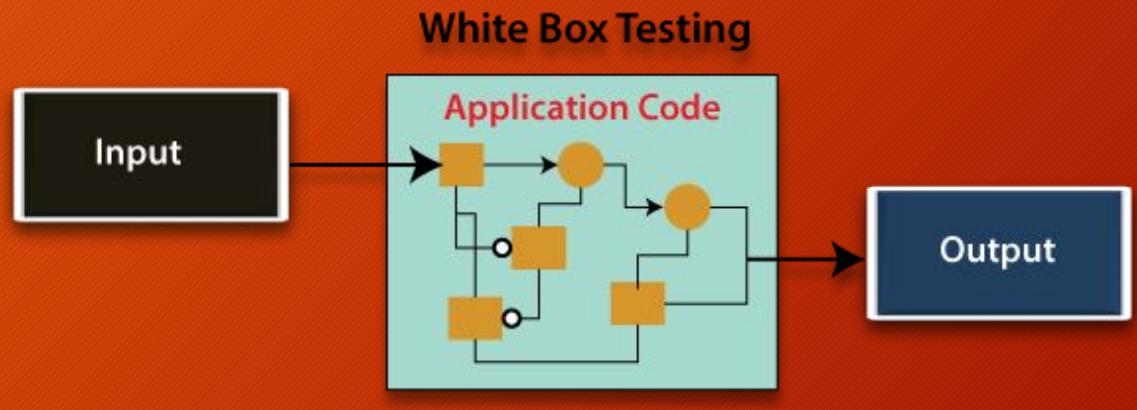
Types of Testing

White Box Testing

- In white-box testing, the developer will inspect every line of code before handing it over to the testing team or the concerned test engineers.

White Box Testing

- Subsequently, the code is noticeable for developers throughout testing; that's why this process is known as WBT (White Box Testing).
- In other words, we can say that the developer will execute the complete white-box testing for the particular software and send the specific application to the testing team.
- The purpose of implementing the white box testing is to emphasize the flow of inputs and outputs over the software and enhance the security of an application.
- White box testing is also known as open box testing, glass box testing, structural testing, clear box testing, and transparent box testing.



Black Box Testing

- In this testing, the test engineer will analyze the software against requirements, identify the defects or bug, and sends it back to the development team.

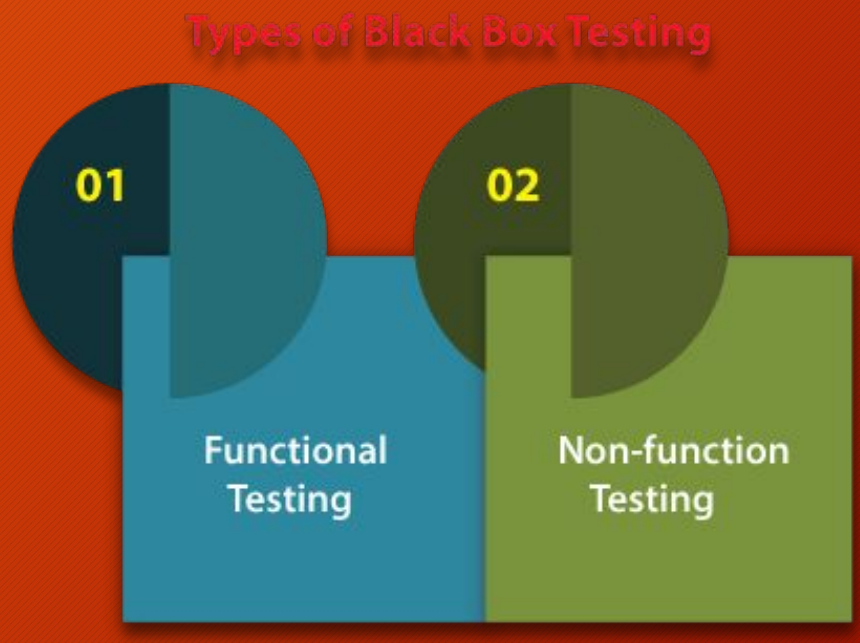
Black Box Testing

- In other words, we can say that black box testing is a process of checking the functionality of an application as per the customer requirement. The source code is not visible in this testing; that's why it is known as black-box testing.

