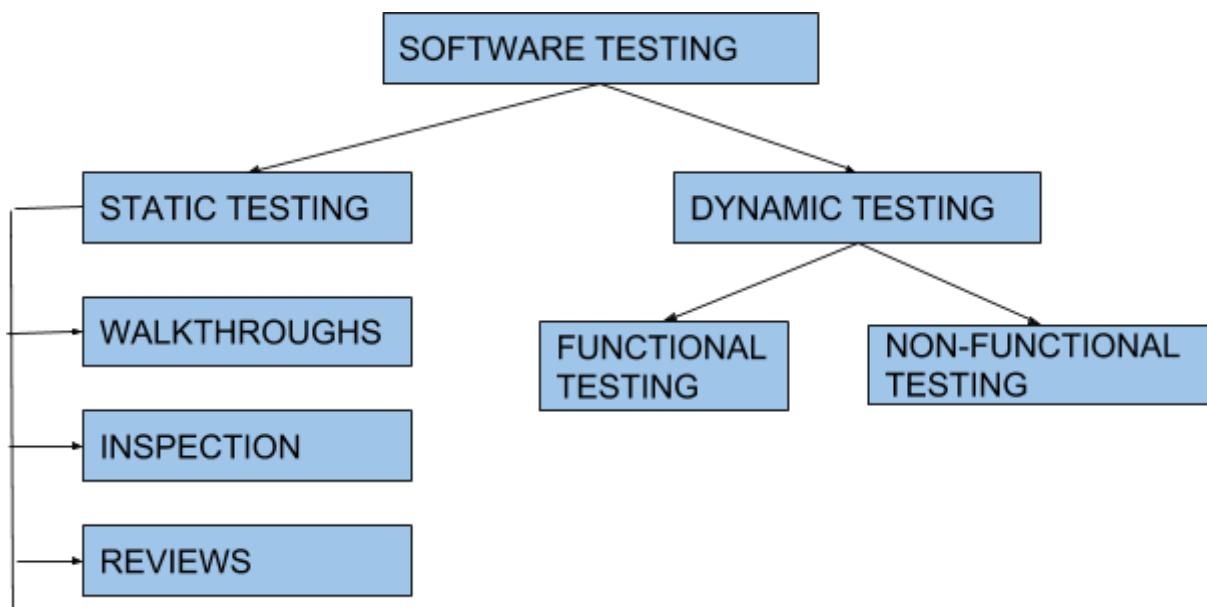


SOFTWARE TESTING

Software Testing is a process to make sure that the actual result matches the expected result. It is a process of finding bugs in the software which is being developed.

Software testing is the process of evaluating a software item to detect differences between a given input and expected output.

