SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

Re-accredited by NAAC with A+ grade, Accredited by NBA(CSE, IT, ECE, EEE & Mechanical)
Approvedy by AICTE, New Delhi, Recognized by UGC, Affiliated to Anna University, Chennai

Department of MCA

Topic: Using Black Box Approaches to Test Case Design

COURSE	UNIT - II	CLASS
19CAE712 STQA	TESTING TECHNIQUES AND LEVELS OF TESTING	III Semester / II MCA

Black Box Testing

- What is Black Box Testing?
- Types of Black Box Testing
  - #1) Functional Testing
  - #2) Non-Functional Testing
- Black Box Testing Tools
- Black Box Testing Techniques
  - #1) Equivalence Partitioning
  - #2) Boundary Value Analysis
  - #3) Decision Table Testing
  - #4) State Transition Testing
  - #5) Error Guessing

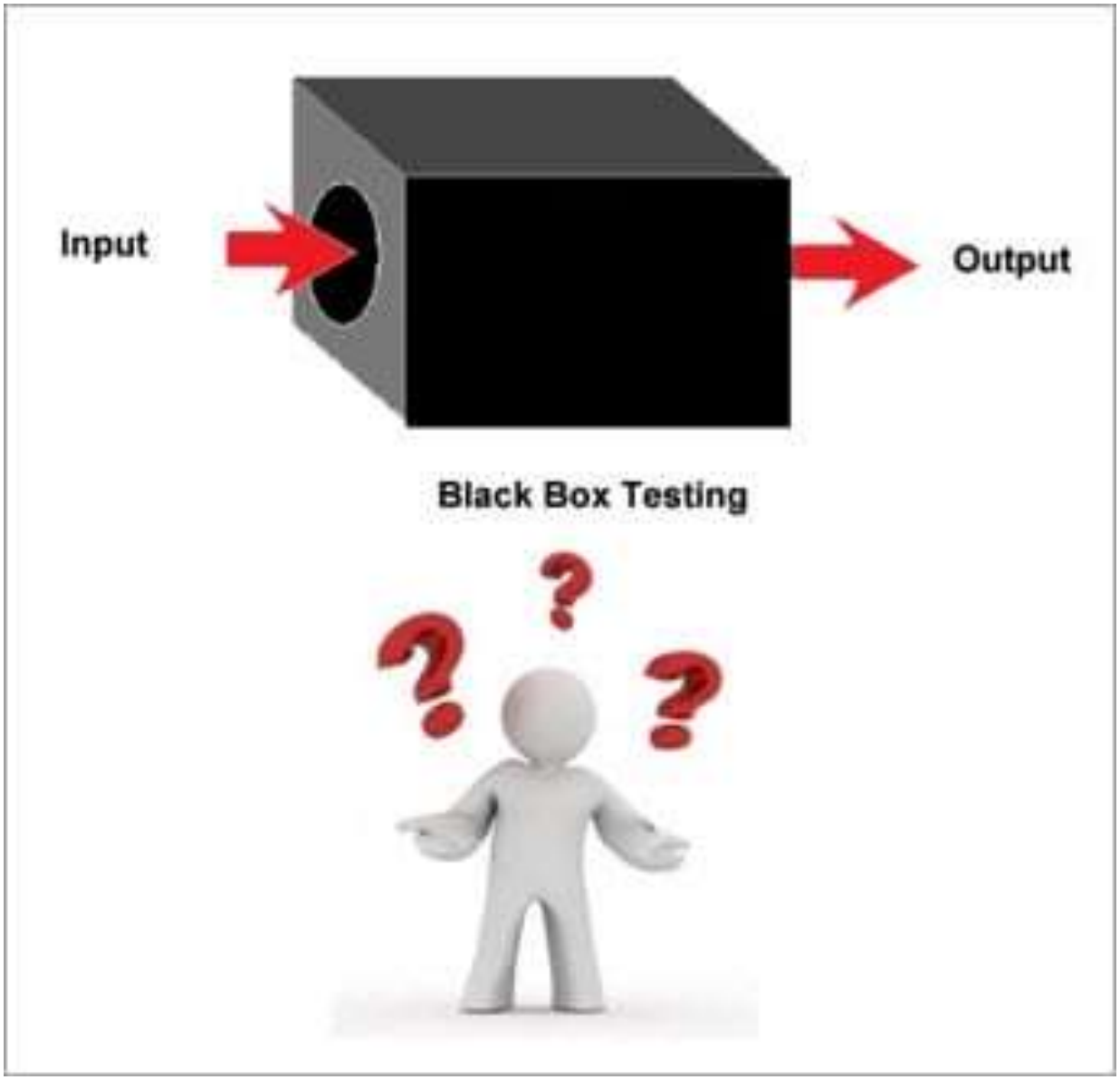
What is Black Box Testing?

