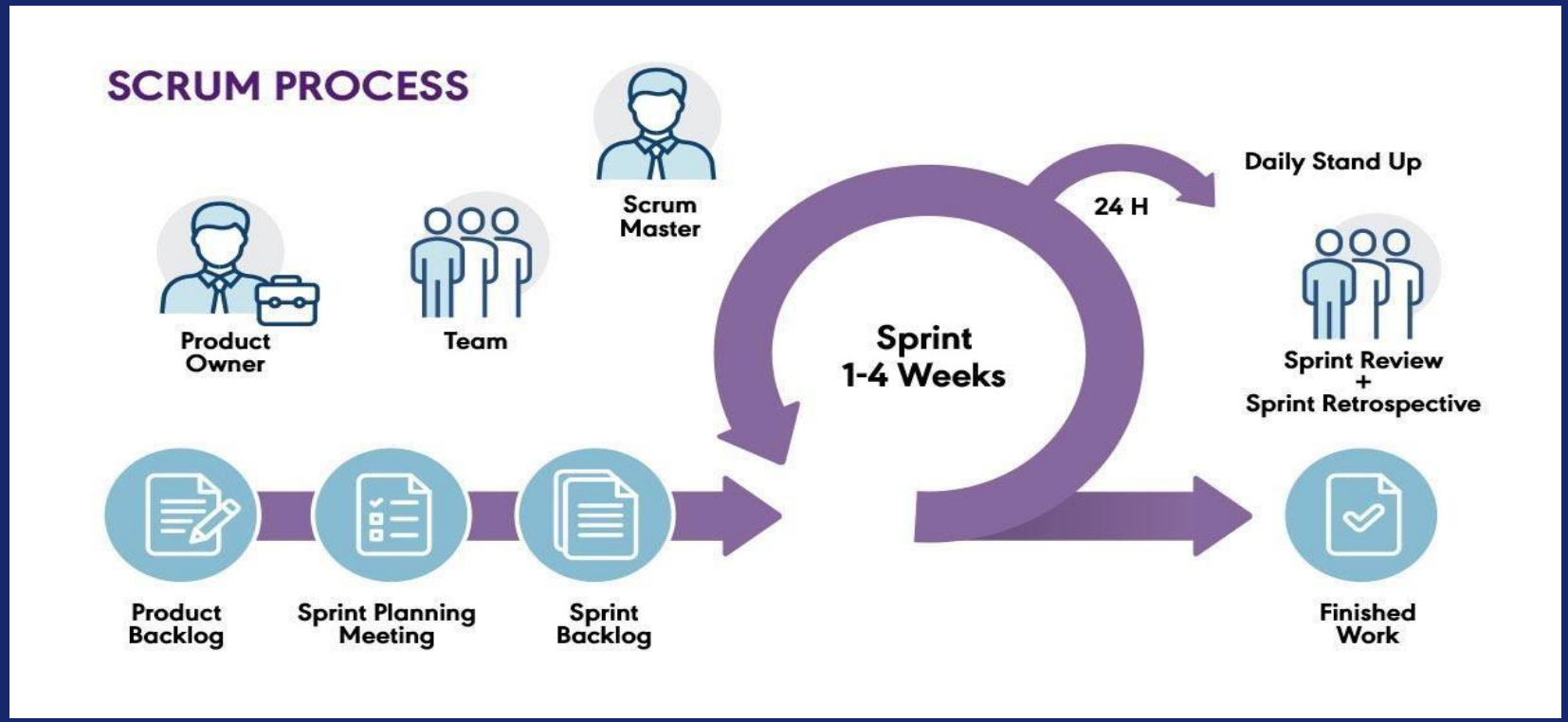



Fig. - The Spiral Model

MOUNA YOUSEF

# *AGILE MODEL/* SCRUM

**Agile Model: This** refers to a software development approach based on iterative development. Agile methods break tasks into smaller iterations, or parts do not directly involve long-term planning



