



FUNCTIONAL TESTING EXPLAINED

WHAT TO EXPECT



1

Functional Testing Overview

2

Objectives of Functional Testing

3

Types of Functional Testing

A

Unit Testing

B

Integration Testing

C

System Testing

D

Regression Testing

E

Acceptance Testing

4

Best Practices for Functional Testing

4

Challenges in Functional Testing

Functional Testing

Functional testing is a type of software testing that focuses on verifying the functional requirements and behavior of a software application.

It involves testing the system's features, functions, and interactions to ensure that they work as intended and meet the specified requirements. The primary goal of functional testing is to validate that the software performs its intended tasks accurately and reliably, without any errors or unexpected behavior.

Functional Testing is crucial for any product success !

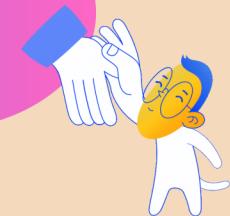
Ensuring Correct Functionality



Compliance and Regulations



Meeting User Expectations



Quality Assurance and Bug Detection



Iterative Development and Continuous Improvement

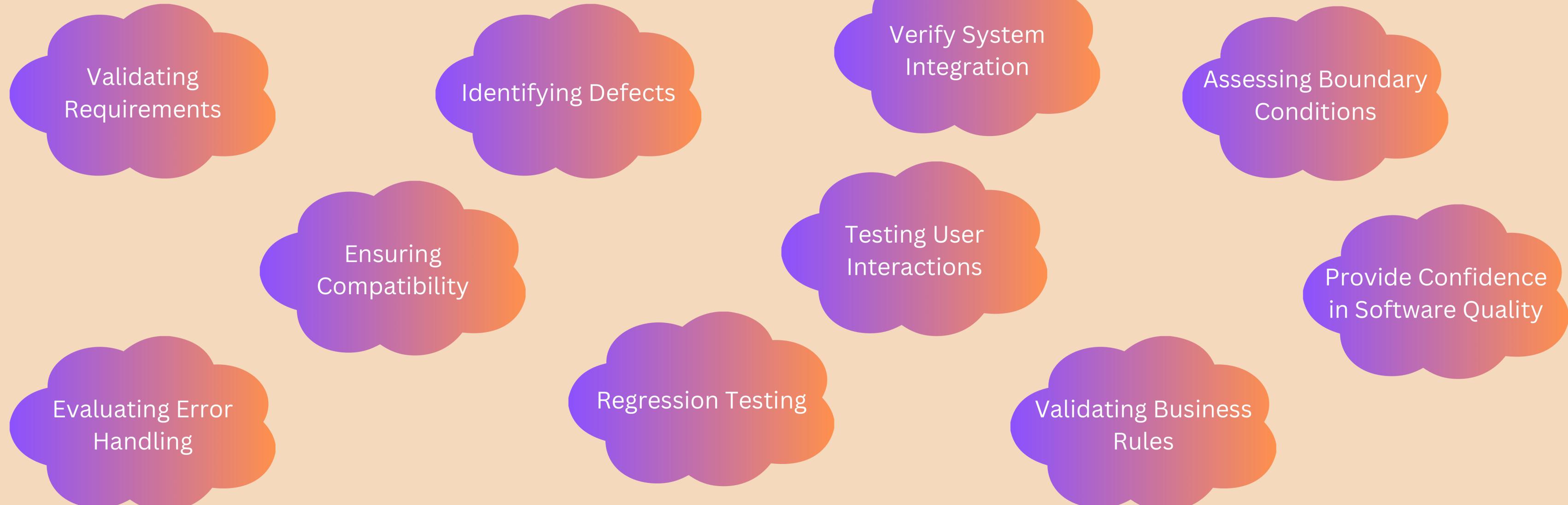
NEW DEVELOPMENTS



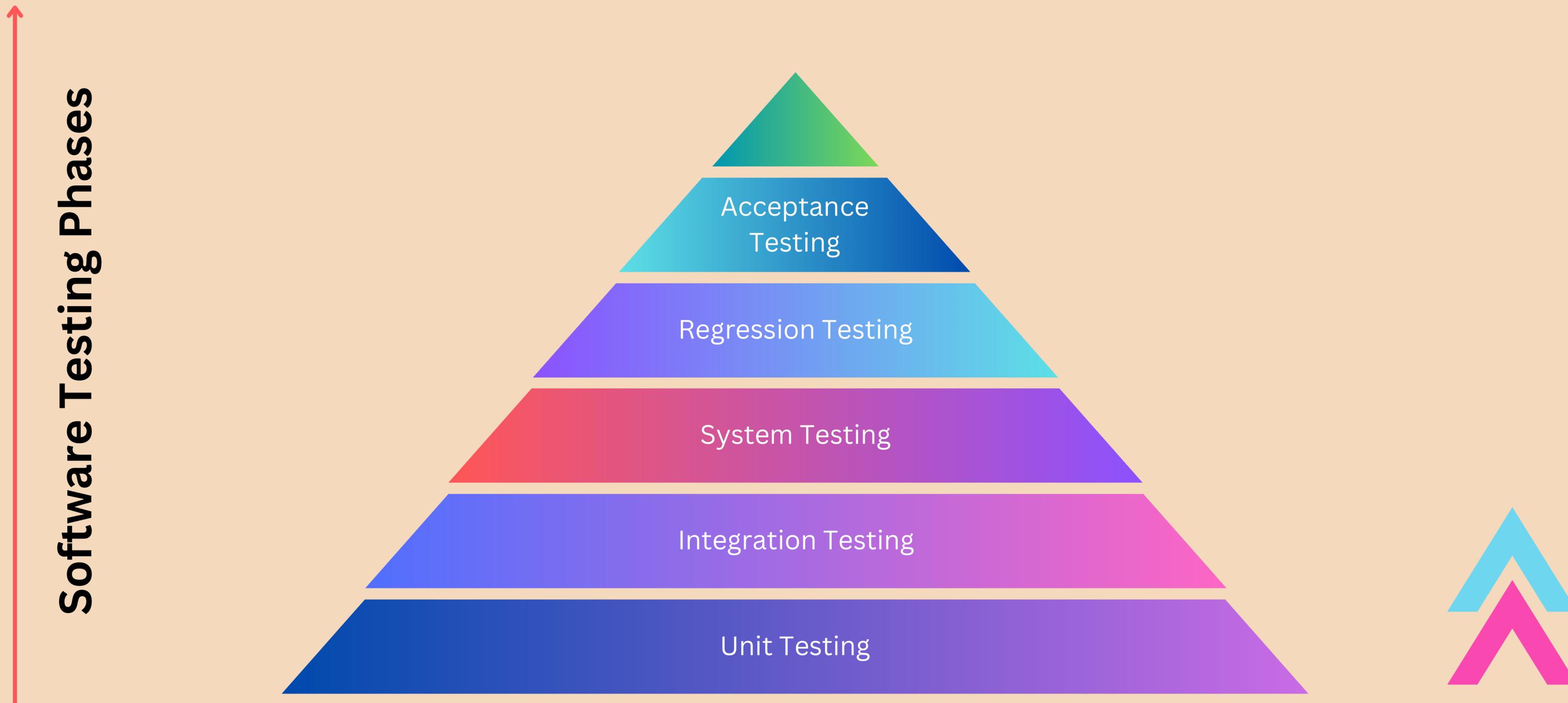
Objectives of Functional Testing

The objectives of functional testing can vary depending on the specific project and requirements. However, some common objectives of functional testing include:

These objectives collectively aim to ensure that the software application functions correctly, meets user needs, and delivers the desired outcomes. Functional testing helps identify and address issues early in the development process, leading to improved software quality and user satisfaction.