- Black Box Testing is also known as behavioral, opaque-box, closed-box, specification-based or eye-to-eye testing.
- It is a Software Testing method that analyzes the functionality of a software/application without knowing much about the internal structure/design of the item that is being tested and compares the input value with the output value.

Black Box Testing

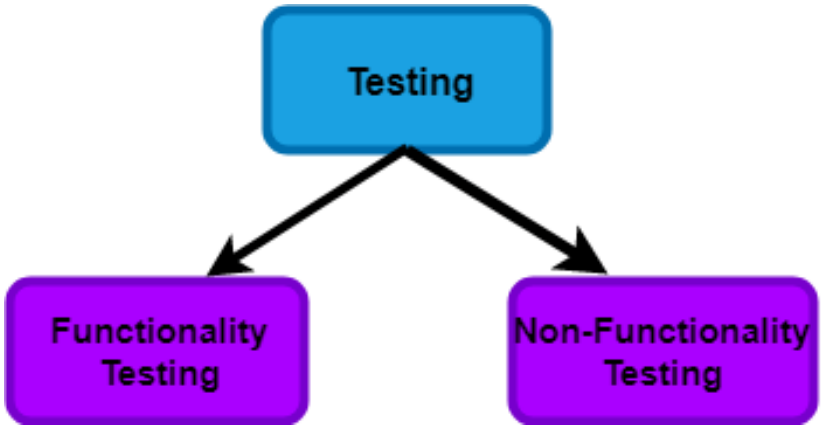
- **The main focus of Black Box Testing is on the functionality of the system as a whole.** The term **'Behavioral Testing'** is also used for Black Box Testing.
- A majority of the applications are tested using the Black Box method. We need to cover the majority of test cases so that most of the bugs will get discovered by the Black-Box method.
- This testing occurs throughout the Software Development and Testing Life Cycle i.e in Unit, Integration, System, Acceptance, and Regression Testing stages.
- This can be either Functional or Non-Functional.

Black Box Testing



Types of Black Box Testing

- #1) Functional Testing
- This testing type deals with the functional requirements or specifications of an application. Here, different actions or functions of the system are being tested by providing the input and comparing the actual output with the expected output.
- **For example**, when we test a Dropdown list, we click on it and verify if it expands and all the expected values are showing in the list.
- **Few major types of Functional Testing are:**
- Smoke Testing
- Sanity Testing
- Integration Testing
- System Testing
- Regression Testing
- User Acceptance Testing



