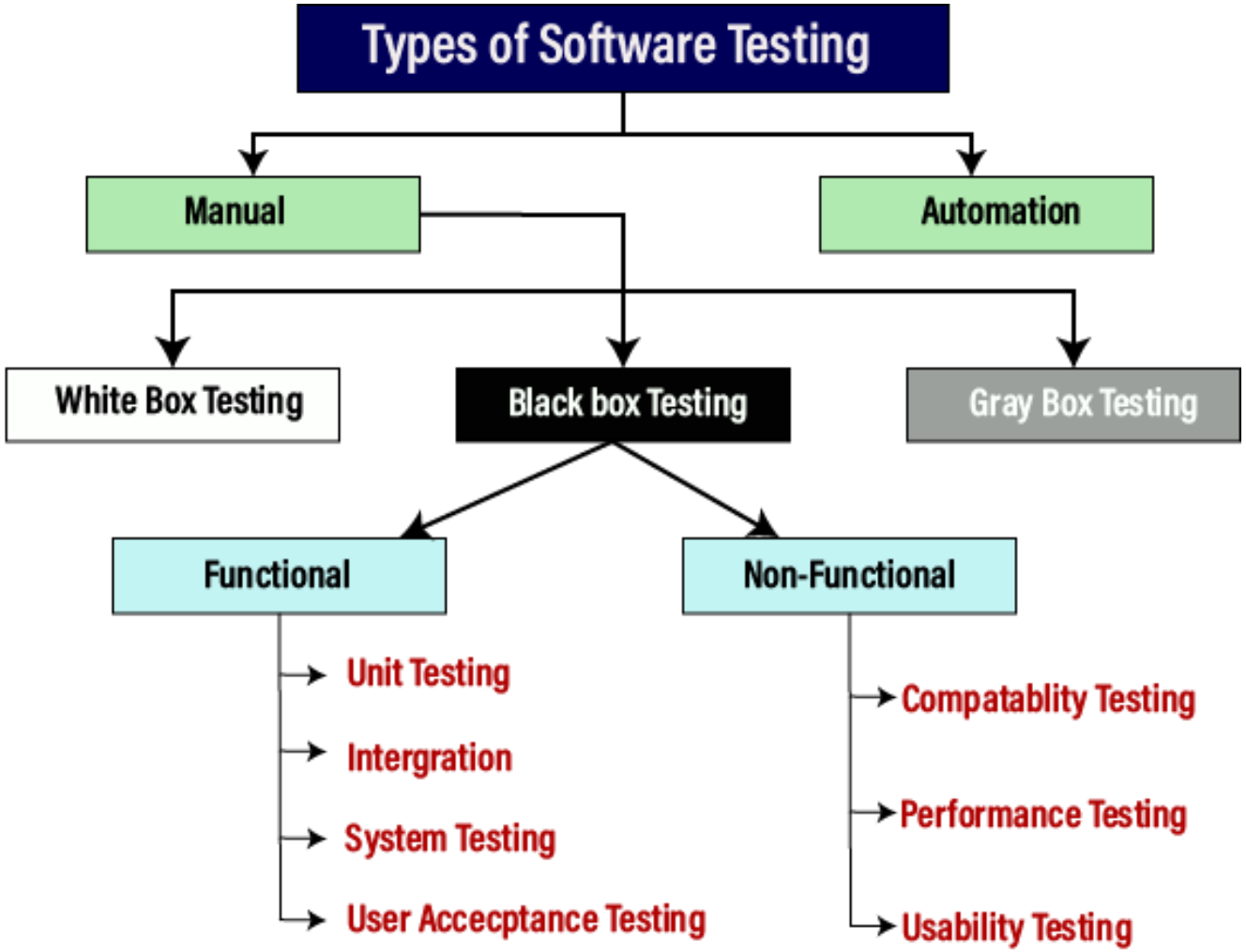
CSE 411
Software Engineering and System Analysis and Design

Topic 7: Software Testing

Software Testing

Software testing is a process of identifying the correctness of software by considering its all attributes (Reliability, Scalability, Portability, Re-usability, Usability) and evaluating the execution of software components to find the software bugs or errors or defects.

