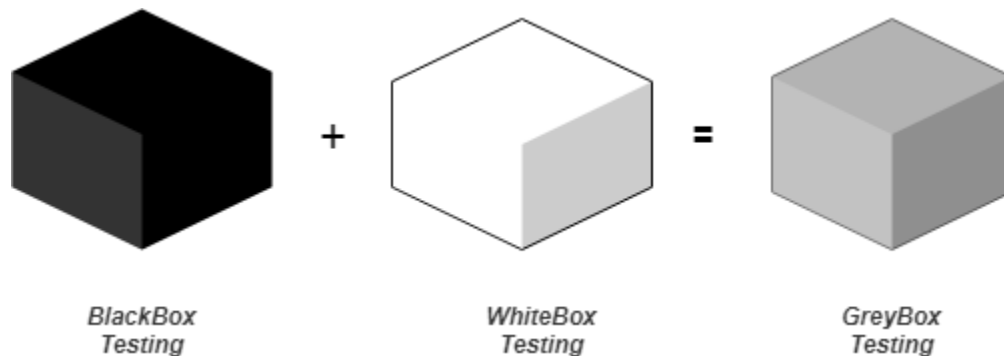


## Gray Box Testing

Grey-box testing is a software testing method to test the software application with partial knowledge of the internal working structure. It is a **combination of black box and white box testing** because it involves access to internal coding to design test cases as white box testing and testing practices are done at functionality level as black box testing.



### Why we use Gray Box Testing?

- It provides combined benefits of both Blackbox testing and WhiteBox testing.
- It includes the input values of both developers and testers at the same time to improve the overall quality of the product.
- It reduces time consumption of long process of functional and non-functional testing.
- It gives sufficient time to the developer to fix the product defects.
- It includes user point of view rather than designer or tester point of view.
- It involves examination of requirements and determination of specifications by user point of view deeply.

### Strategy Behind Gray Box Testing-

- Grey box testing does not make necessary that the tester must design test cases from source code.
- To perform this testing test cases can be designed on the base of, knowledge of architectures, algorithm, internal states or other high -level descriptions of the program behavior.
- It uses all the straight-forward techniques of black box testing for function testing.

## Gray Box Techniques-

- Matrix Testing
- OAT or Orthogonal Array Testing
- Pattern Testing

### Matrix Testing-

This testing technique comes under Grey Box testing. It defines all the used variables of a particular program.

In any program, variable are the elements through which values can travel inside the program. It should be as per requirement otherwise; it will reduce the readability of the program and speed of the software.

Matrix technique is a method to remove unused and uninitialized variables by identifying used variables from the program.

### OAT-

The purpose of this testing is to cover maximum code with minimum test cases. Test cases are designed in a way that can cover maximum code as well as GUI functions with a smaller number of test cases.

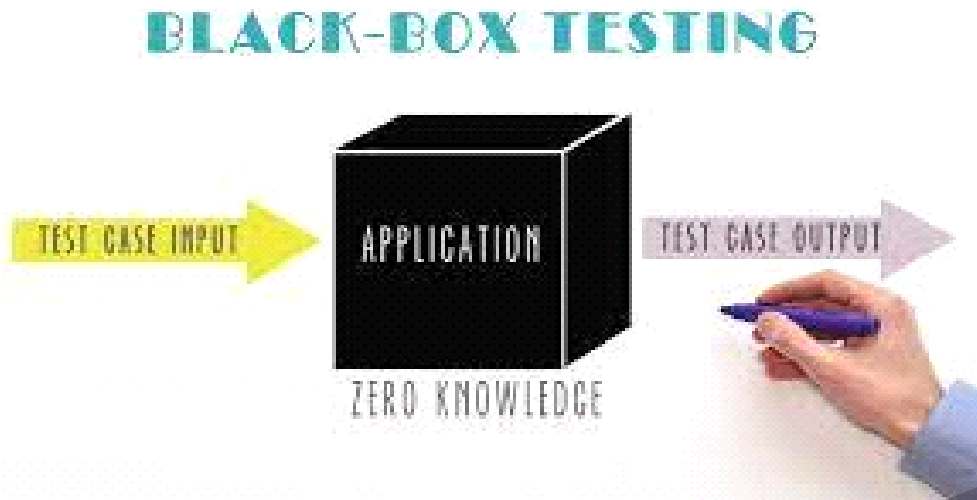
### Pattern Testing-

Pattern testing is applicable to such type of software that is developed by following the same pattern of previous software.

In this type of software possibility to occur the same type of defects.

Pattern testing determines reasons of the failure so they can be fixed in the next software.

# Black Box Testing



- Black box testing is a technique of software testing which examines the functionality of software without peering into its internal structure or coding.
- The primary source of black box testing is a specification of requirements that is stated by the customer.
- In this method, tester selects a function and gives input value to examine its functionality, and checks whether the function is giving expected output or not.
- If the function produces correct output, then it is passed in testing, otherwise failed.
- The test team reports the result to the development team and then tests the next function.
- After completing testing of all functions if there are severe

problems, then it is given back to the development team for correction.

## Steps of black box testing

- The black box test is based on the specification of requirements, so it is examined in the beginning.
- In the second step, the tester creates a positive test scenario and an adverse test scenario by selecting valid and invalid input values to check that the software is processing them correctly or incorrectly.
- In the third step, the tester develops various test cases such as decision table, all pairs test, equivalent division, error estimation, cause-effect graph, etc.
- The fourth phase includes the execution of all test cases.
- In the fifth step, the tester compares the expected output against the actual output.
- In the sixth and final step, if there is any flaw in the software, then it is cured and tested again.

## Techniques of Black box testing-

1. Equivalence Partitioning Technique
2. Boundary Value Technique
3. Decision Table Technique
4. State Transition Technique

## 5. Error Guessing Technique

## 6. All-pair Testing Technique

### Equivalence Partitioning Technique-

As the name suggests, the inputs are partitioned into groups or more literally partitions. Only one input from every group is tested to find the results. The inputs are usually numeric values or a set of values or Boolean conditions.

For example, if the field accepts an integer in the range 1 and 20 then: Valid Equivalence Class Partition: 1 to 20 inclusive. Invalid Equivalence Class Partition: Less than 1 or more than 20, decimal numbers or alphabets and other non-numeric characters.