Software testing is broadly divided as:-



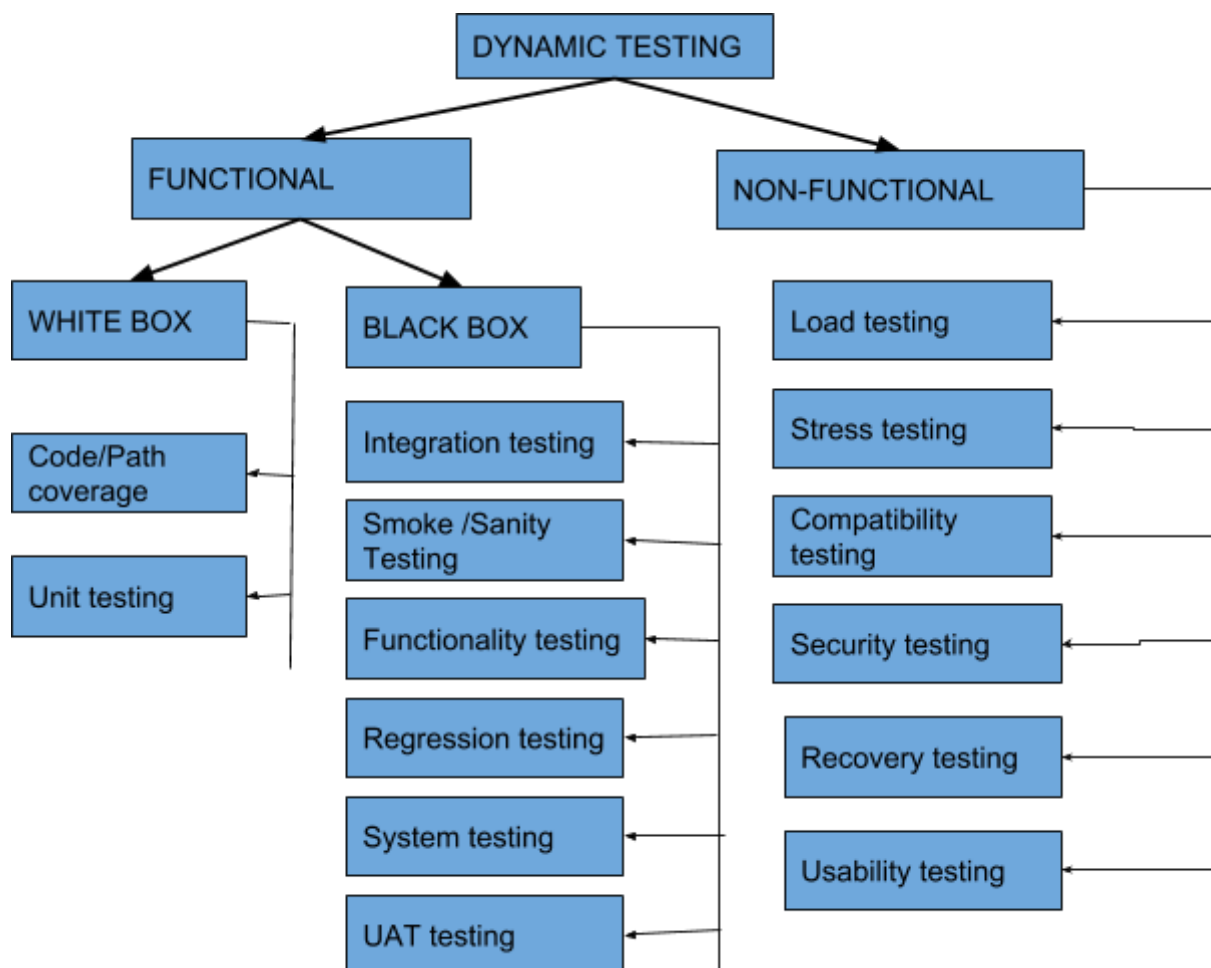
1. Static testing

- Static testing is a type of testing which is performed without actually running the software or code.
- It manually checks the code, requirement documents, and design documents to find errors.
- Static testing is a method of bug prevention.
- It is done at an early stage of software development.
- Static testing does the verification process.
- It is very cost effective.
- It focuses on reviewing software documents, discussions, meeting etc., to avoid confusion in the software requirements.

2. Dynamic testing

- Dynamic testing is a type of testing which is performed by actually running the software tool or the code.
- It is performed in a runtime environment.
- It is done after code deployment.
- It is performed to find the bugs in the developed code.
- Dynamic testing does the validation process.
- Dynamic testing needs test cases for execution.
- Cost of finding and fixing bugs is high.

Dynamic testing is of the following types:-

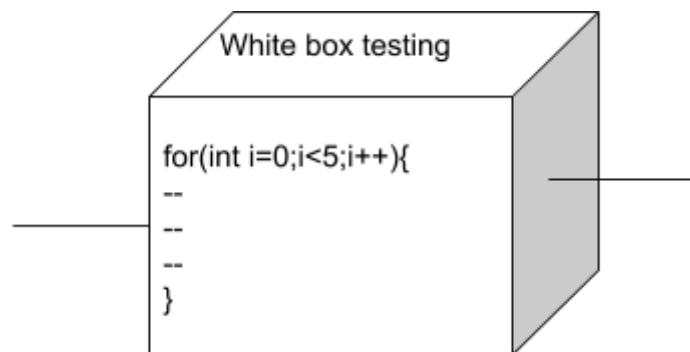


Functional testing

- Functional testing is the testing to ensure that the specified functionality required in the system requirements works.
- It checks all the functionality of the developed software.