Types of Black Box Testing

Black box testing further categorizes into two parts, which are as discussed below:

- Functional Testing
- Non-function Testing



Functional Testing

- The test engineer will check all the components systematically against requirement specifications is known as functional testing. Functional testing is also known as Component testing.

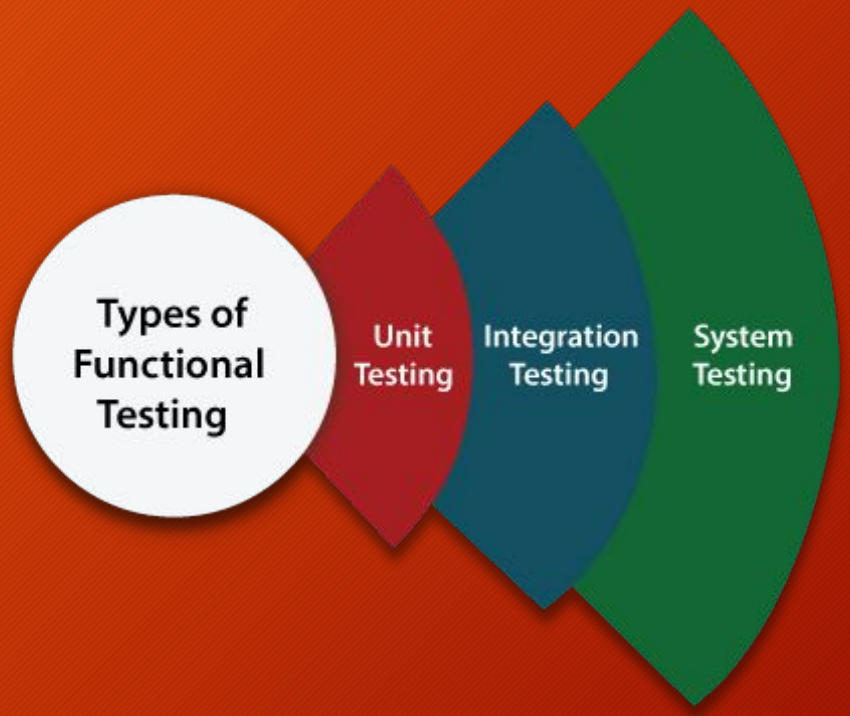
- In functional testing, all the components are tested by giving the value, defining the output, and validating the actual output with the expected value.

- Functional testing is a part of black-box testing as its emphases on application requirement rather than actual code. The test engineer has to test only the program instead of the system.

Types of Functional Testing

- The diverse types of Functional Testing contain the following:

- Unit Testing
- Integration Testing
- System Testing



Unit Testing

- Unit testing is the first level of functional testing in order to test any software. In this, the test engineer will test the module of an application independently or test all the module functionality is called unit testing.

- The primary objective of executing the unit testing is to confirm the unit components with their performance. Here, a unit is defined as a single testable function of a software or an application. And it is verified throughout the specified application development phase.

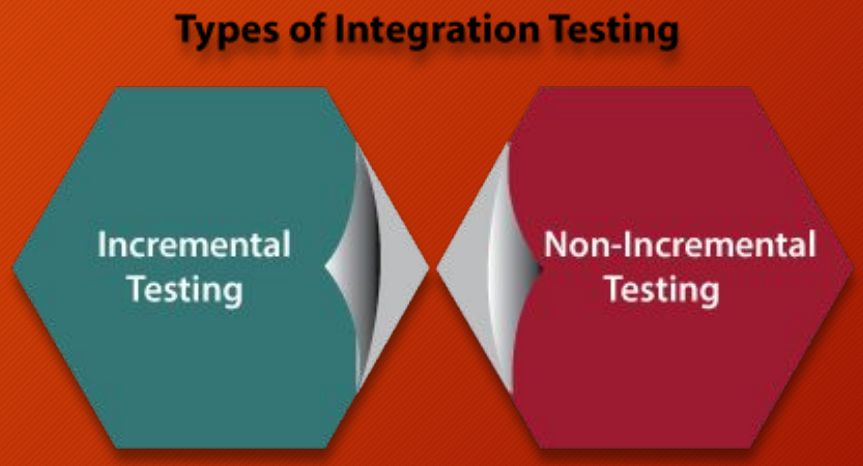
Integration Testing

- Once we are successfully implementing the unit testing, we will go integration testing. It is the second level of functional testing, where we test the data flow between dependent modules or interface between two features is called integration testing.
- The purpose of executing the integration testing is to test the statement's accuracy between each module.