### Boundary Value Technique-

In boundary value analysis the answers are within specific boundaries. The two ends, the inner and the outer limits are considered in this type of testing.

For example, an offer is valid for customers between the ages of 18 and 30 only. Therefore, other values such as 17, 18, 30, or 31 can be tested to check whether the inputs are accepted.

### Decision Table Technique-

If the results are obtained depending on certain conditions it comes under Decision table testing. If the conditions are combined, they form rules. The outputs of these rules are formed and the test cases are structured.

For example, the government provides different schemes depending on the age of the citizen. Whether they are minors or above 18 or retired makes them eligible for different schemes.

### State Transition Technique-

If there are different responses during transition from one state to another then state transition testing can be incorporated.

For example, if the user wants to login to any application but enters the password wrong, after a particular number of attempts the account is locked. Therefore, there is a transition from an available state to a locked account.

### **Error Guessing Technique-**

Error Guessing is a technique based on guessing the errors that can stay in the application. It can help to look for various faults which systematic techniques may fail to detect. The method almost always depends on the tester's prior experience and their foresight to guess where defects can occur in the future.

### **All-pair Testing Technique-**

All-pair testing Technique is used to test all the possible discrete combinations of values. This combinational method is used for testing the application that uses checkbox input, radio button input, list box, text box, etc.

### **Advantages-**

- It's one of the best options for functional testing.
- Test cases can be created fast with the help of black box testing's
- Helps in easy implementation of the application.
- The tester does not need knowledge of programming language or system implementation.
- The tester and the designer do not work together hence the testing is unbiased.
- Can be implemented by testers without technical knowledge.

### **Disadvantages-**

- If the number of test inputs is more, then it can be time-consuming and exhausting.
- Achieving no result at the end of a test is a big probability.

- The whole software is not tested as it can be time-consuming.
- Black box testing's depend on specifications hence building test cases without specifications becomes very difficult.
- Cannot be used for complicated software.

#### NOTE-

#### Which errors are not identified by Black Box Testing?

The errors that are not identified by black box testing are: missing functions, incorrect functions, data structure errors, behavior errors, interface errors, performance errors, initialization and termination errors.

#### Now, Understand Types of Functional Testing-

- **Unit Testing-**  
**Unit testing** is a type of software testing, where the individual unit or component of the software tested. Unit testing, examine the different part of the application, by unit testing functional testing also done, because unit testing ensures each module is working correctly.

The developer does unit testing. Unit testing is done in the development phase of the application.

- **Smoke Testing-**  
**Smoke testing**, also called **build verification testing** or confidence testing, is a software testing method that is used to determine if a new software build is ready for the next testing phase.

- **Sanity Testing-**

**Sanity testing** involves the entire high-level business scenario is working correctly. Sanity testing is done to check the functionality/bugs fixed. Sanity testing is little advance than smoke testing.

- **Regression Testing-**

This type of testing concentrates to make sure that the code changes should not side effect the existing functionality of the system. Regression testing specifies when bug arises in the system after fixing the bug, regression testing concentrate on that all parts are working or not. Regression testing focuses on is there any impact on the system.

- **Integration Testing-**

**Integration testing** combined individual units and tested as a group. The purpose of this testing is to expose the faults in the interaction between the integrated units.

- **User Acceptance Testing-**

It is a type of testing performed by the **client** to certify the system according to requirement. The final phase of testing is user acceptance testing before releasing the software to the market or production environment.

- **Retesting-**

To ensure that the defects which were found and posted in the earlier build were fixed or not in the current build.

**Retesting** is running the previously failed test cases again on the new software to verify whether the defects posted earlier are **fixed or not**.

- **Ad-hoc Testing-**

**Ad-hoc testing** is an informal testing type whose aim is to break the system. This type of software testing is unplanned activity. It does not follow any test design to create the test cases. Ad-hoc testing is done randomly on any part of the application; it does not support any structured way of testing. It is also known as **Monkey or Random Testing**.



- Recovery Testing-

**Recovery testing** is used to define how well an application can recover from crashes, hardware failure, and other problems. The purpose of recovery testing is to verify the system's ability to recover from testing points of failure.

Recovery testing is a type of **non-functional testing** technique performed in order to determine how quickly the system can recover after it has gone through system crash or hardware failure.

- Static Testing/Verification Testing-

**Static/Verification testing** is a software testing technique by which we can check the defects in software without actually executing it. Static testing is done to avoid errors in the early stage of the development as it is easier to find failure in the early stages. Static testing used to detect the mistakes that may not found in **dynamic/validation testing**.

- Dynamic/Validation Testing-

**Dynamic/Validation testing** refers to analyzing code's dynamic behavior in the software. In this type of testing, you have to give input and get output as per the expectation through executing a test case.

- Component Testing-

**Component Testing** is also a type of software testing in which testing is performed on each component separately without integrating with other parts. Component testing also referred to as **Unit testing**, program testing, or module testing.

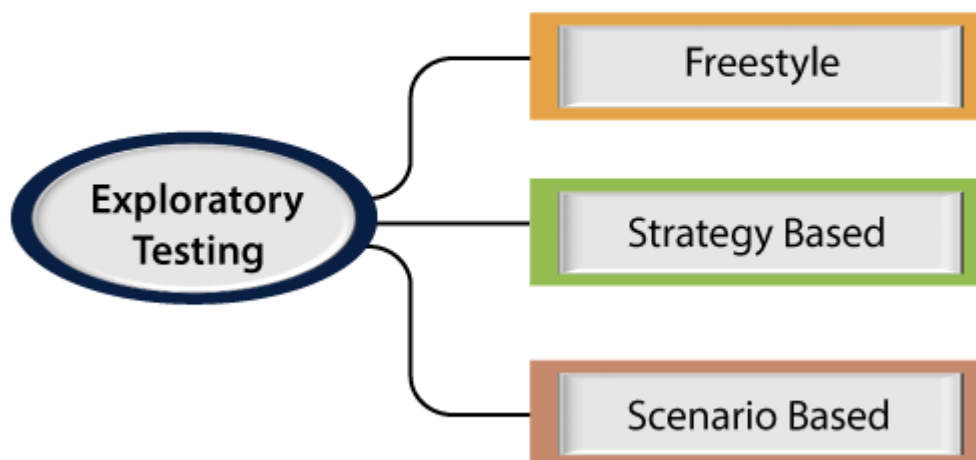
## Exploratory Testing

If requirement is not clear or does not exist, then we do one round of exploratory testing.

