***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:***

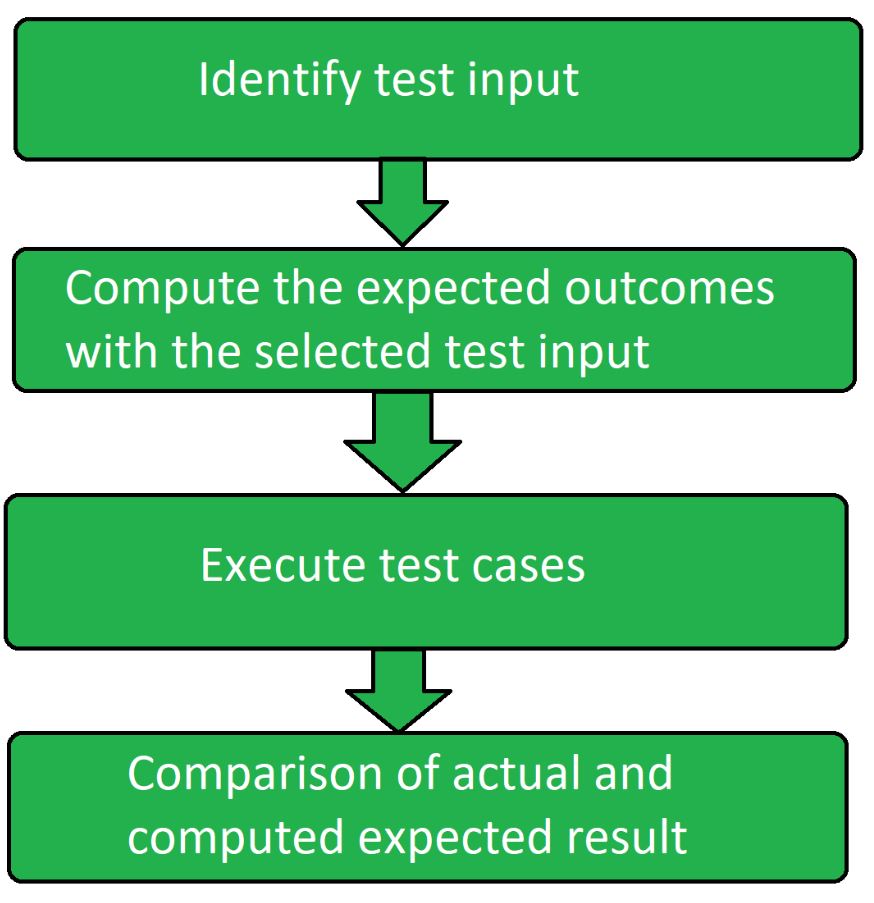
Functional testing mainly involves black box testing and can be done manually or using automation. The purpose of functional testing is to:

* **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.

***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.

***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.

***NON-FUNCTIONAL TESTING:***

Non-functional Testing is a type of Software Testing that is performed to verify the non-functional requirements of the application. It verifies whether the behavior of the system is as per the requirement or not. It tests all the aspects which are not tested in functional testing. Non-Functional testing is a software testing technique that checks the non-functional attributes of the system. Non-functional testing is defined as a type of software testing to check non-functional aspects of a software application. It is designed to test the readiness of a system as per nonfunctional parameters which are never addressed by functional testing. Non-functional testing is as important as functional testing.

***Objectives of Non-functional Testing:***

* To increase usability, efficiency, maintainability and portability of the product.
* To help in the reduction of production risk related with non-functional aspects of the product.
* To help in the reduction of cost related with non-functional aspects of the product.
* To optimize the installation, execution and monitoring way of the product.
* To collect and produce measurements and metrics for internal research and development.
* To improve and enhance knowledge of the product behavior and technologies in use.

***Non-Functional Testing Techniques:***

1. **Compatibility testing:** A type of testing to ensure that a software program or system is compatible with other software programs or systems.
2. **Compliance testing:** A type of testing to ensure that a software program or system meets a specific compliance standard, such as HIPAA or Sarbanes-Oxley.
3. **Endurance testing:** A type of testing to ensure that a software program or system can handle a long-term, continuous load.
4. **Load testing:** A type of testing to ensure that a software program or system can handle a large number of users or transactions.
5. **Performance testing:** A type of testing to ensure that a software program or system meets specific performance goals, such as response time or throughput.
6. **Recovery testing:** A type of testing to ensure that a software program or system can be recovered from a failure or data loss.
7. **Security testing:** A type of testing to ensure that a software program or system is secure from unauthorized access or attack.
8. **Scalability testing:** A type of testing to ensure that a software program or system can be scaled up or down to meet changing needs.
9. **Stress testing:** A type of testing to ensure that a software program or system can handle an unusually high load.
10. **Usability testing:** A type of testing to ensure that a software program or system is easy to use.
11. **Volume testing:** A type of testing to ensure that a software program or system can handle a large volume of data.