Types of Integration Testing

- Integration testing is also further divided into the following parts:

- Incremental Testing
- Non-Incremental Testing



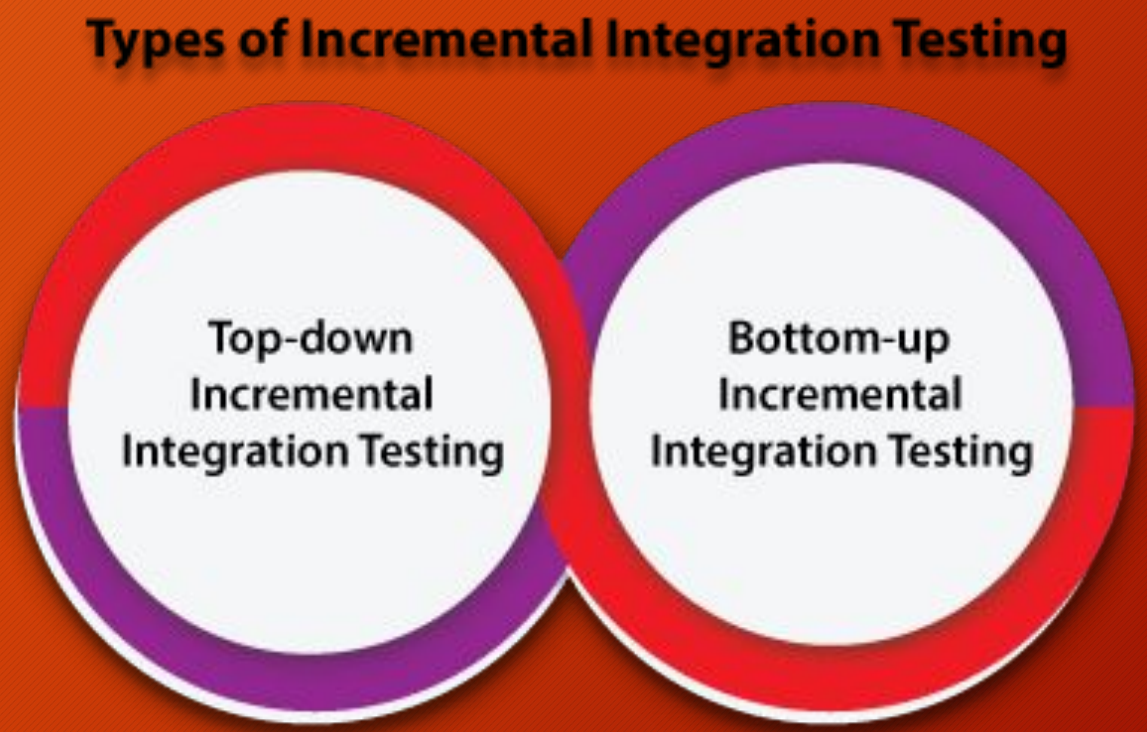
Incremental Integration Testing

- Whenever there is a clear relationship between modules, we go for incremental integration testing. Suppose, we take two modules and analysis the data flow between them if they are working fine or not.
- If these modules are working fine, then we can add one more module and test again. And we can continue with the same process to get better results.
- In other words, we can say that incrementally adding up the modules and test the data flow between the modules is known as Incremental integration testing.