So, for this first, we will be exploring the application in all possible ways, understanding the flow of the application, preparing a test document and then testing the application, this approach is known as **exploratory testing**.



## Types of Exploratory Testing



**Freestyle-**

We did not follow any rules, there is no maximum coverage, and we will explore the application just like Ad-hoc testing.

**Strategy-**

Strategy based exploratory testing can be performed with the help of multiple testing techniques such as risk-based.

It is done by the **experienced tester** and who is using the application for the longest time because he/she is known the application very much.

**Scenario-**

Scenario-based exploratory testing is performed with the help of multiple scenarios such as end-to-end, test scenarios, and real user scenarios.

UNCODEMY

## Visual Testing

**Visual testing** is used to examine what happened at the point of software failure by defining the data in such a way that the developer can quickly identify the reason of failure, and the information is expressed clearly so that any other developer can utilize this information.

**Visual testing** requires the video recording of the entire process. It captures everything that happens at the time of system testing in video format. Tester gives a picture in a picture webcam and audio commentary from microphones as an input value.

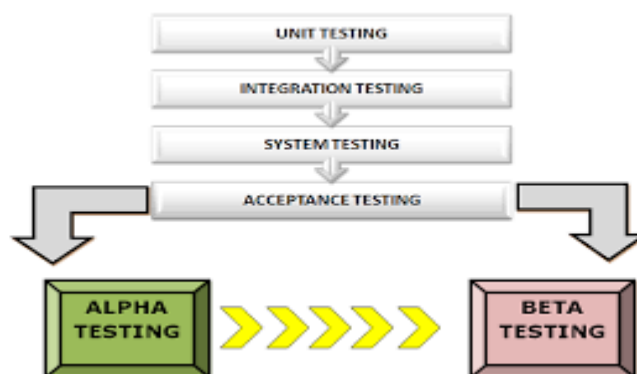
### Why Visual Testing is important?

Visual testing is essential to ensuring that the UI of an application or website meets the expected design and usability standards. Any visual defects or inconsistencies can impact the user experience and lead to user frustration, reduced productivity, or even abandonment of the application.

## Alpha Testing

Alpha Testing is a type of **Acceptance Testing** performed by the testers who are part of the organization, in other words: internal employees. It is the final stage of testing and it is usually done to verify that an application is free of errors / bugs before being launched on the market.

This test uses black box and white box techniques and it is performed near the end of the development of the software and before Beta Testing.



## Beta Testing

Beta Testing is performed by **real users** and it is unstructured. It can be considered as a form of external User Acceptance Testing.

Users can freely use the application and then they are encouraged to give feedback about their experience. This test is more focused on performance and scalability.



## Globalization Testing

Globalization testing is another type of software testing which is used to test the software that is developed for multiple languages, is called globalization testing, and improving the application or software for various languages is known as globalization.

This testing ensures that the application will support multiple languages and multiple features because, in current scenarios, we can see the enhancement in several technologies as the applications are planned in such a way that it is used globally.

For Example-

**Google.com** supports most of the languages,



### Reason Behind Globalization Testing-

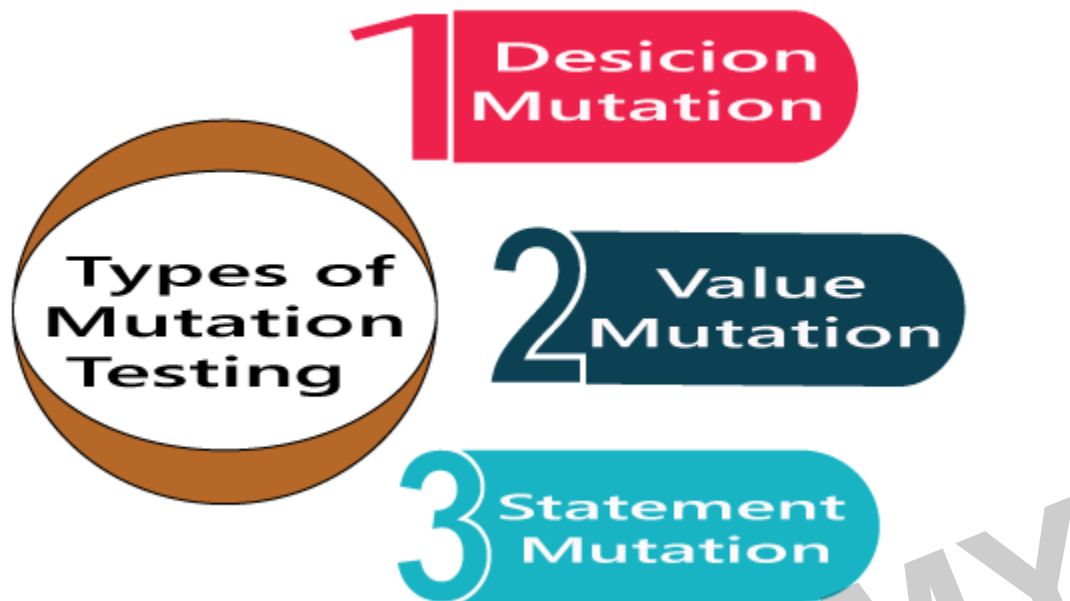
- It is used to make sure that the application is to support all the languages around the world.
- It is used for the identification of the various phases of the implementation.
- This testing will focus on the world-wide experiences of the application.

### Mutation Testing

Mutation testing is a white box method in software testing where we insert errors purposely into a program (under test) to verify whether the existing test case can detect the error or not. In this testing, the mutant of the program is created by making some modifications to the original program.

The primary objective of mutation testing is to check whether each mutant created an output, which means that it is different from the output of the original program. We will make slight modifications in the mutant program because if we change it on a massive scale than it will affect the overall plan.