Type of Software testing



Manual testing

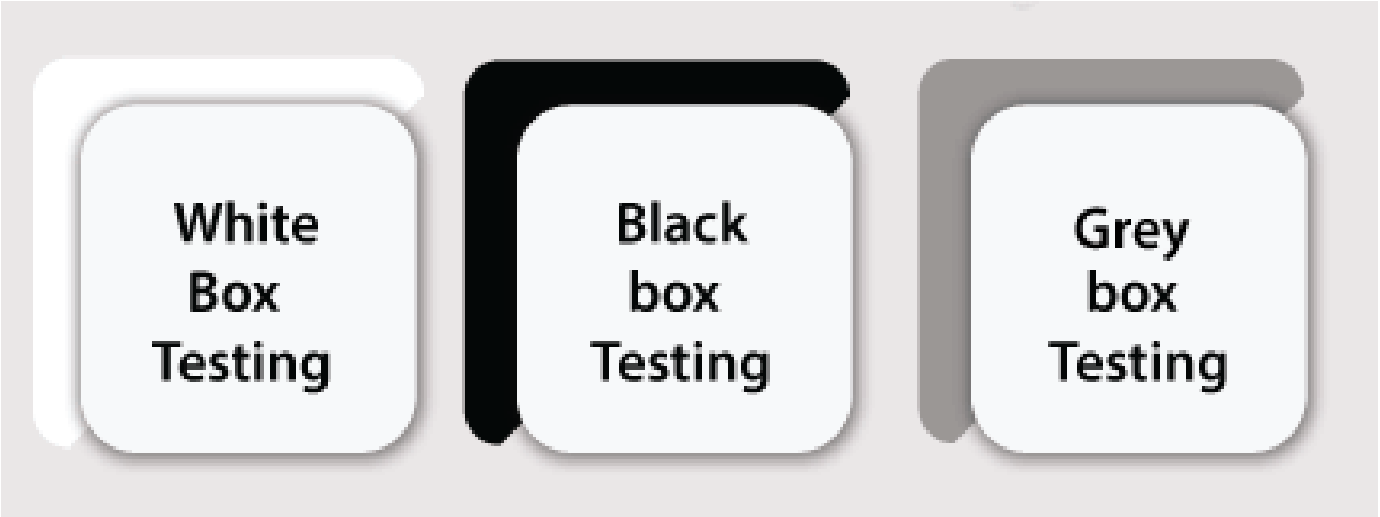
- The process of checking the functionality of an application as per the customer needs without taking any help of automation tools is known as manual testing.

- Do not need any specific knowledge of any testing tool, rather than have a proper understanding of the product so we can easily prepare the test document.

Classification of Manual Testing

In software testing, manual testing can be further classified into three different types of testing, which are as follows:

