UNIT 4

SOFTWARE TESTING

FUNCTIONAL TESTING & STRUCTURAL TESTING

FUNCTIONAL TESTING

Functional Testing is a type of Software Testing in which the system is tested against the functional requirements and specifications. Functional testing ensures that the requirements or specifications are properly satisfied by the application. This type of testing is particularly concerned with the result of processing. It focuses on the simulation of actual system usage but does not develop any system structure assumptions.

What is Functional Testing?

Functional testing is basically defined as a type of testing that verifies that each function of the software application works in conformance with the requirement and specification. This testing is not concerned with the source code of the application. Each functionality of the software application is tested by providing appropriate test input, expecting the output, and comparing the actual output with the expected output. This testing focuses on checking the user interface, APIs, database, security, client or server application, and functionality of the Application Under Test. Functional testing can be manual or automated.

Purpose of Functional Testing

Functional testing mainly involves black box testing and can be done manually or using automation. The purpose of functional testing is to:

- **Test each function of the application:** Functional testing tests each function of the application by providing the appropriate input and verifying the output against the functional requirements of the application.
- **Test primary entry function:** In functional testing, the tester tests each entry function of the application to check all the entry and exit points.
- **Test flow of the GUI screen:** In functional testing, the flow of the GUI screen is checked so that the user can navigate throughout the application.

What to Test in Functional Testing?

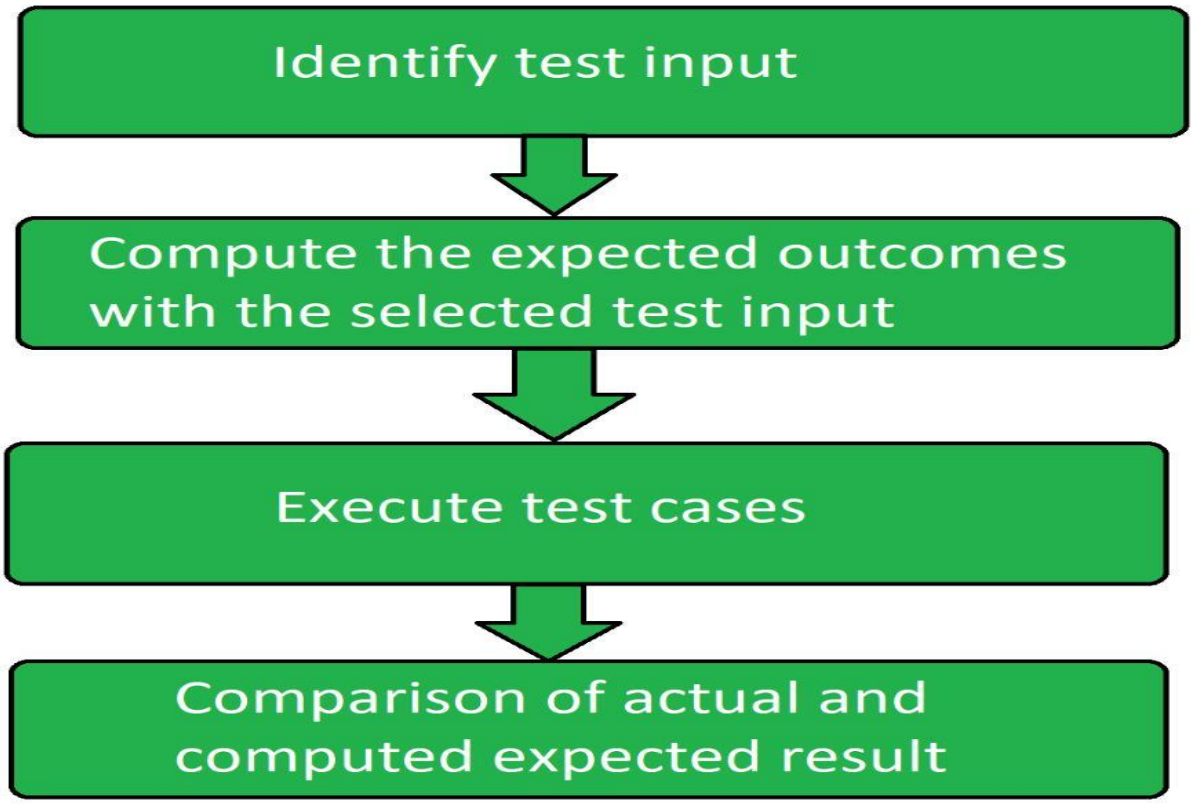
The goal of functional testing is to check the functionalities of the application under test. It concentrates on:

- **Basic Usability:** Functional testing involves basic usability testing to check whether the user can freely navigate through the screens without any difficulty.
- **Mainline functions:** This involves testing the main feature and functions of the application.
- **Accessibility:** This involves testing the accessibility of the system for the user.
- **Error Conditions:** Functional testing involves checking whether the appropriate error messages are being displayed or not in case of error conditions.