Types of Incremental Integration Testing

- Incremental integration testing can further classify into two parts, which are as follows:

- Top-down Incremental Integration Testing
- Bottom-up Incremental Integration Testing



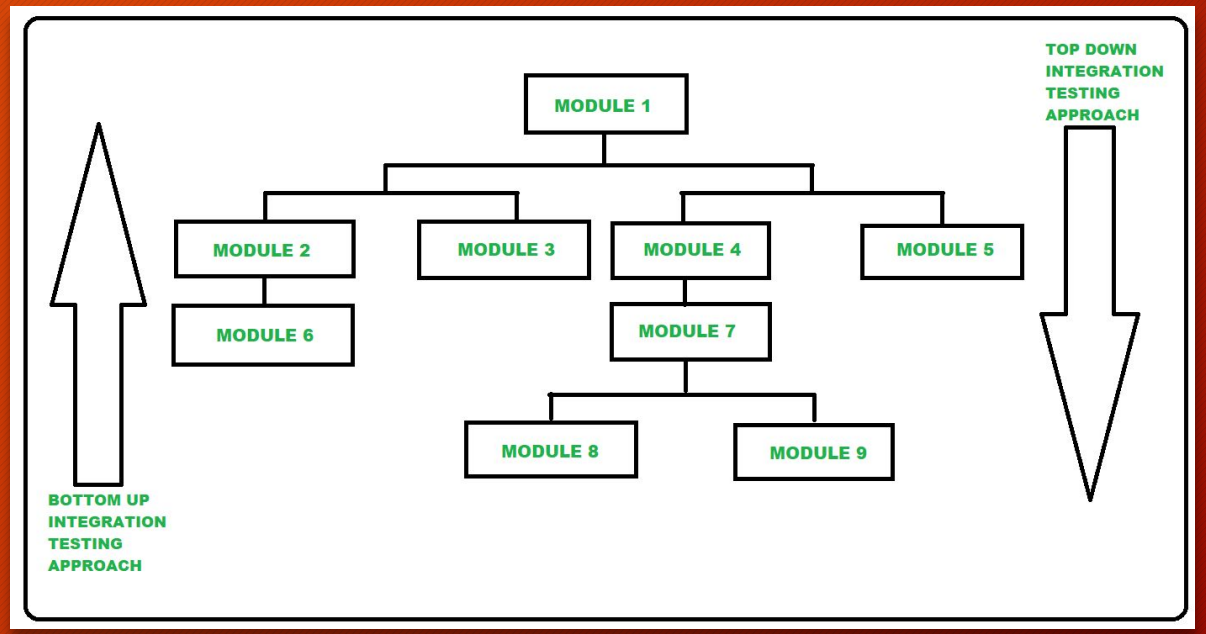
Types of Incremental Integration Testing

- Top-down Incremental Integration Testing

In this approach, we will add the modules step by step or incrementally and test the data flow between them. We have to ensure that the modules we are adding are the child of the earlier ones.

- Bottom-up Incremental Integration Testing

In the bottom-up approach, we will add the modules incrementally and check the data flow between modules. And also, ensure that the module we are adding is the parent of the earlier ones.



Non-Incremental Integration Testing/ Big Bang Method

- Whenever the data flow is complex and very difficult to classify a parent and a child, we will go for the non-incremental integration approach. The non-incremental method is also known as the Big Bang method.