- White Box Testing
- Black Box Testing
- Grey Box Testing

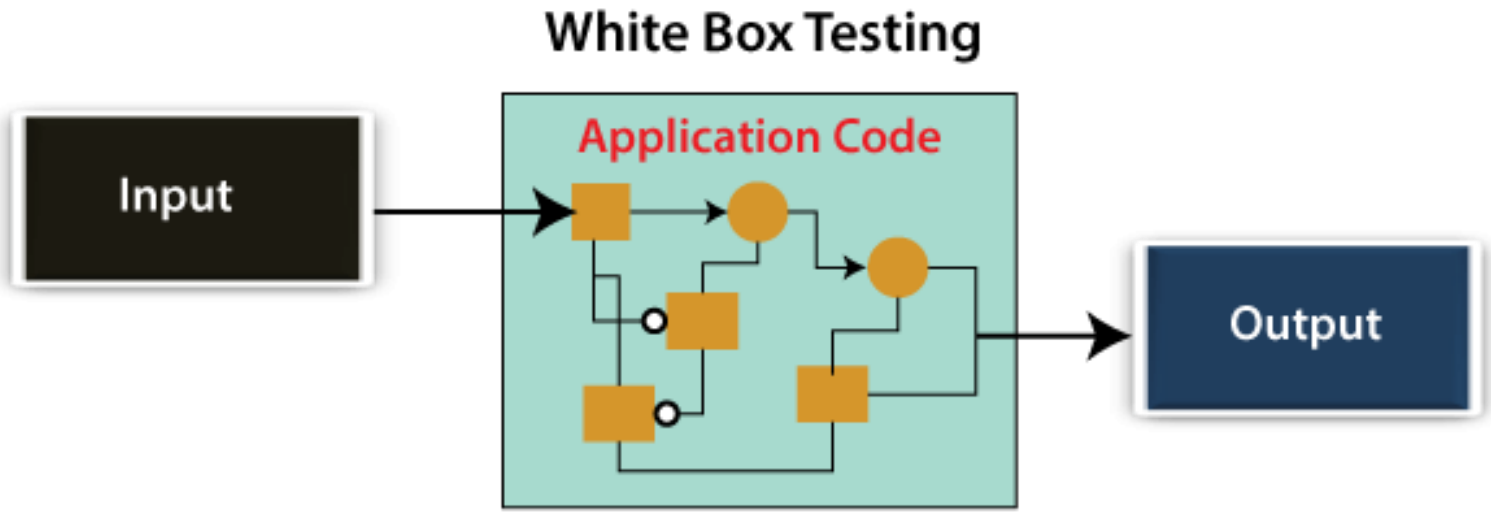


White Box Testing

- 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.
- Subsequently, the code is noticeable for developers throughout testing; that's why this process is known as WBT (White Box Testing).
- 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

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

Another type of manual testing is 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

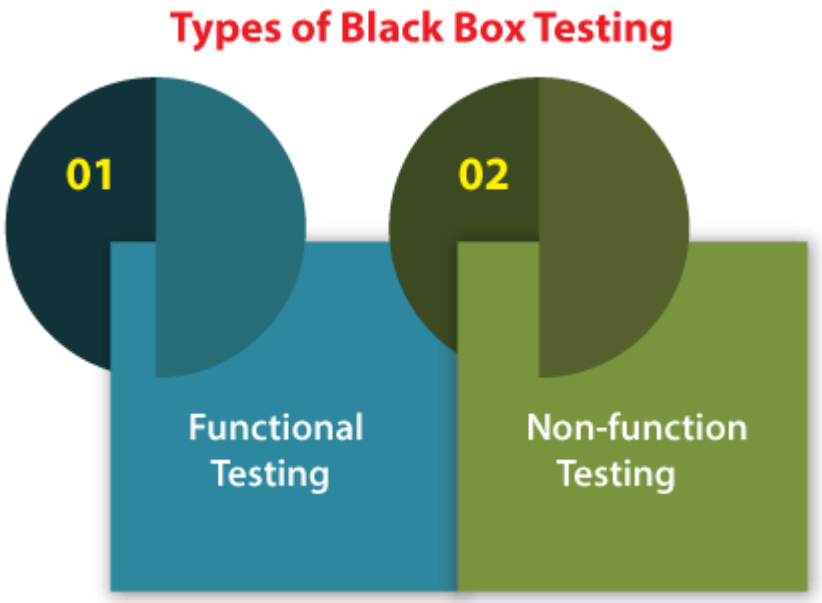
Black Box Testing

- Then, the developers will fix those defects, do one round of White box testing, and send it to the testing team.
- Here, fixing the bugs means the defect is resolved, and the particular feature is working according to the given requirement.
- The main objective of implementing the black box testing is to specify the business needs or the customer's requirements.
- 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