Types of Black Box Testing

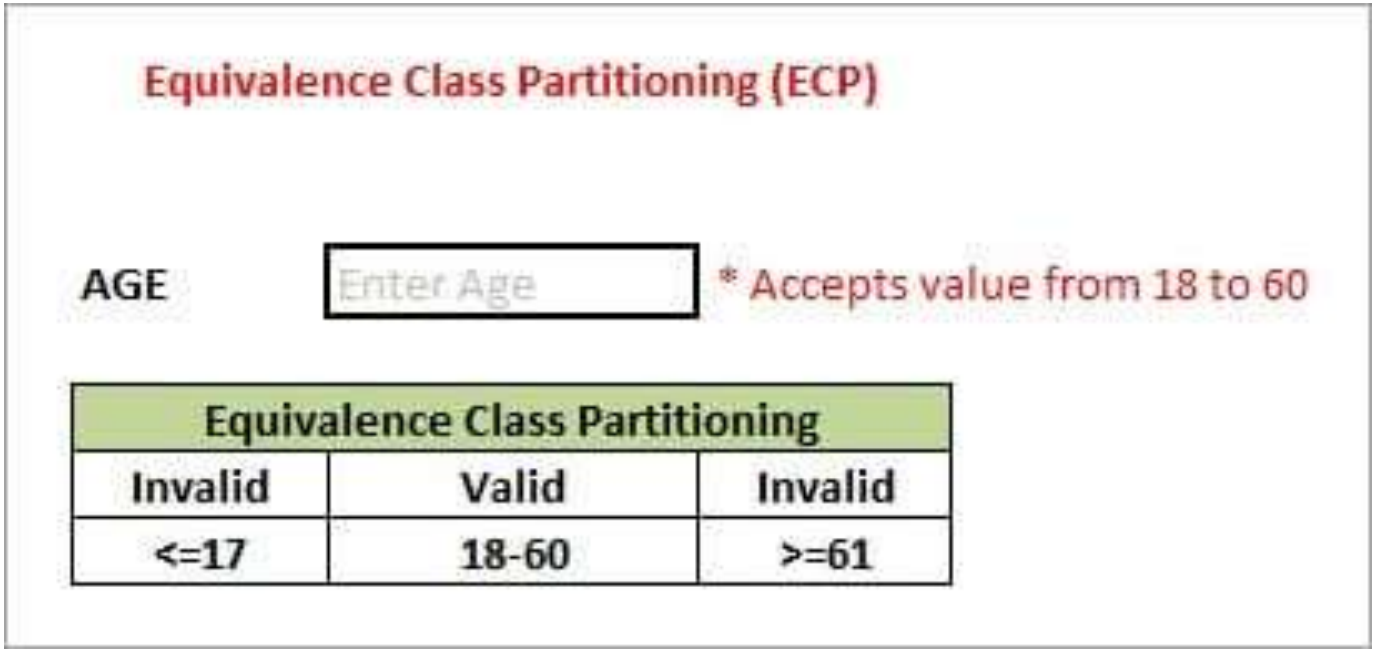
- #2) Non-Functional Testing
- Apart from the functionalities of the requirements, there are even several non-functional aspects that are required to be tested to improve the quality and performance of the application.
- **Few major types of Non-Functional Testing include:**
- Usability Testing
- Load Testing
- Performance Testing
- Compatibility Testing
- Stress Testing
- Scalability Testing

Black Box Testing Tools

- Black Box Testing tools are mainly record and playback tools. These tools are used for Regression Testing to check whether a new build has created any bugs in the previous working application functionality.
- These record and playback tools record test cases in the form of scripts like TSL, VB script, Javascript, Perl, etc.
- **Black Box Testing Techniques**
- In order to systematically test a set of functions, it is necessary to design test cases. Testers can create test cases from the requirement specification document using the following Black Box Testing techniques:
  - **Equivalence Partitioning**
  - **Boundary Value Analysis**
  - **Decision Table Testing**
  - **State Transition Testing**
  - **Error Guessing**
  - **Graph-Based Testing Methods**
  - **Comparison Testing**

Black Box Testing Tools

- #1) Equivalence Partitioning
- This technique is also known as Equivalence Class Partitioning (ECP). In this technique, input values to the system or application are divided into different classes or groups based on its similarity in the outcome.
- Hence, instead of using each and every input value, we can now use any one value from the group/class to test the outcome. This way, we can maintain test coverage while we can reduce the amount of rework and most importantly the time spent.