Functional Testing Process

Functional testing involves the following steps:

1. **Identify test input:** This step involves identifying the functionality that needs to be tested. This can vary from testing the usability functions, and main functions to error conditions.
2. **Compute expected outcomes:** Create input data based on the specifications of the function and determine the output based on these specifications.
3. **Execute test cases:** This step involves executing the designed test cases and recording the output.
4. **Compare the actual and expected output:** In this step, the actual output obtained after executing the test cases is compared with the expected output to determine the amount of deviation in the results. This step reveals if the system is working as expected or not.



Type of Functional Testing Techniques

1. **Unit Testing:** Unit testing is the type of functional testing technique where the individual units or modules of the application are tested. It ensures that each module is working correctly.
2. **Integration Testing:** In Integration testing, combined individual units are tested as a group and expose the faults in the interaction between the integrated units.

3. **Smoke Testing:** Smoke testing is a type of functional testing technique where the basic functionality or feature of the application is tested as it ensures that the most important function works properly.
4. **User Acceptance Testing:** User acceptance testing is done by the client to certify that the system meets the requirements and works as intended. It is the final phase of testing before the product release.
5. **Interface Testing:** Interface testing is a type of software testing technique that checks the proper interaction between two different software systems.
6. **Usability Testing:** Usability testing is done to measure how easy and user-friendly a software application is.
7. **System Testing:** System testing is a type of software testing that is performed on the complete integrated system to evaluate the compliance of the system with the corresponding requirements.
8. **Regression Testing:** Regression testing is done to make sure that the code changes should not affect the existing functionality and the features of the application. It concentrates on whether all parts are working or not.
9. **Sanity Testing:** Sanity testing is a subset of regression testing and is done to make sure that the code changes introduced are working as expected.
10. **White box Testing:** White box testing is a type of software testing that allows the tester to verify the internal workings of the software system. This includes analyzing the code, infrastructure, and integrations with the external system.
11. **Black box Testing:** Black box testing is a type of software testing where the functionality of the software system is tested without looking at the internal working or structures of the software system.
12. **Database Testing:** Database testing is a type of software testing that checks the schema, tables, etc of the database under test.
13. **Adhoc Testing:** Adhoc testing also known as monkey testing or random testing is a type of software testing that does not follow any documentation or test plan to perform testing.
14. **Recovery Testing:** Recovery testing is a type of software testing that verifies the software's ability to recover from the failures like hardware failures, software failures, crashes, etc.
15. **Static Testing:** Static testing is a type of software testing which is performed to check the defects in software without actually executing the code of the software application.
16. **Greybox Testing:** Grey box testing is a type of software testing that includes black box and white box testing.
17. **Component Testing:** Component testing also known as program testing or module testing is a type of software testing that is done after the unit testing. In this, the test objects can be tested independently as a component without integrating with other components.

Functional Testing Tools

Below are the tools for functional testing:

1. Selenium: Selenium is an open-source umbrella project for a range of tools and libraries developed with the aim to support browser automation.

- It is used to automate web browsers.
- It provides a single interface that lets the tester write test scripts in languages like Ruby, Java, NodeJS, etc.
- It provides a playback tool for authoring functional tests across most modern web browsers.

2. QTP: QTP tool now can UFT is a tool designed to perform automated functional testing without the need to monitor the system in intervals.

- It can be used to test web, desktop applications, and client-server.
- It is based on the VB scripting language.
- It is one of the widely used automation tools in the testing industry.

3. JUnit: JUnit is a unit-testing open-source framework for the Java programming language. It is used by Java developers to write and execute automated test cases.

- It can be used along with the Selenium WebDriver to automate tests for web applications.
- It provides several annotations to identify test methods.
- It has test runners to run tests.

4. SoapUI: It is one of the leading tools for SOAP and web service testing. It allows for easy and rapid creation and execution of functional, regression, and load tests.

- It has easy to use graphical interface.
- It provides a code-free test environment where one can create and execute complex test cases with drag-and-drop options.
- It lets to dynamically analyze how well SOAP and REST service contract is covered by the functional tests.

5. Cucumber: Cucumber is an open-source testing tool written in Ruby language.

- This tool focuses on end-user experience.
- Quick and easy setup and execution.
- This tool allows for easy reuse of code in tests due to the style of writing the tests.