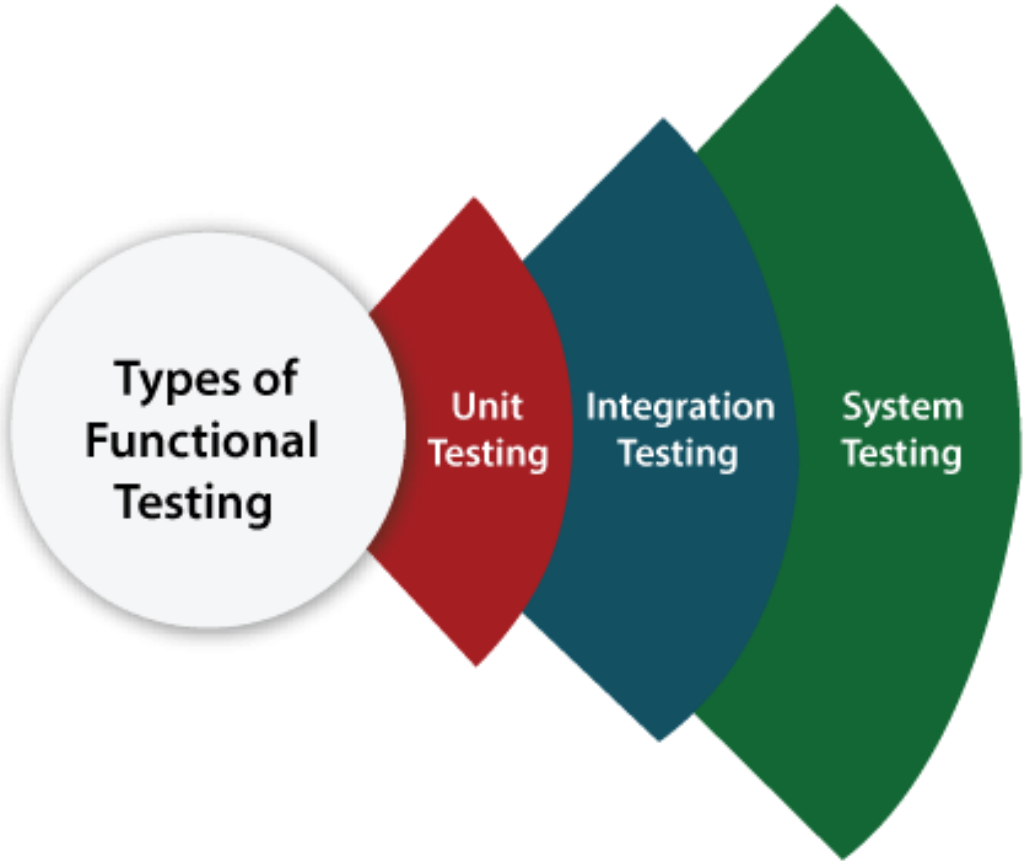
- The test engineer will check all the components systematically against requirement specifications is known as functional testing. Functional testing is also known as Component testing.

- 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

Types of Functional Testing

- Unit Testing
- Integration Testing
- System Testing



1. 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.

2. Integration Testing

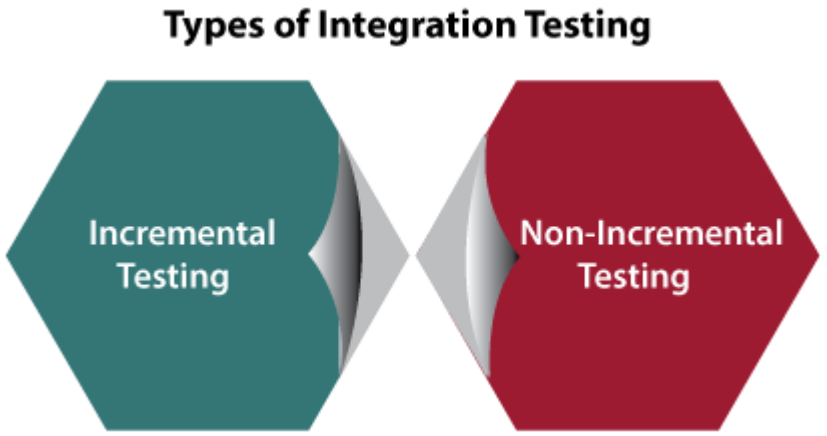
- Once we are successfully implementing the unit testing, we will go integration testing. It is the second level of functional testing, where we test the data flow between dependent modules or interface between two features is called integration testing.