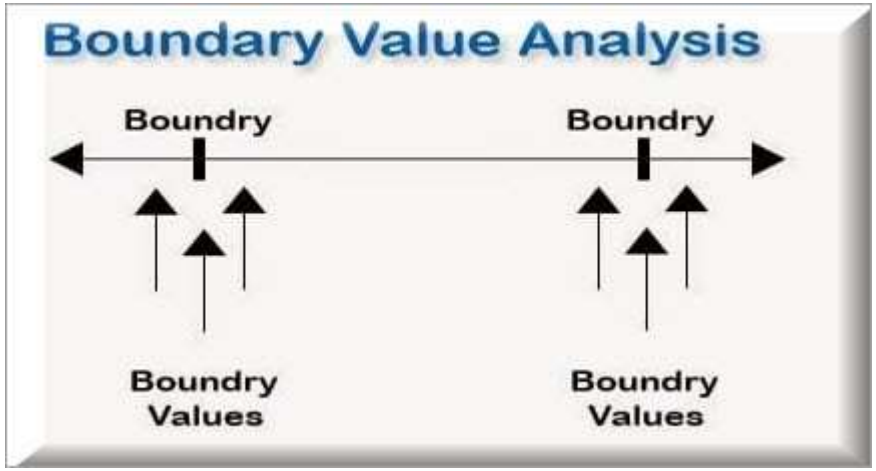
Equivalence Class Partitioning		
Invalid	Valid	Invalid
<=17	18-60	>=61

Black Box Testing Tools

- **Two invalid classes will be:**
- a) Less than or equal to 17.
- b) Greater than or equal to 61.
- A valid class will be anything between 18 and 60.
- We have thus reduced the test cases to only 3 test cases based on the formed classes thereby covering all the possibilities. So, testing with any one value from each set of the class is sufficient to test the above scenario.

Black Box Testing Tools

- #2) Boundary Value Analysis
- The name itself defines that in this technique, we focus on the values at boundaries as it is found that many applications have a high amount of issues on the boundaries.
- Boundary refers to values near the limit where the behavior of the system changes. In boundary value analysis, both valid and invalid inputs are being tested to verify the issues.



- If we want to test a field where values from 1 to 100 should be accepted, then we choose the boundary values: 1-1, 1, 1+1, 100-1, 100, and 100+1. Instead of using all the values from 1 to 100, we just use 0, 1, 2, 99, 100, and 101.

Black Box Testing Tools

- #3) Decision Table Testing
- As the name itself suggests, wherever there are logical relationships like:
- *If*
  {
  (Condition = True)
  then action1 ;
  }
  else action2; /(condition = False)*/*
- Then a tester will identify two outputs (action1 and action2) for two conditions (True and False). So based on the probable scenarios a Decision table is carved to prepare a set of test cases.
- **For Example:**
- Take an example of XYZ bank that provides an interest rate for the Male senior citizen as 10% and 9% for the rest of the people.

Black Box Testing Tools

- #3) Decision Table Testing
- As the name itself suggests, wherever there are logical relationships like:
- *If*
  {
  (Condition = True)
  then action1 ;
  }
  else action2; /(condition = False)*/*
- Then a tester will identify two outputs (action1 and action2) for two conditions (True and False). So based on the probable scenarios a Decision table is carved to prepare a set of test cases.
- **For Example:**
- Take an example of XYZ bank that provides an interest rate for the Male senior citizen as 10% and 9% for the rest of the people.