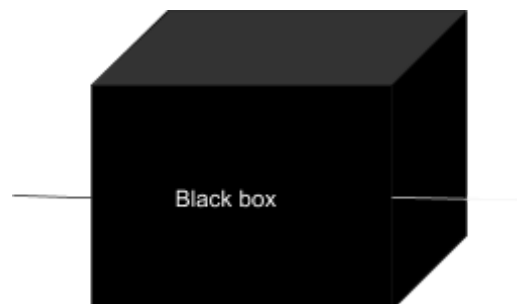
1. White box testing

- Also known as Clear box/open box/code based Testing
- It is a software testing method in which the internal structure of the code is tested.
- To conduct White Box Testing, knowledge of underlying programming language is essential.
- It includes testing of each and every individual unit of code.



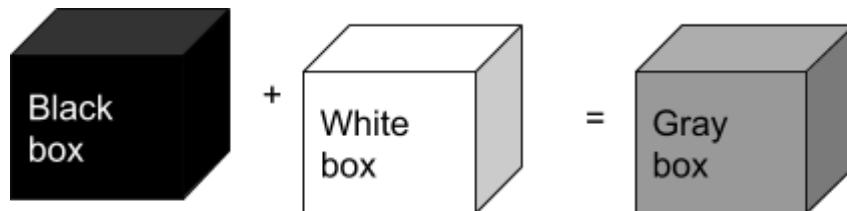
2. Black box testing

- Also known as Behavioral Testing
- It is a software testing method in which the internal structure of the code is not tested.
- It just focuses on how the system is working rather than focusing on how the code is working.
- It is a software testing method in which the internal structure of the code is tested.
- It is a software testing method in which the internal structure of the code is tested.
- It is a software testing method in which the internal structure of the code is tested.



3. Gray box testing

- It is a software testing method in which the internal structure of the code is tested.
- In the gray box testing tester is usually has knowledge of limited access of code and based on this knowledge test cases are designed.
- It offers combined benefits of black and white box testing.



The difference is as follows:-

Black box Testing	Gray box testing	White Box Testing
The internal working of an application need not to be known.	The tester has limited knowledge of the internal working of the application	The tester has full knowledge of the internal working of the application
Also known as closed box testing	Also known as translucent box testing	Also known as clear box testing
Can be performed by an end user, tester and developer	Can be performed by an end user, tester and developer	Normally done by tester and developer
It is exhaustive and less time-consuming.	It is exhaustive and partly.	It is Most exhaustive and time-consuming.

Unit testing

- It is a type of white box testing.
- It focuses on the smallest unit of software design.
- It is often done by the programmer by using sample input and observing its corresponding outputs.
- This type of testing is performed by developers before the setup is handed over to the testing team to formally execute the test cases.
- The goal of unit testing is to examine each part of the program and ensure that individual parts are correct in terms of requirements and functionality.

Integration testing

- Integration testing is defined as the testing of combined parts of an application to determine if they function correctly.
- The objective is to take unit tested components and build a program structure that has been dictated by design.
- Integration testing is testing in which a group of components is combined to produce output

Smoke testing/Sanity testing

- Smoke Testing is performed to discover that the critical functionalities of the program are working fine.
- The objective of this testing is to verify the "stability" of the system in order to proceed with more rigorous testing.
- Smoke testing is a subset of Regression testing.
- Smoke testing is like General Health Check Up.
- Smoke testing exercises the entire system from end to end.
- Sanity Testing is done to check the new functionality/bugs have been fixed.
- Both sanity tests and smoke tests are ways to avoid wasting time and effort by quickly determining whether an application is too flawed to merit any rigorous testing.
- Both smoke and sanity tests can be executed manually or using an automation tool. When automated tools are used, the tests are often initiated by the same process that generates the build itself.