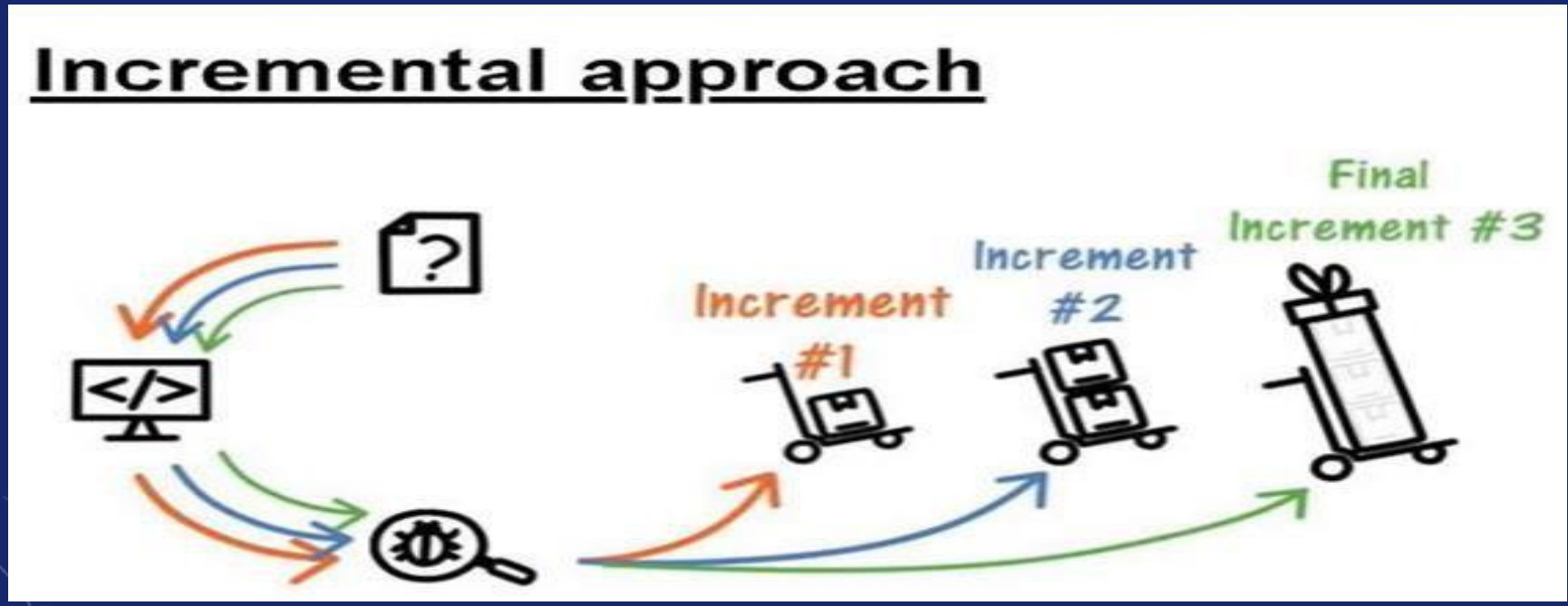
MOUNA YOUSEF

# *AGILE MODEL/* KANBAN

**Kanban :** The Kanban Method is a means to design, manage, and improve flow systems for knowledge work.

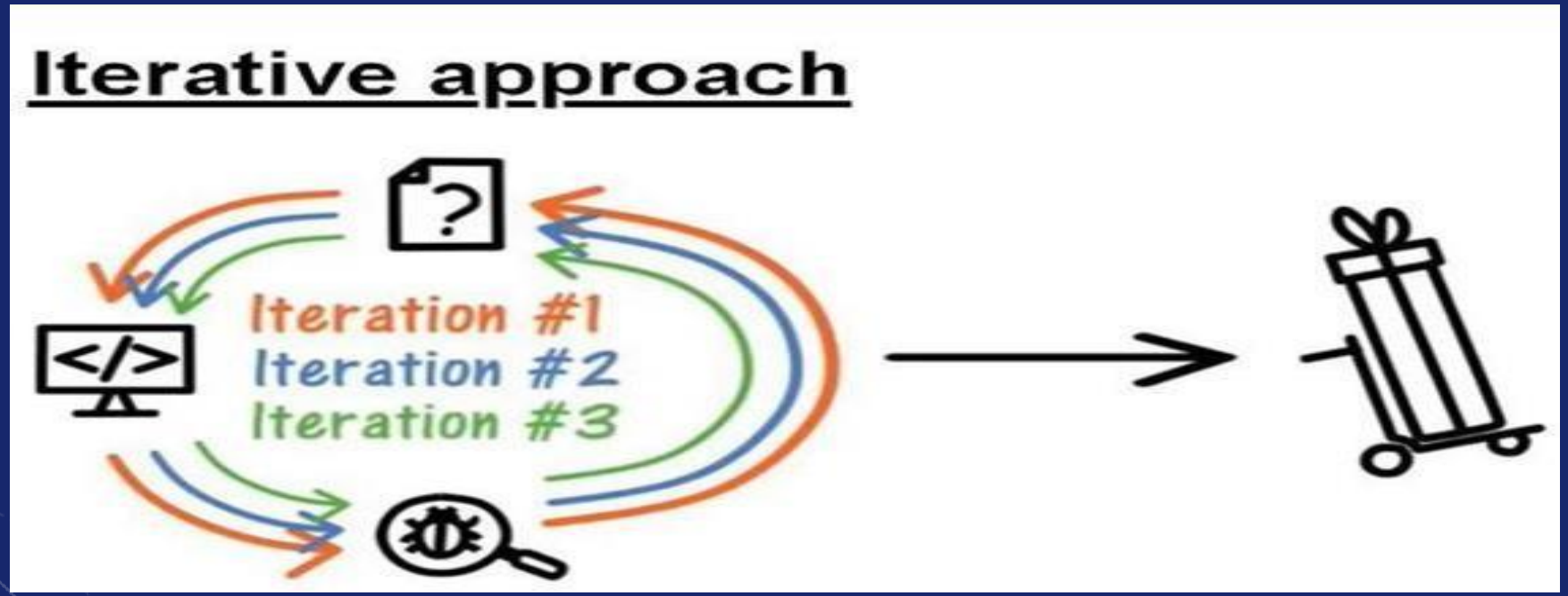
# INCREMENTAL MODEL

Incremental model: the development process is broken down into smaller parts called increments, and each increment delivers a working version of the software.