Black Box Testing Tools

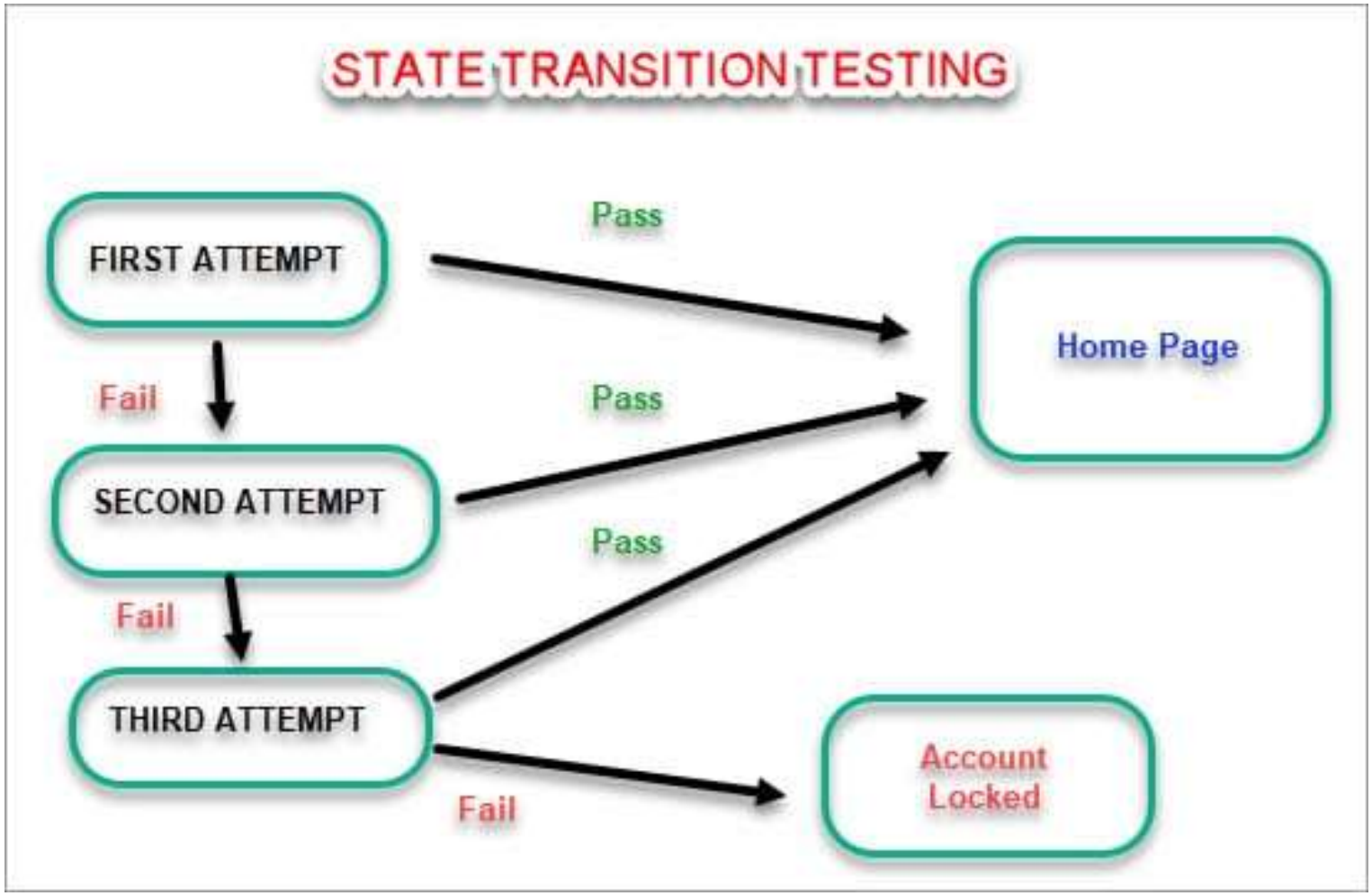
- In this example condition, C1 has two values as true and false, C2 also has two values as true and false. The total number of possible combinations would then be four. This way we can derive test cases using a decision table.