Testing Pyramid



Unit Testing

What ?

- Unit testing is a software testing technique that focuses on testing individual units of code in isolation to ensure their correctness and reliability.
- A unit refers to the smallest testable part of an application, usually a function, method, or class.

How ?

- Unit tests are designed to run independently of the rest of the application.
- Typically run by developers
- Unit tests are usually automated, meaning they can be executed repeatedly without manual intervention.
- Validations are usually called "assertions"

Benefits

- Unit tests catch issues early in the development cycle, making it easier and cheaper to fix them.
- Unit tests serve as a form of documentation by demonstrating how units of code are intended to be used and the expected behavior.
- By confirming the accuracy of individual units of code, it ensures code blocks work as intended

Integration Testing

What ?

- Integration testing is a type of software testing that checks how different components or modules of an application work together to ensure they function correctly as a unified system.
- Focuses on testing the interactions and interdependencies between different components or modules of an application.

How ?

- Integration testing focuses on testing the interfaces or connections between different modules or components.
- During integration testing, the behavior and functionality of multiple units are tested together to ensure they work correctly as a cohesive system.

Benefits

- Integration testing helps uncover issues related to the interactions between different modules, such as data inconsistencies, communication failures, or compatibility problems.
- By testing the integrated system, integration testing ensures that the application functions as intended and meets the specified requirements.

System Testing

What ?

- System testing is a comprehensive software testing process that evaluates the behavior and functionality of an entire software system as a whole.
- Performed after integration testing and focuses on verifying that the system meets the specified requirements and functions as intended in a real-world or environment.

How ?

- System testing is conducted in an environment that closely resembles the production environment to ensure realistic testing conditions.
- System testing aims to achieve high test coverage, ensuring that all system features, use cases, and user workflows are tested thoroughly.

Benefits

- System testing helps identify defects or inconsistencies that may arise due to the interactions between different components or modules.
- System testing verifies that the software system meets the specified requirements and behaves as expected from a user's perspective.

Regression Testing

What ?

- Regression testing is a software testing technique that focuses on retesting previously tested functionalities or components after making modifications or enhancements to the software.
- The purpose of regression testing is to ensure that any changes made to the system do not introduce new defects or cause unintended side effects that may impact the existing functionality.

How ?

- Regression testing involves selecting a subset of existing test cases from a test suite that are relevant to the areas impacted by the changes.
- Regression testing is often automated to ensure efficiency and consistency.
- Test cases within the regression test suite may be prioritized based on their criticality or the areas of the system that have been affected by the changes. High-priority test cases are executed first to quickly identify any major issues.

Benefits

- Regression testing helps ensure that modifications or enhancements do not unintentionally introduce new defects or disrupt the stability and reliability of the software system.
- Regression testing contributes to maintaining a high level of software quality

Acceptance Testing

What ?

- Usually performed by a group of End Users
- Acceptance testing, also known as user acceptance testing (UAT), is a software testing phase that focuses on determining whether a system meets the requirements and expectations of the end-users or stakeholders.

How ?

- Acceptance testing is performed from the perspective of end-users or stakeholders who will be utilizing the software in real-world scenarios.
- Acceptance testing can be done in alpha or beta phases. Alpha testing involves limited users within the development team, testing in a controlled environment. Beta testing involves a larger group of external users providing feedback based on real-world usage.

Benefits

- Acceptance testing ensures that the software meets the requirements and expectations of the end-users, resulting in higher user satisfaction.
- By involving end-users in the testing process, acceptance testing helps uncover issues, usability problems, or missing features early, allowing for timely resolution.

Best Practices for Functional Testing

- Test early and often
- Use automated testing tools
- Test in realistic environments
- Test for both positive and negative scenarios
- Collaborate with stakeholders

Challenges in Functional Testing

- Limited resources and budget
- Lack of skilled testers
- Complex and constantly changing requirements
- Integration issues
- Time constraints

“
See you in Next Video