# ITERATIVE MODEL

Iteration is a time box during which development takes place.



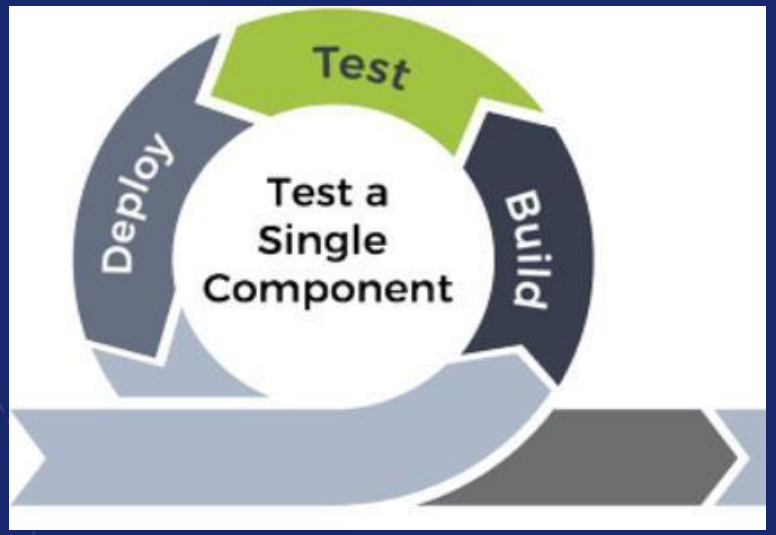
# Test Levels

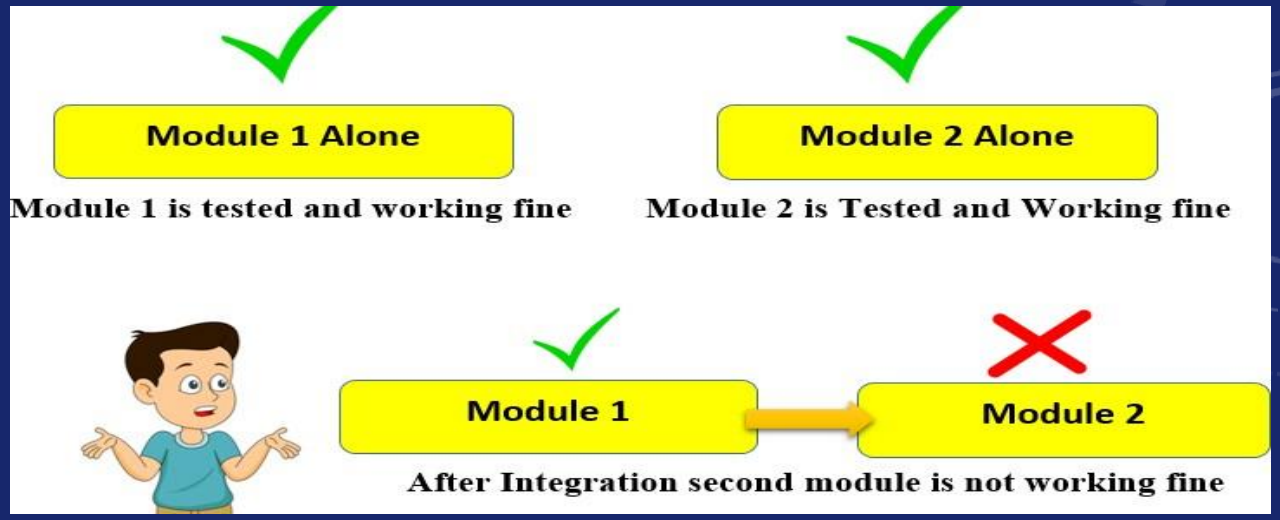
**1. Component Testing (Unit or Module Testing)**

- Developers usually execute this testing, and it is done to ensure that each unit or module of the software works as expected

**2. Integration Testing**

- **The testing performed to find defects in the interfaces between components, Interacting with different parts of systems, and in the Interfaces between systems.**



MOUNA YOUSEF

# Integration testing technique used

1. Big Bang Integration Testing: This technique involves integrating all the software components at once and testing the entire system as a whole.

2. Top-Down Integration Testing: This technique starts with testing the highest-level modules or components and gradually integrating lower-level modules or components.

3. Bottom-Up Integration Testing: This technique starts with testing the lower-level modules or components and gradually integrating higher-level modules or components. It is useful for identifying defects in the lower-level modules, but it can be difficult to test the system as a whole.