Decision Table / Cause-Effect				
Decision Table	**Rule 1**	**Rule 2**	**Rule 3**	**Rule 4**
Conditions				
C1 - Male	F	F	T	T
C2 - Senior Citizen	F	T	F	T
Actions				
A1 - Interest Rate 10%				X
A2 - Interest Rate 9%	X	X	X	

Black Box Testing Tools

- #4) State Transition Testing
- State Transition Testing is a technique that is used to test the different states of the system under test. The state of the system changes depending upon the conditions or events. The events trigger states which become scenarios and a tester needs to test them.
- A systematic state transition diagram gives a clear view of the state changes but it is effective for simpler applications. More complex projects may lead to more complex transition diagrams thereby making it less effective.

Black Box Testing Tools



Black Box Testing Tools

- #5) Error Guessing
- This is a classic example of Experience-Based Testing.
- In this technique, the tester can use his/her experience about the application behavior and functionalities to guess the error-prone areas. Many defects can be found using error guessing where most of the developers usually make mistakes.
- **Few common mistakes that developers usually forget to handle:**

- Divide by zero.
- Handling null values in text fields.
- Accepting the Submit button without any value.
- File upload without attachment.
- File upload with less than or more than the limit size.

Black Box Testing Tools

- **#6) Graph-Based Testing Methods**
- Each and every application is a build-up of some objects. All such objects are identified and the graph is prepared. From this object graph, each object relationship is identified and test cases are written accordingly to discover the errors.
- **#7) Comparison Testing**
- In this method, different independent versions of the same software are used to compare to each other for testing.

How do I do Step-wise?

- In general, when a systematic process is followed to test a project/application then quality is maintained and is useful in the long run for further rounds of testing.
- The foremost step is to understand the requirement specification of an application. Properly documented SRS (Software Requirement Specification) should be in place.
- Using the above mentioned Black Box Testing techniques such as Boundary Value Analysis, Equivalence partitioning etc, sets of valid and invalid inputs are identified with their desired outputs and test cases are designed based on that.
- The designed test cases are executed to check if they Pass or Fail by verifying the actual results with the expected results.
- Failed test cases are raised as Defects/Bugs and addressed to the development team to get it Fixed.
- Further, based on the defects being fixed, the tester retests the defects to verify if they are recurring or not.

Advantages and Disadvantages

- **Advantages**
- The tester does not need to have a technical background. It is important to test by being in the user's shoes and think from the user's point of view.
- Testing can start once the development of the project/application is done. Both the testers and developers work independently without interfering in each other's space.
- It is more effective for large and complex applications.
- Defects and inconsistencies can be identified in the early stages of testing.
- **Disadvantages**
- Without any technical or programming knowledge, there are chances of ignoring possible conditions of the scenario to be tested.
- In a stipulated time there is a possibility of testing less and skipping all possible inputs and their output testing.
- Complete Test Coverage is not possible for large and complex projects.

Difference Between White Box Testing and Black Box Testing

Black Box Testing	White Box Testing
It is a testing method without having knowledge about the actual code or internal structure of the application.	It is a testing method having knowledge about the actual code and internal structure of the application.
This is a higher level testing such as functional testing.	This type of testing is performed at a lower level of testing such as Unit Testing, Integration Testing.
It concentrates on the functionality of the system under test.	It concentrates on the actual code – program and its syntax's.
Black box testing requires Requirement specification to test.	White Box testing requires Design documents with data flow diagrams, flowcharts etc.
Black box testing is done by the testers.	White box testing is done by Developers or testers with programming knowledge.

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.
- **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.
- Developers and testers perform integration testing.
- **Stress Testing**, which is an important part of **Performance testing** and used to checks the behavior of an application by applying a load greater than the desired load.
- **Stress Testing** is testing used to check the accessibility and robustness of software beyond usual functional limits. It mainly considers for critical software but it can also be used for all types of software applications.
- It is also known as ***Endurance Testing, fatigue testing*** or ***Torture Testing***.