### Three types of Mutation Testing-



#### 1. Decision Mutation-

We check the design errors. we will do the modification in arithmetic and logical operator to detect the errors in the program.

plus (+) → minus (-)

asterisk (\*) → double asterisk (\*\*)

Original Code	Modified Code
if(p > q) r = 5; else r = 15;	if(p < q) r = 5; else r = 15;

#### 2. Value mutation

The values will modify to identify the errors in the program.

Original Code	Modified Code
<pre>int add =9000008; int p = 65432; int q =12345; int r = (p+ q);</pre>	<pre>int mod = 9008; int p = 65432; int q =12345; int r = (p + q);</pre>

### 3. Statement mutation

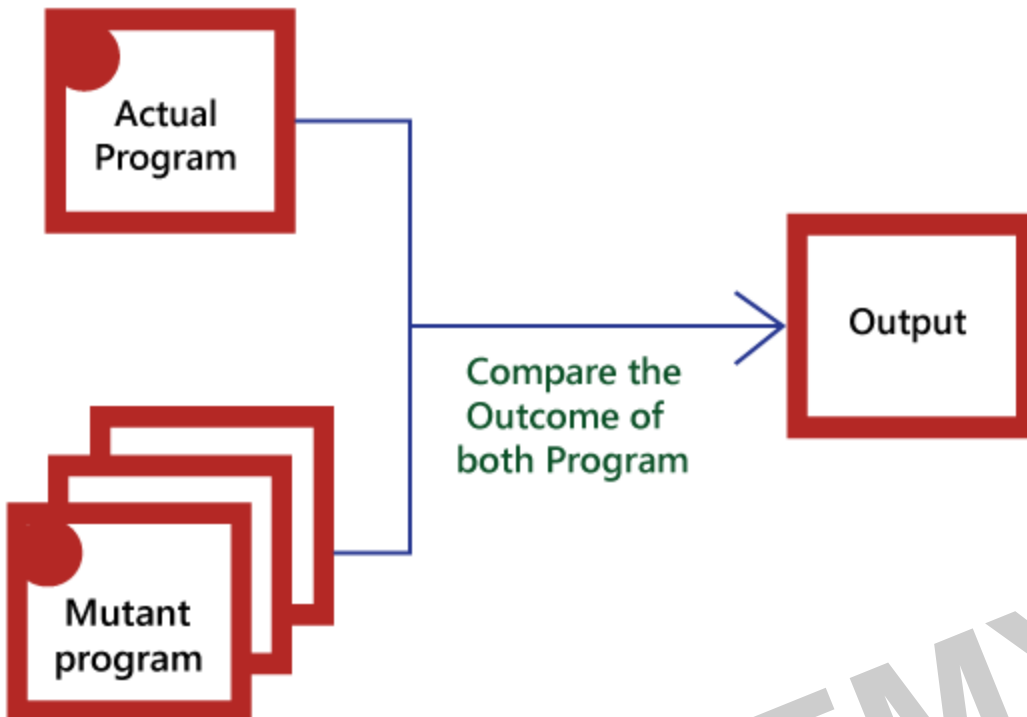
We can do the modifications into the statements by removing or replacing the line.

Original Code	Modified Code
<pre>if(p * q) r = 15; else r = 25;</pre>	<pre>if(p* q) s = 15; else s = 25;</pre>

### How to perform Mutation Testing-

- We will add the errors into the source code of the program by producing various versions, which are known mutants.
- After that we will take the help of the test cases in the mutant program and the actual application will find the errors in the code.
- Once we identify the faults, we will match the output of the actual code and mutant code.
- After comparing the output of both actual and mutant programs, if the results are not matched, then the mutant is executed by the test cases. Therefore the test case has to be sufficient for identifying the modification between the actual program and the mutant program.





### Advantages-

- This technique can identify all the errors in the program and also helps us to discover the doubts in the code.
- It is a right approach for error detection to the application programmer

### Disadvantages-

- This testing is a bit of time taking and costlier process.
- As it is a tedious process, so we can say that this testing requires the automation tools to test the application.

### Accessibility testing

Accessibility testing is the practice of making your web and mobile apps usable to as many people as possible. It makes apps accessible to those with disabilities, such as vision impairment, hearing disabilities, and other physical conditions.

Here the physical disability could be old age, hearing, color blindness, and other underprivileged groups. It is also known as 508 compliance testing. In this, we will test a web application to ensure that every user can access the website.

### Testcase for Accessibility Testing-

- Labels — Used by assistive technologies, like VoiceOver.
- Text contrast — Ratio between text or images and background color.
- Hit area size — Area designated for user interaction.
- View hierarchy of UI — Determines how easy the Android app is to navigate.
- Dynamic font size — Option for users to increase font size to fit their needs.

### Myths-

Accessibility testing is only for physically challenged people.