Sanity Testing	Smoke Testing
Sanity Testing is done to check the new functionality / bugs have been fixed.	Smoke Testing is performed to discover that the critical functionalities of the program are working fine.
The objective of the testing is to	The objective of this testing is to

verify the "rationality" of the system in order to proceed with more rigorous testing.	verify the "stability" of the system in order to proceed with more rigorous testing.
Sanity testing is usually performed by testers.	This testing is performed by the developers or testers.
Sanity testing is usually not documented and is unscripted.	Smoke testing is usually documented or scripted.
Sanity testing is a subset of Acceptance testing.	Smoke testing is a subset of Regression testing.
Sanity testing exercises only a particular component of the entire system.	Smoke testing exercises the entire system from end to end.
Sanity Testing is like a specialized health check-up.	Smoke testing is like General Health Check Up.

Functionality testing

- Functional testing is the testing to ensure that the specified functionality required in the system requirements works.
- Functionality testing verifies that an application is fully functional.

Regression testing

- Functionality testing verifies that an application is still fully functional.
- Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly.
- Regression testing is performed to verify that a fixed bug hasn't resulted in another functionality violation.

System testing

- System testing tests the system as a whole.
- System testing is the one where an application is tested as a whole.
- This type of testing is performed by a specialized testing team.

UAT testing

- User acceptance testing.
- Acceptance testing is often done by the customer to ensure that the delivered product meets the requirements and works as the customer expected.

- User acceptance testing, a testing methodology where the clients/end users involved in testing the product to validate the product against their requirements

UAT can be divided as:-

Alpha testing

- Alpha testing takes place at the developer's site by the internal teams, before releasing to external customers.
- This testing is performed without the involvement of the development teams.

Beta testing

- Beta testing is performed by some selected users and based on their feedback final release is designed.

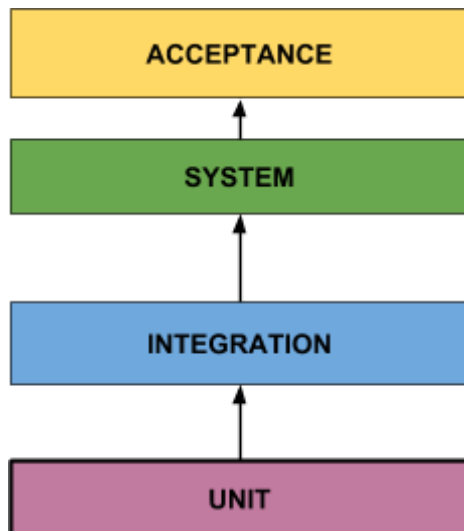
Non- functional testing

- The non Functional Testing is the type of testing done against the nonfunctional requirements.
- Non-functional requirements tend to be those that reflect the quality of the product, particularly in the context of the suitability perspective of its users.
- It can be started after the completion of Functional Testing.
- The testing of software attributes which are not related to any specific function or user action like performance, scalability, security or behavior of application under certain constraints.

NOTE:-

Software testing is divided into different levels as:-

1. Unit testing
 - Unit testing is performed on individual units of software.
2. Integration testing
 - Integration testing is performed by combining individual units together to see how they are behaving together.
3. System testing
 - System testing is performed by taking the whole system or product into consideration.
 - How the developed software behaves as a complete product.
4. Acceptance testing
 - It is final testing which is performed to check the overall performance and usability of the system.



Note:-

On the basis of the way of testing, software testing is divided as

1. Manual testing
 - In manual testing, tester performs testing manually.
 - It is performed without using any automation tool.
 - With time efficiency of manual tester decreases, so it's very hectic and tedious.
 - It is important too because you can automate everything.
2. Automation testing
 - In Automation testing, tester writes script and use automation tool.
 - More efficient and performance remains constant.
 - Time-saving and good for repeated work.