System Testing

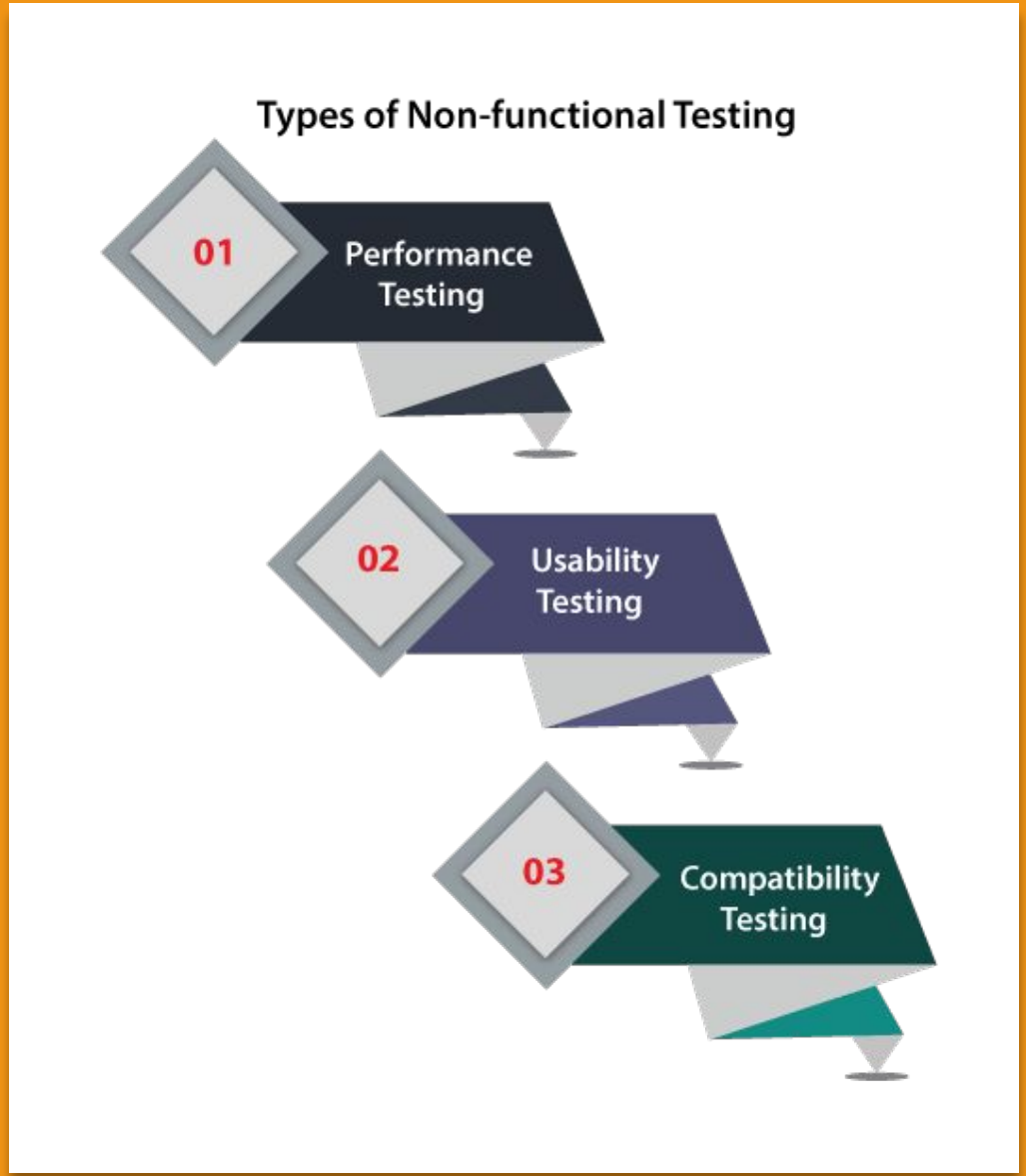
- Whenever we are done with the unit and integration testing, we can proceed with the system testing.
- In system testing, the test environment is parallel to the production environment. It is also known as end-to-end testing.
- In this type of testing, we will undergo each attribute of the software and test if the end feature works according to the business requirement. And analysis the software product as a complete system.

Non-function Testing

- The next part of black-box testing is non-functional testing. It provides detailed information on software product performance and used technologies.
- Non-functional testing will help us minimize the risk of production and related costs of the software.
- Non-functional testing is a combination of performance, load, stress, usability and, compatibility testing.

Types of Non-functional Testing

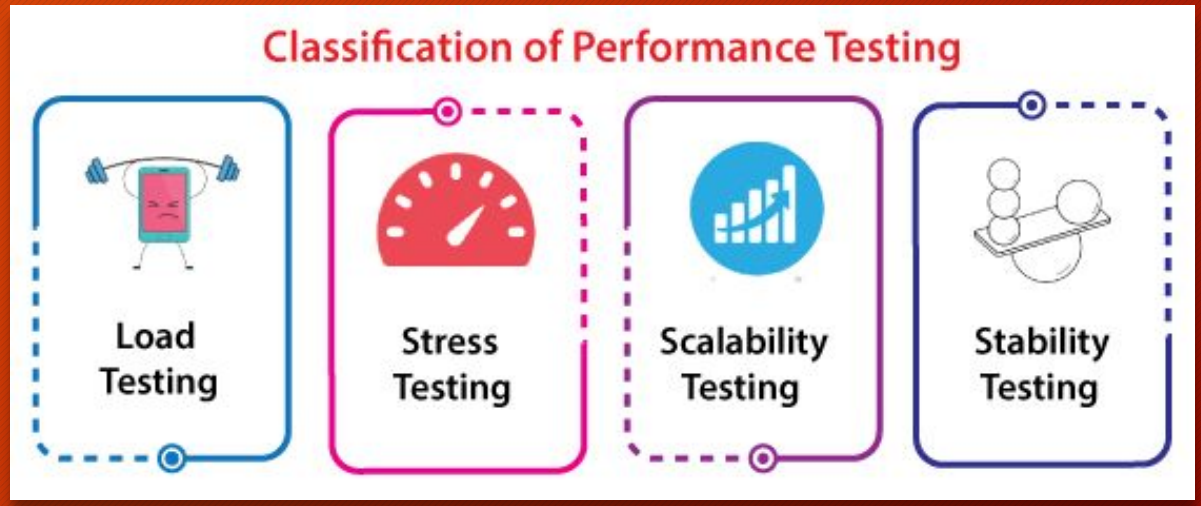
- Performance Testing
- Usability Testing
- Compatibility Testing