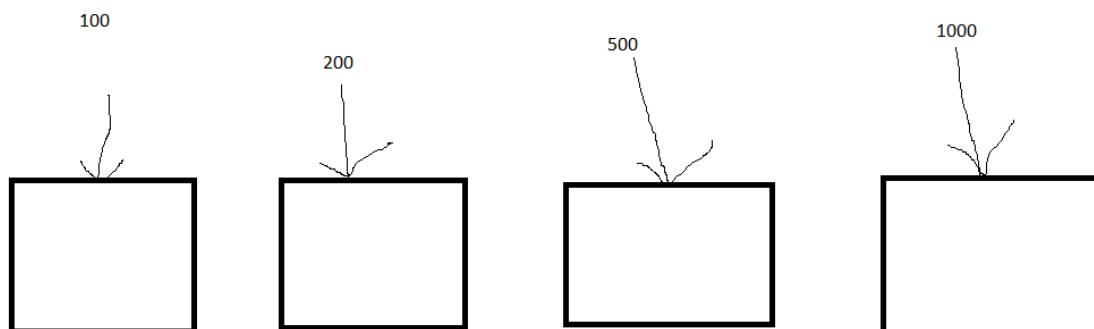
### Facts-

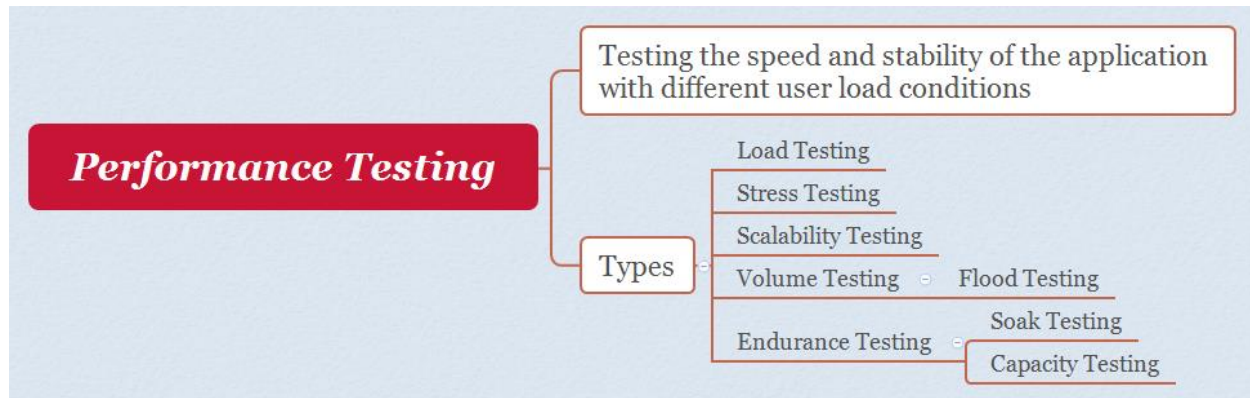
All types of users can use the accessibility testing as they enhance the credibility of software.

## Non-Functional Testing

- **Performance Testing-**

Testing **Speed** and **Stability** of the application under different **user load** conditions is called Performance testing.





### 1. Load Testing-

Load testing is a type of testing where you test the expected load means expected load is met by our application or not.

E.g- if in your application, user handle limit will be 1000 according to the requirement, then in load testing you have to test that your application is handled user less-than or equal to 1000( $\leq 1000$ ) or not, we will not check user more than 1000.

### 2. Stress Testing-

Stress testing is a type of performance testing in which a system is subjected to a workload beyond the limits of its normal operations to determine how it behaves under such scenarios. The primary objective of stress testing is to identify bottlenecks, evaluate system response times, and detect any flaws or failures when the system is under stress.

E.g.- if in your application according to the requirement the maximum load is 1000 user at a time but if suddenly 2000 users will login in your application, then at this condition or stress how your application is handling that user. Either your application is crashed or not behaving properly/facing issues in stability of your application.

#### Aim/Benefits of Doing Stress Testing is-

- It can identify areas of a system that might not be able to handle extreme conditions, such as high traffic or a downturn in the market. This might result in system failures.

- The objective of a stress test is to identify and address system weaknesses to improve the overall resiliency of a system, making the system more capable of coping with unexpected events.

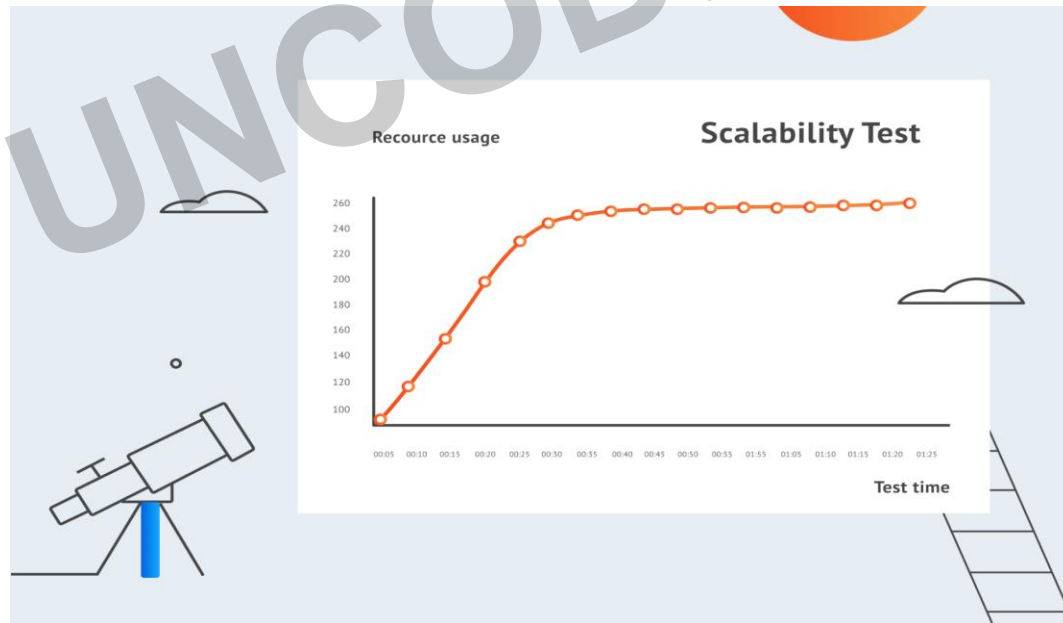
#### Real world Examples-

- During mega sales, eCommerce websites would see a significant increase in traffic.
- When university websites publish exam results, multiple users will check the results simultaneously.

### 3. Scalability Testing

A scalability test is a type of load testing that measures the application's ability to scale up or down as a reaction to an increase in the number of users.

In this testing we increase user one by one until unless the application is crashed. To check at which point/at which user load point the application is crashed or not working properly this is called scalability testing.



#### 4. Volume Testing

**Volume testing** is a type of testing where we test the data of the application not the user load.

It helps us to check the behavior of an application by inserting a **massive volume** of the load in terms of **data** is known as volume testing. It is also known as **Flood testing**.

E.g. Whenever we purchase any new mobile in the beginning the speed of the mobile is very fast because they have limited data once we start inserted data into the mobile then day by day the speed is decreasing. So, in this type of situation we have to check that at what is the limit of the data we have to insert in the application in which our application remains/constantly provides the same speed. As he provides in the beginning. This testing is called volume testing.

#### 5. Endurance Testing

Endurance testing which is done to check if the software system can sustain under a huge expected load continued over a long period of time. It is a subset of Load Testing.

Generally, it is also known as **Soak testing** and **longevity testing**.