Functional Testing

- **load testing**, which is the important part of **Performance testing** and used to check the performance of the software by applying some load.
- Load testing is testing where we check an application's performance by applying some load, which is either less than or equal to the desired load.
- Here, load means that when **N-number** of users using the application simultaneously or sending the request to the server at a time.
- Load testing will help to detect the maximum operating capacity of an application and any **blockages** or bottlenecks.
- It governs how the software application performs while being accessed by several users at the same time.
- The load testing is mainly used to test the **Client/Server's performance and applications that are web-based.**

Functional Testing

- **scalability testing**, which checks the performance of an application by increasing or decreasing the load in particular scales like number of a user.
- It is used to check an application's performance by increasing or decreasing the load in particular scales known as **scalability testing**. It is executed at a **hardware, software, or database level**.
- It is specified as the capacity of a **network, system, application, product, or process** to make the function correctly when modifications are made in the system's size or volume to meet an increasing need.
- In this testing, the **Test Cases** are designed and implemented in a well-organized manner. It also analysis the **system, processes, or database's ability** to meet an upward need.
- **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**.

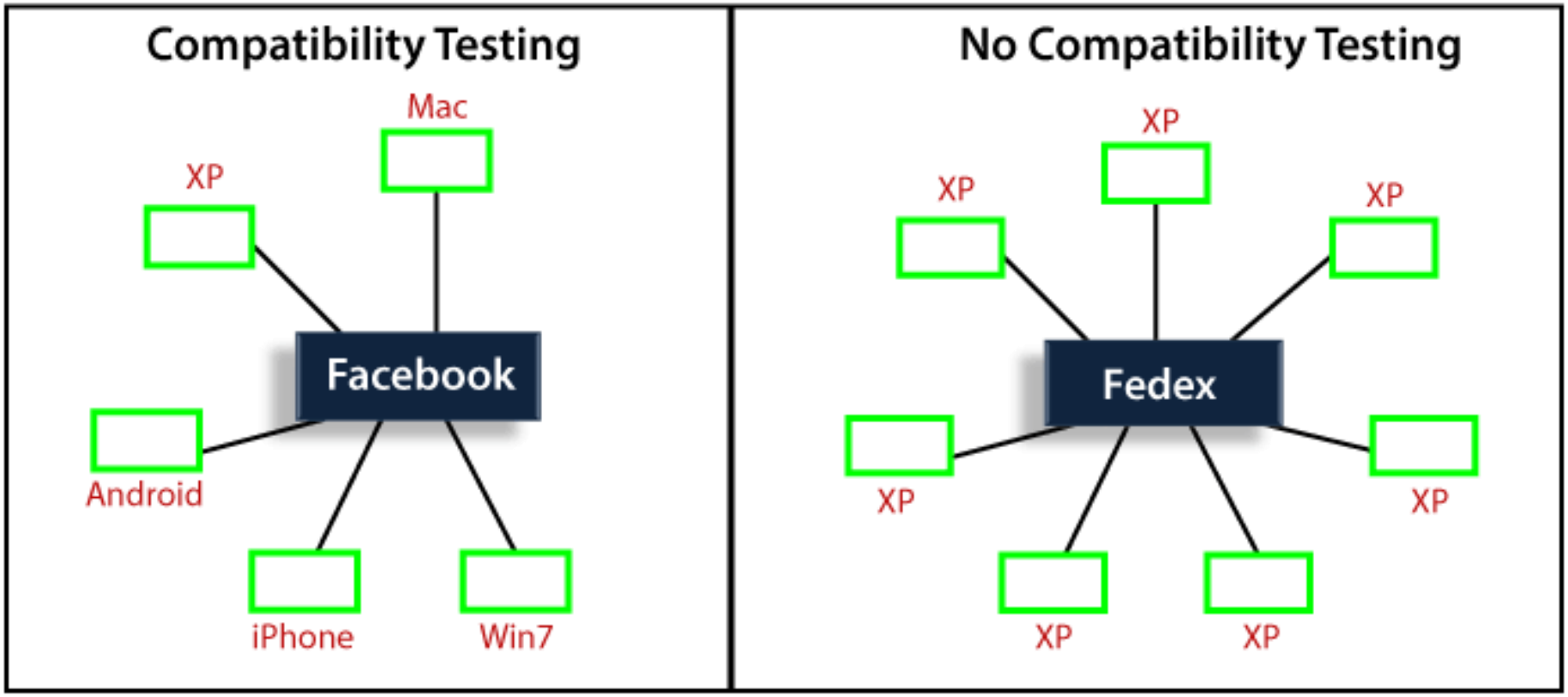
Functional Testing

- The main objective of functional testing is checking the functionality of the software system. It concentrates on:
- **Basic Usability:** Functional Testing involves the usability testing of the system. It checks whether a user can navigate freely without any difficulty through screens.
- **Accessibility:** Functional testing test the accessibility of the function.
- **Mainline function:** It focuses on testing the main feature.
- **Error Condition:** Functional testing is used to check the error condition. It checks whether the error message displayed.
- **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.

Functional Testing

- Why we use compatibility testing?
- Once the application is stable, we moved it to the production, it may be used or accessed by multiple users on the different platforms, and they may face some compatibility issues, to avoid these issues, we do one round of compatibility testing.
- When should we perform Compatibility testing?
- Generally, we go for compatibility testing, only when the application or software is functionally stable.

Functional Testing



Error Guessing

- Error guessing is a technique in which there is no specific method for identifying the error. It is based on the experience of the test analyst, where the tester uses the experience to guess the problematic areas of the software. It is a type of black box testing technique which does not have any defined structure to find the error.