1.Performance Testing

- In performance testing, the test engineer will test the working of an application by applying some load.

- In this type of non-functional testing, the test engineer will only focus on several aspects, such as Response time, Load, scalability, and Stability of the software or an application.



Load Testing

- While executing the performance testing, we will apply some load on the particular application to check the application's performance, known as load testing. Here, the load could be less than or equal to the desired load.
- It will help us to detect the highest operating volume of the software and bottlenecks.
- The load testing is mainly used to test the Client/Server's performance and applications that are web-based.
- The load testing is used to perform the maximum quantity of software applications without important performance breakdown.

Stress Testing

- It is used to analyze the user-friendliness and robustness of the software beyond the common functional limits.
- In other words, we can say that Stress testing is used to verify the constancy and dependability of the system and also make sure that the system would not crash under disaster circumstances.
- To analyses how the system works under extreme conditions, we perform stress testing outside the normal load.
- The primary purpose of executing the stress testing is to confirm that the software does not crash in lacking computational resources like disk space, memory, and network request.
- The implementation of stress testing certifies that the system fails and improves effortlessly, known as the recoverability process.

Scalability Testing

- Checks the performance of an application by increasing or decreasing the load in particular scales like number of a user.
- For example, a web page scalability testing depends on the number of users, CPU usage, network usage. In contrast, scalability testing of a web server depends on the number of requests processed.

Stability Testing

- Stability testing is a software testing procedure where we analyze the application's performance by applying the load for a particular duration of time.
- It helps us to detect the bugs when the system is pushed to strict circumstances and fix those bugs or defects that can increase the software's constancy or the application.

2. Usability Testing

- Another type of non-functional testing is usability testing. In usability testing, we will analyze the user-friendliness of an application and detect the bugs in the software's end-user interface.

Here, the term user-friendliness defines the following aspects of an application:

- The application should be easy to understand, which means that all the features must be visible to end-users.
- The application's look and feel should be good that means the application should be pleasant looking and make a feel to the end-user to use it.

3. Compatibility Testing

- In compatibility testing, we will check the functionality of an application in specific hardware and software environments. Once the application is functionally stable then only, we go for compatibility testing.

- Here, software means we can test the application on the different operating systems and other browsers, and hardware means we can test the application on different sizes.

Some other Types of testing

- **Alpha Testing** will be conducted at the developer's site. An in-house virtual user environment can be created for this type of testing.
- An Acceptance Test is performed by the client, and it verifies whether the end-to-end flow of the system is as per the business requirements or not and if it is as per the needs of the end-user.

- Client accepts the software only when all the features and functionalities work as expected. This is the last phase of testing, after which the software goes into production. This is also called User Acceptance Testing (UAT).

Beta Testing

- Beta Testing is a formal type of Software Testing which is carried out by the customer. It is performed in the Real Environment before releasing the product to the market for the actual end-users.
- Beta Testing is carried out to ensure that there are no major failures in the software or product, and it satisfies the business requirements from an end-user perspective. Beta Testing is successful when the customer accepts the software.
- Usually, this testing is typically done by the end-users or others. This is the final testing done before releasing the application for commercial purposes. Usually, the Beta version of the software or product released is limited to a certain number of users in a specific area.
- So the end-user uses the software and shares the feedback with the company. The company then takes necessary action before releasing the software worldwide.