E.g. Endurance testing is also same as Load testing the only difference is in Load testing we test the expected load is matched in the application or not. But, in Endurance testing we test the expected load as well as test the how much time that load should be sustain by are application this is called endurance testing.

## 6. Spike Testing-

Spike testing is a type of performance testing in which an application receives a sudden and extreme increase or decrease in load. The goal of spike testing is to determine the behavior of a software application when it receives extreme variations in traffic.



E.g. Take a website like Flipkart in general flipkart have 100000(1 lakh users) in daily basis. But suddenly users are increases to 300000(3 lakh user) then how the application is handled that users. Either is application is crashing or not at this point label. This type of testing is called spike testing.

### Spike Scenarios-

- When a web application is live streaming a favorite TV program. E.g.- Live match on Hotstar.
- A new system is released for production, and multiple users want to access the system.
- When a flash sale is going on a daily deal site.

## Security Testing

- **What is Security Testing?**

Security Testing is a type of Non-Functional Software Testing.

The purpose of Security Testing is to identify/uncover any Security Loopholes (i.e., Vulnerabilities or Security Bugs) in the given Software System (i.e., Applications, Databases, Machines, Operating Systems, Organizations etc.) and get them fixed, with an intention of protecting the Information in these Systems from Hackers.

Security Testing finds-out (i.e., Assess) whether the data and resources related to the given Software System is protected from possible hacking/breaches/intrusions.

Let's understand this with the help of an example:

**Example:** Performing Security Testing on the Login functionality of a Web Application.

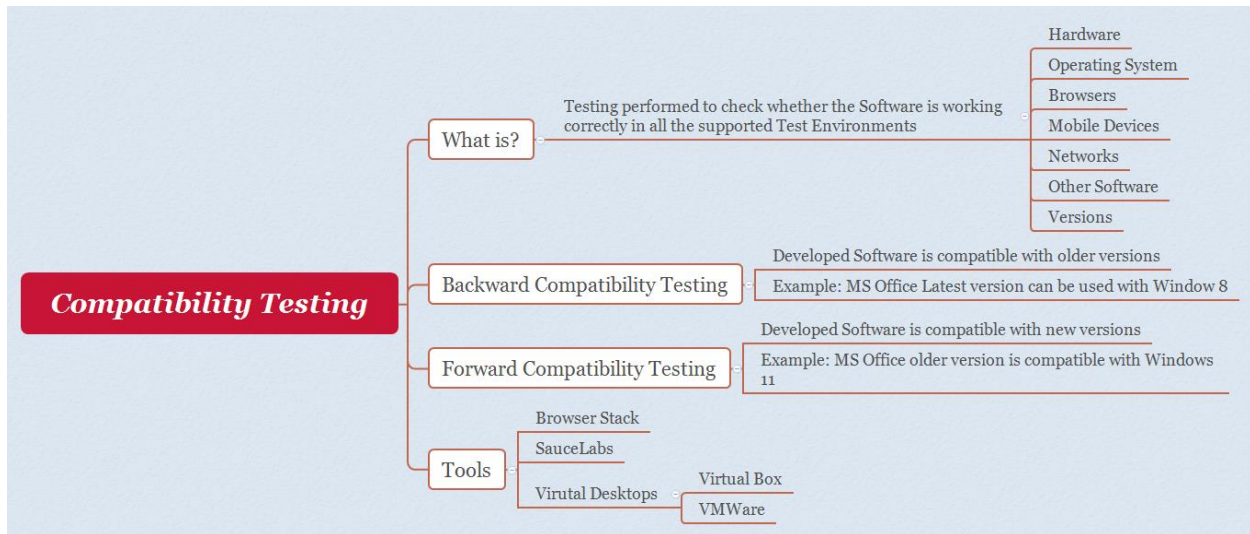
As a Security Tester, we investigate the Login functionality of the given Web Application:

Identify any security loopholes (i.e. Vulnerabilities) in the Login Functionality implementation in the given Web Application, using which the Hacker can login to the Web Application without having any login credentials.

If the Hacker is able to hack the Login functionality and intrude inside any User account on the Software System, the data (May be sensitive like a credit card information etc.) of the User can be compromised and then later be used for purchases by the hacker.