- 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

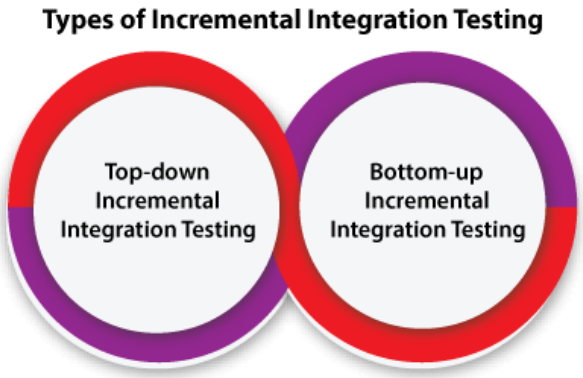
- Whenever there is a clear relationship between modules, we go for incremental integration testing. Suppose, we take two modules and analysis the data flow between them if they are working fine or not.

- 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.

Types of Incremental Integration Testing

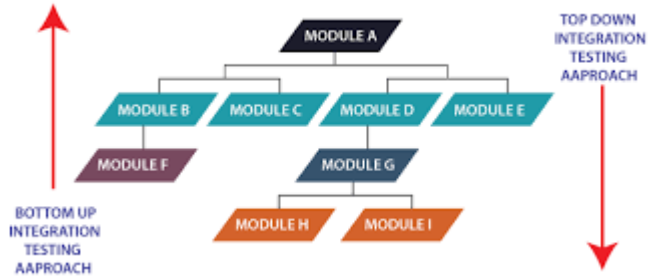
Incremental integration testing can further classify into two parts, which are as follows:

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



1. 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.



2. 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.

3. 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.

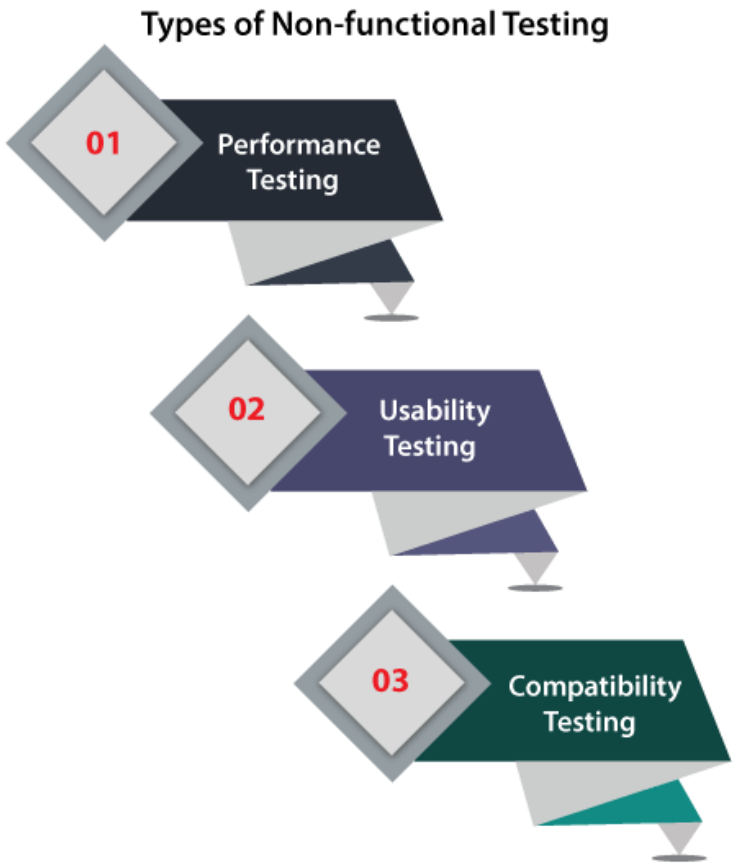
- Non-functional testing will help us minimize the risk of production and related costs of the software.

- Non-functional testing is a combination of performance, load, stress, usability and, compatibility testing.