Benefits of Functional Testing

- **Bug-free product:** Functional testing ensures the delivery of a bug-free and high-quality product.
- **Customer satisfaction:** It ensures that all requirements are met and ensures that the customer is satisfied.
- **Testing focussed on specifications:** Functional testing is focussed on specifications as per customer usage.
- **Proper working of application:** This ensures that the application works as expected and ensures proper working of all the functionality of the application.
- **Improves quality of the product:** Functional testing ensures the security and safety of the product and improves the quality of the product.

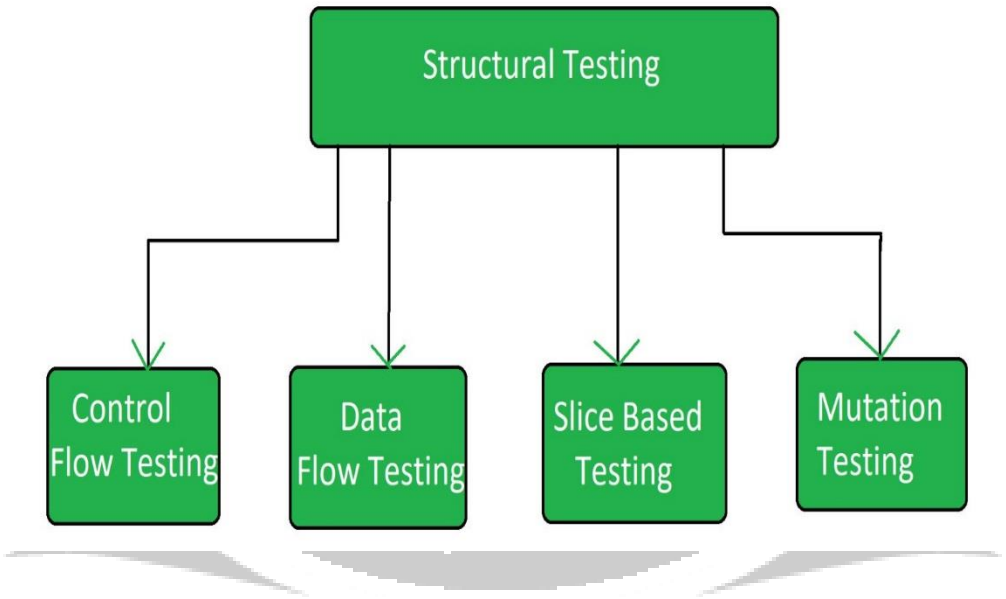
Limitations of Functional Testing

- **Missed critical errors:** There are chances while executing functional tests that critical and logical errors are missed.
- **Redundant testing:** There are high chances of performing redundant testing.
- **Incomplete requirements:** If the requirement is not complete then performing this testing becomes difficult.

Structural Testing

Structural testing is a type of software testing which uses the internal design of the software for testing or in other words the software testing which is performed by the team which knows the development phase of the software, is known as structural testing.
Structural testing is basically related to the internal design and implementation of the software i.e. it involves the development team members in the testing team. It basically tests different aspects of the software according to its types. Structural testing is just the opposite of behavioral testing.

Types of Structural Testing:
There are 4 types of Structural Testing:



Control Flow Testing:

Control flow testing is a type of structural testing that uses the programs control flow as a model. The entire code, design and structure of the software have to be known for this type of testing. Often this type of testing is used by the developers to test their own code and implementation. This method is used to test the logic of the code so that required result can be obtained.

Data Flow Testing:

It uses the control flow graph to explore the unreasonable things that can happen to data.

The detection of data flow anomalies are based on the associations between values and variables. Without being initialized usage of variables. Initialized variables are not used once.

Slice Based Testing:

It was originally proposed by Weiser and Gallagher for the software maintenance. It is useful for software debugging, software maintenance, program understanding and quantification of functional cohesion. It divides the program into different slices and tests that slice which can majorly affect the entire software.

Mutation Testing:

Mutation Testing is a type of Software Testing that is performed to design new software tests and also evaluate the quality of already existing software tests. Mutation testing is related to modification a program in small ways. It focuses to help the tester develop effective tests or locate weaknesses in the test data used for the program.

Advantages of Structural Testing:

- It provides thorough testing of the software.
- It helps in finding out defects at an early stage.
- It helps in elimination of dead code.
- It is not time consuming as it is mostly automated.

Disadvantages of Structural Testing:

- It requires knowledge of the code to perform test.
- It requires training in the tool used for testing.
- Sometimes it is expensive.

Structural Testing Tools:

- JBehave
- Cucumber
- Junit
- Cfix