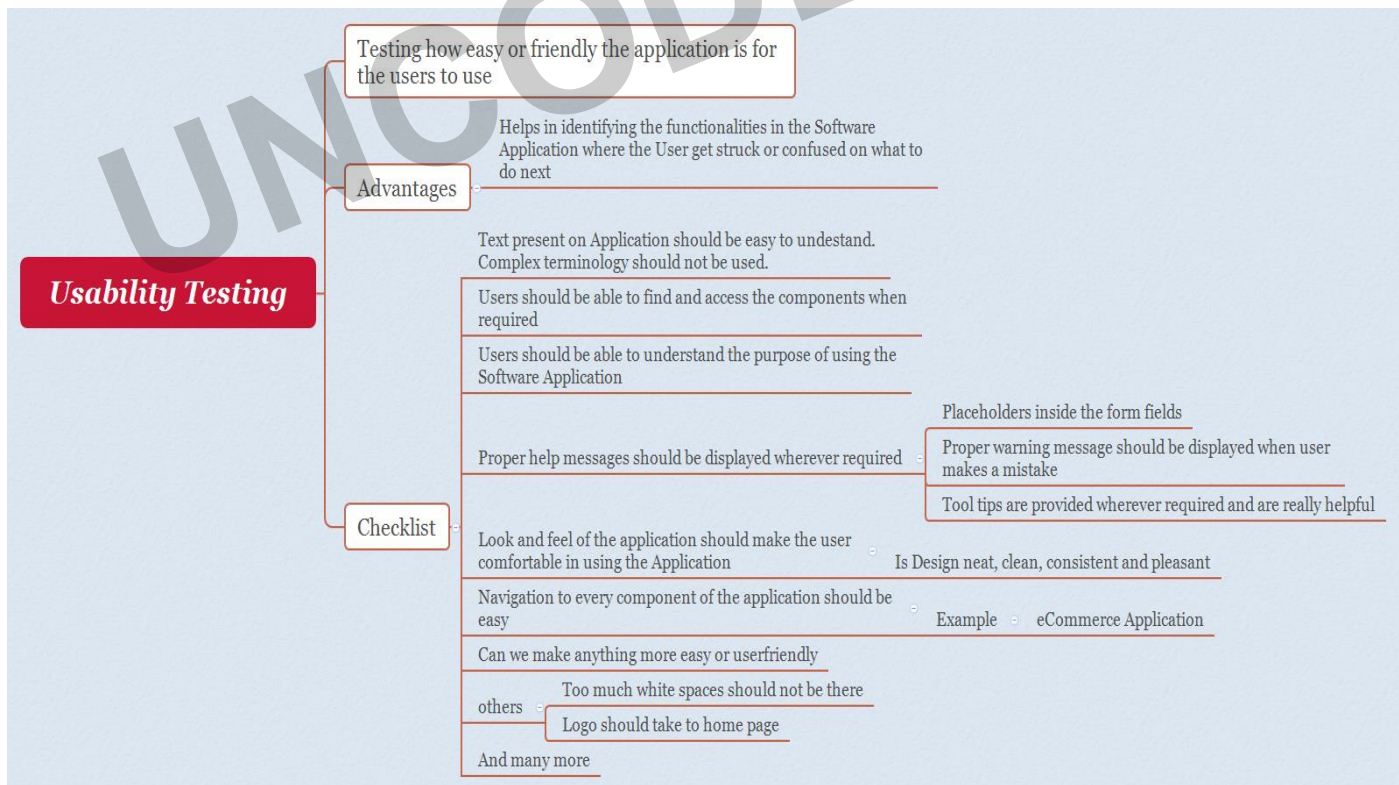
## Compatibility Testing

A compatibility test is an assessment used to ensure a software application is properly working across different browsers, databases, operating systems (OS), mobile devices, networks and hardware.



## Usability Testing

Usability testing is instead focused on the **end-user**, on how easily he was able to use the interface and if the design of the interface was **friendly** enough. Going back to the e-commerce shoes example, a typical question could be: Do users easily find the “buy now” button?





## Reliability testing

Reliability Testing is an important software testing technique that is performed by the team to ensure that the software is performing and functioning consistently in each environmental condition as well as in a specified period. It ensures that product is fault free and is reliable for its intended purpose.

E.g.- Testers and developers test the reliability of an app/system as per several factors. For example, if the app can handle heavy user traffic and requests for several hours without crashing down or not.

In simple words, we can say that it verifies whether the software can achieve a failure-free operation for a specific period in a technological environment or not.

### Types of Reliability Testing-

- **Feature Testing**
- **Regression Testing**
- **Load Testing**

- **Feature Testing**

Feature testing is the software development process of testing multiple variations of a feature to determine the best user experience. With feature tests, you can validate whether a new feature for web page or app is a good fit.

E.g. Apple and OnePlus keep on adding new features to their products to enhance the quality and to make customers keen to know the new add-on. Customers keep an eye on the new features to be launched, and the attractive and effective features let them buy the same as well.

- **Regression Testing**

Regression testing is a software testing practice that ensures an application still functions as expected after any code changes, updates, or improvements. Regression testing is responsible for the overall stability and functionality of the existing features.

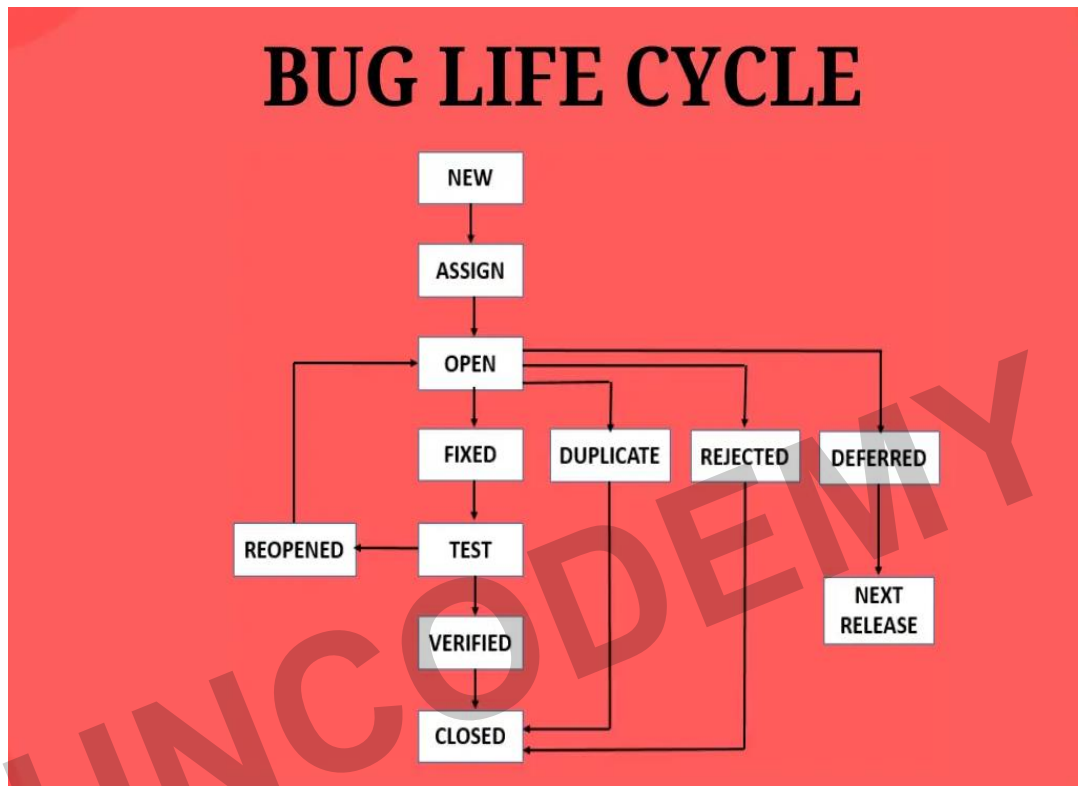
- **Load Testing**

Load testing is a type of testing where you test the expected load means expected load is met by our application or not.

UNCODEMY

## Defect Tracking/Bug Reporting

### ⇒ Bug Life Cycle



**Bug life cycle** is a cyclic process, which describes how a defect or bug passes through different stages from the identification stage to the Fixing stage.