Types of Non-functional Testing

Non-functional testing:

- Performance Testing
- Usability Testing
- Compatibility Testing

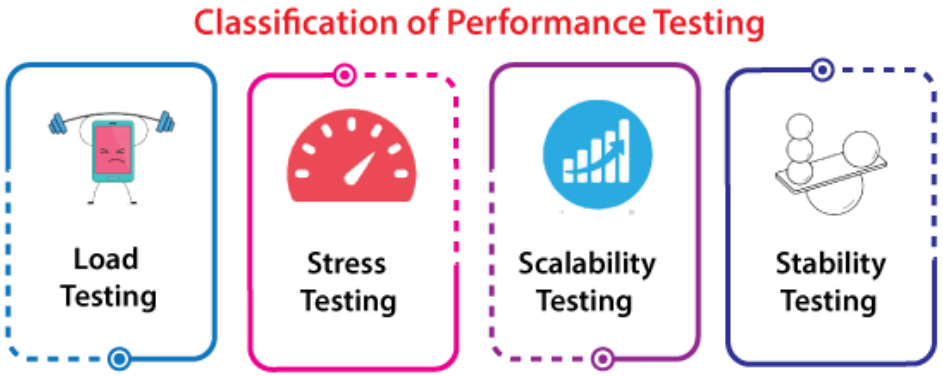


1. Performance Testing

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.

Classification of Performance Testing

- Load Testing
- Stress Testing
- Scalability Testing
- Stability Testing



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.

Stress Testing

- It is used to analyze the user-friendliness and robustness of the software beyond the common functional limits.

- Primarily, stress testing is used for critical software, but it can also be used for all types of software applications.

Scalability Testing

- To analysis, the application's performance by enhancing or reducing the load in particular balances is known as scalability testing.

- In scalability testing, we can also check the system, processes, or database's ability to meet an upward need. And in this, the Test Cases are designed and implemented efficiently.

Stability Testing

- Stability testing is a procedure where we evaluate the application's performance by applying the load for a precise time.

- It mainly checks the constancy problems of the application and the efficiency of a developed product. In this type of testing, we can rapidly find the system's defect even in a stressful situation.

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.

Grey Box Testing

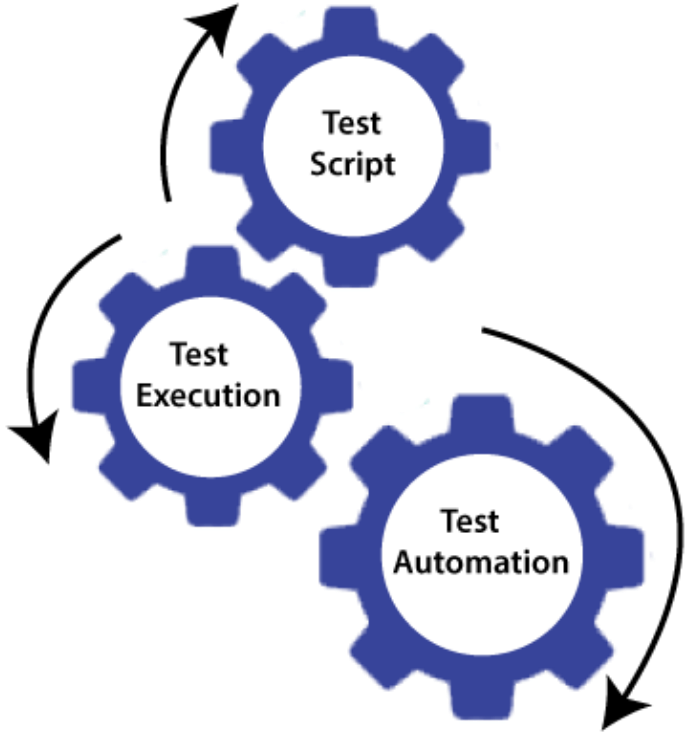
- Another part of manual testing is Grey box testing. It is a collaboration of black box and white box testing.

- Since, the grey box testing includes access to internal coding for designing test cases. Grey box testing is performed by a person who knows coding as well as testing.

- In other words, we can say that if a single-person team done both **white box and black-box testing**, it is considered **grey box testing**.

Automation testing

- The most significant part of Software testing is Automation testing. It uses specific tools to automate manual design test cases without any human interference.
- Automation testing is the best way to enhance the efficiency, productivity, and coverage of Software testing.
- It is used to re-run the test scenarios, which were executed manually, quickly, and repeatedly.



Some other types of Software Testing

- Smoke Testing
- Sanity Testing
- Regression Testing
- User Acceptance Testing
- Exploratory Testing
- Adhoc Testing
- Security Testing
- Globalization Testing

End of Topic 7