**Software Testing**

Contents

[What is Software Testing? 1](#_Toc152758695)

[Types Of Software Testing: 1](#_Toc152758696)

[Functional Testing: 1](#_Toc152758697)

[What is Functional Testing? Give 3 examples. 1](#_Toc152758698)

[Non-Functional Testing: 4](#_Toc152758699)

[What is Non-Functional Testing? Give 3 examples. 4](#_Toc152758700)

# What is Software Testing?

Testing involves systematically and methodically evaluating a product, process, or system to determine its reliability, performance, and functionality. Testing has a long history and is an essential part of the software, where basic tests are used to assess the quality of goods and services. And software development has a vital role in preventing any disaster, for example, in a customer account. Testing also improves the software quality and can speed up the production process, removing bugs from any program before the software is used.

Testing aims to identify defects or weaknesses in a product or system before it is released to the market or deployed in the field. Testing also plays a critical role in compliance and regulatory requirements, ensuring that products and systems meet specific standards and regulations.

# Types Of Software Testing:

## Functional Testing:

### What is Functional Testing? Give 3 examples.

The most common type of testing is functional or black-box testing. It's commonly done by testers who don't have access to the source code and rely on what they can see from the outside, such as UI/UX components. This type of testing requires knowledge about how software works and an understanding of its features. A tester will use these skills to perform different scenarios that may cause problems with the application.

Functional testing can be divided into different categories, such as acceptance and regression testing.

Acceptance testing is performed by business stakeholders to ensure that the application meets their requirements before release.

Regression testing, on the other hand, is meant to catch any issues that were previously fixed but weren't properly taken care of or weren't noticed.

Functional Testing types:

* Unit testing
* Smoke testing
* User Acceptance
* Integration Testing
* Regression testing
* Localization
* Globalization
* Interoperability

#### Examples of Functional testing:

• Integrated Testing

Integrated/Integration Testing is a second level of software testing techniques. This kind of examination involves verifying the parts of the software or running the software to show the errors if they are. Also, the test is running together with the design level.

Many software developers usually build the application software, so this integration testing runs to check the integration between modules.

Advantages of the integrated testing:

- Every module is running correctly.

- Involve shows the interface issues.

- The test can be implemented between two modules, not the entire application.

- Easy to detect any problems with the tester.

- The tester can analyse the entire system more quickly and more securely.

• System Level Testing

System Level Testing includes testing the entire project in its natural environment. System testing includes a series of minor tests and testing software and hardware.

There is a specific order of system testing:

- Unit testing checks every module or an entire block of the software code. The software developer who wrote the code usually performs the unit test.

- Integration testing is performed at every beginning and end of application software integration. Every module must be tested because every module can be created separately, so the testing stage is here essential.

- The system is tested by a professional tester using one unique testing tool.

- Acceptance testing is beta testing which the typical user carries out.

• Acceptance Testing

Acceptance Testing is performed after the testers and the project team implement and approve the application to the system. The test is carried out by customers and/or a few users to check the actual application.

There are various possibilities for this testing:

- User acceptance testing. Potential users are testing the software to see if all the features work correctly, such as logging in and registering.

- Business acceptance testing. This test is to judge if the product meets the customers' specifications.

- Contract acceptance testing. This test checks the documentation to see if this product met all the contract points and should give a use case expectation.

- Regulations/compliance acceptance testing. The tester checks all the regulations to see if they are implemented correctly.

- Operational acceptance testing. This non – functional testing and test includes recovery, compatibility, maintainability, technical support availability localisation, and many others.

- Alpha testing.

- Beta testing.

• User Interface Testing

UI testing is the GUI testing, including any user contact with this application software. The visualisation of the application is tested to ensure that the user performs with the software at its top.

User interface testing includes:

- Functionality. User experience with entering their details or researching a product.

- Visual design. No confusion to the user about colours or fonts.

- Responsiveness. Usually, the time to respond to the task.

- Performance. A general filing from using the application.

- Usability.

- Accessibility. Special tools for users to help them use the app.

- Compliance.

Actual tests for the user interface:

- Data type error test.

- The text holder's test.

- Navigation elements test.

- Progress bars test.

- Menu and drop-down menu test.

- Table scrolling test.

- Error logging test.

- Working shortcuts test.

## Non-Functional Testing:

### What is Non-Functional Testing? Give 3 examples.

It includes the system's function; its importance is much less than functional.

The testing check usually functions outside the app in the environment where the program is implemented.

#### Examples of the non – functional testing:

• Performance Testing

- Load Testing

User is testing the app in their typical environment.

- Stress Testing

Testing the app in a highly challenging environment.

- Endurance Testing

Test the time how long the app can run at a typical load.

- Spike Testing

The test is running when there is an unexpected user load.

• UI Testing

The testing includes the interface design.

• Security Testing

The security testing is running to prevent unauthorized access.

• Configuration testing

Check the behaviour of the application by applying various options in the program.

• Compatibility Testing

It is the software testing using this application on various operating systems platforms.

• Localization Testing

Testing the local settings on the different regions.

• Usability Testing

This kind of testing checks if the application is user–friendly and accessible.