**E.g.** it begins when a tester finds or logs a bug and it ends when the bug is fixed.

#### Steps-

- As soon as the test engineer finds the bug, status is given as **New**, which indicates that a bug is just found.
- This new bug needs to be reported to the concerned Developer by changing the status as **Assigned** so that the responsible person should take care of the bug.
- Then the Developer first go through the bug, which means that the Developers read all the navigation steps to decide whether it is a valid **bug** or **not (Duplicate, Rejected or Deferred)**.

- Based on this, if the bug is valid, the Developer starts reproducing the bug on the application, once the bug is successfully reproduced, the Developer will analyze the code and does the necessary changes, and change the status as **Fixed**.
- Once the code changes are done, and the bug is fixed, the test engineer **re-test** the bug, which means that the test engineer performs the same action once again, which is mentioned in the bug report, and changes the status accordingly:
- **Close**, if the bug fixes properly, and functionally working according to the requirement.

OR

- **Re-open**, if the bug still exists or not working properly as per the requirement, then the bug sends it back to the Developer once again.

This process is going on continuously until all the bugs are fixed and closed.

#### Note-

The test engineer cannot tell the bug orally to the Developer due to some reason:

- Developers might ignore the bug.
- Forget the bug.
- The bug may not be found in the exact location.

The **Bug can be assigned** to following:

- **Developers**- If we know who has developed that particular module.
- **Developers lead/ Project manager**- If we don't know the Developer who has developed the particular module.
- **Test lead/QA manager**- When we don't have any interaction with the development team.

#### Invalid / Rejected Bug-

When the Test Engineer wrote an incorrect Bug Report because of misunderstanding the requirements, then the Developer will not accept the bug, and gave the status as Invalid and sent it back. (Sometime Developer can also misunderstand the requirements).

“Any bug which is not accepted by the developer is known as an invalid bug.”

## ⇒ Severity and Priority

### Severity-

The impact of the bug on the application is known as severity/Severity is a term that denotes how severely a defect can affect the functionality of the software.

It can be a **blocker**, **critical**, **major**, and **minor** for the bug.

- ⇒ **Blocker**: if the severity of a bug is a blocker, which means we cannot proceed to the next module, and unnecessarily test engineer sits ideal.
- ⇒ **Critical**: if it is critical, that means the main functionality is not working, and the test engineer cannot continue testing.
- ⇒ **Major**: if it is major, which means that the supporting components and modules are not working fine, but test engineer can continue the testing.
- ⇒ **Minor**: if the severity of a bug is minor, which means that all the U.I problems are not working fine, but testing can be processed without interruption.

### Priority-

Priority is important for fixing the bug or which bug to be fixed first or how soon the bug should be fixed.

It can be **high**, **medium**, and **low**.

- ⇒ **High**: it is a major impact on the customer application, and it has to be fixed first.
- ⇒ **Medium**: In this, the problem should be fixed before the release of the current version in development.
- ⇒ **Low**: The flow should be fixed if there is time, but it can be deferred with the next release.

### Note-

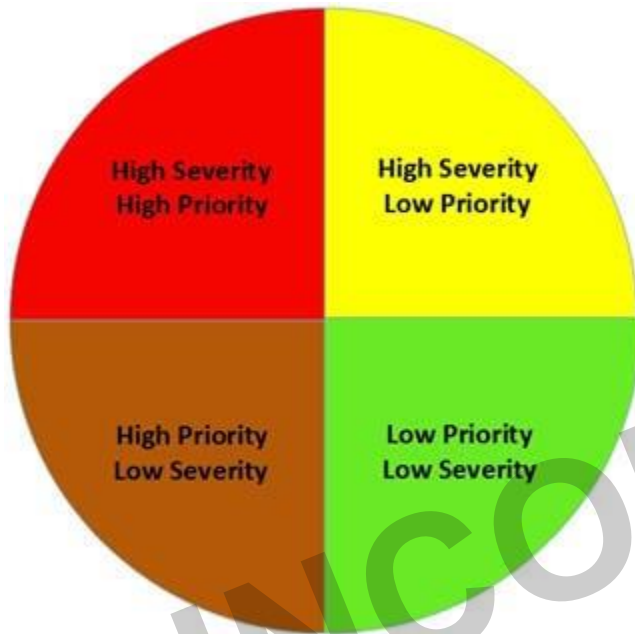
The test engineer decides the severity and Priority (Sometimes), and the developer can also change the severity with a proper reason and comments on the bug reports.

A developer cannot change the Priority, because if the developer changes the Priority, he/she may fix the easy bug's first.

### Different Levels-

Let's take a look at the different levels for both Priority and Severity-

- ⇒ High Priority, High Severity
- ⇒ High Priority, Low Severity
- ⇒ High Severity, Low Priority
- ⇒ Low Severity, Low Priority



- ⇒ **High Severity and High Priority-**

Any Critical/major business case failure automatically gets promoted to this category.

Any defects due to which the testing cannot continue at any cost or causes a severe system failure to fall into this category.

For **Example-**

The system crashes after you made the payment or when you are not able to add the items to the Cart, this defect is marked as High Severity and High Priority defect.

⇒ **High Priority and Low Severity-**

Any minor severity defects that could directly impact the user experience automatically gets promoted to this category.

**For Example-**

The logo of the company in the front-page is wrong, it is considered to be High Priority and Low Severity defect.

In the bank logo, instead of ICICI, it is written as ICCCI.

⇒ **High Severity and Low Priority**

An error which occurs on the functionality of the application (for which there is no workaround) and will not allow the user to use the system but on click of link which is rarely used by the end user.

**For Example-**

Informative pages such as the disclaimer and **privacy policy** are crashing, which may impact other major system functionality. The bug's severity is high, but as many users do not frequently access these pages, it is marked as a low priority by the project manager and business team.

⇒ **Low Severity and Low Priority-**

Any spelling mistakes /font casing/ misalignment in the paragraph of the 3rd or 4th page of the application and not in the main or front page/ title.

**For Example,**

If the privacy policy of the website has a spelling mistake, this defect is set as Low Severity and Low Priority.

UNCODEMY