- In this approach, every test engineer will derive the values or inputs based on their understanding or assumption of the requirements, and we do not follow any kind of rules to perform error guessing technique.
- The accomplishment of the error guessing technique is dependent on the ability and product knowledge of the tester because a good test engineer knows where the bugs are most likely to be, which helps to save lots of time.

How does the error guessing technique be implemented?

- The implementation of this technique depends on the experience of the tester or analyst having **prior experience** with similar applications. It requires only well-experienced testers with quick error guessing technique. This technique is used to find errors that may not be easily captured by formal black box testing techniques, and that is the reason, it is done after all formal techniques.
  - The main purpose of this technique is to identify common errors at any level of testing by exercising the following tasks:
  - Enter blank space into the text fields.
  - Null pointer exception.
  - Enter invalid parameters.
  - Divide by zero.
  - Use maximum limit of files to be uploaded.
  - Check buttons without entering values.
  - *The increment of test cases depends upon the ability and experience of the tester.*

Purpose of Error guessing

- The main purpose of the error guessing technique is to deal with all possible errors which cannot be identified as informal testing.
- The main purpose of error guessing technique is to deal with all possible errors which cannot be identified informal testing.
- It must contain the all-inclusive sets of test cases without skipping any problematic areas and without involving redundant test cases.
- This technique accomplishes the characteristics left incomplete during the formal testing.

Purpose of Error guessing

- There are some factors that can be used by the examiner while using their experience -
    - Tester's intuition
    - Historical learning
    - Review checklist
    - Risk reports of the software
    - Application UI
    - General testing rules
    - Previous test results
    - Defects occurred in the past
    - Variety of data which is used for testing
    - Knowledge of AUT

Error guessing Example

- Suppose we have one bank account, and we have to deposit some money over there, but the amount will be accepted on a particular range of **which is 5000-7000**. So here, we will provide the different input's value until it covers the maximum test coverage based on the error guessing technique, and see whether it is accepted or give the error message:

- **Note:**
- **Condition: if amount >5000 and amount<7000 amount**
- And, if we enter 5000 → error message (not accepted based on the condition)
- 7000→ error message (not accepted based on the condition)

Error guessing Example

value	description
6000	Accept
5555	Accept
4000	Error message
8000	Error message
blank	Error message
100$	Error message
----	----
----	----
Maximum test coverage	


Thank You