## 3. System Testing / End to End testing / System Integration testing (SIT) / combination of functional and non-functional testing

-SIT is a process of checking completeness and correctness of functionality of the system as per SRS and BRS

-System Testing (ST) using a black box testing technique performed to evaluate the complete system against requirements.

-In System testing, the functionalities of the system are tested from an end-to-end means all features.

- Done by testers

-System Testing is usually combination of functional and non-functional testing.

### #1) Functional Testing

-Functional testing is a process of checking correctness and completeness of the functionality of the build.
-Functional testing is the process of checking internal functionality depend upon external functionality
-This type deals with the functional requirements or specifications of an application to be tested by providing the input and comparing with the expected output.

**Few major types of Functional Testing are:**

- Smoke Testing
- Sanity Testing
- Integration Testing
- System Testing
- Regression Testing
- User Acceptance Testing

### #2) Non-Functional Testing

Apart from the functionalities of the requirements, there are several non-functional requirements also need 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
- Recovery testing
- Configuration testing
- Installation testing
- Globalization testing
- Parallel testing
- Sanitation testing

**System Testing Process:**

System Testing is performed in the following steps:

- **Create Test Case:**

  Generate test case for the testing process.
- **Create Test Data:**

  Generate the data that is to be tested.
- **Test Environment Setup:**

  Create testing environment for the better quality testing.
- **Execute Test Case:**

  After the generation of the test case and the test data, test cases are executed.
- **Defect Reporting:**

  Defects in the system are detected.
- **Regression Testing:**

  It is carried out to test the side effects of the testing process.
- **Log Defects:**

  Defects are fixed in this step.

- **Retest:**

  If the test is not successful then again test is performed.

## Black box testing

**Black box testing /** behavioral, opaque-box / closed-box / specification-based / eye-to-eye testing.

-Black box testing is known system and function testing.
-This testing is done by tester.
-Overall functionality get checked in this type of testing.
-Tester check internal functionality depend upon external functionality.
Example-Tester check whenever data is sign module got entered and users press sign up button,this button is process to store entered data.Tester check whether the data is stored correctly or not.
So here internal functionality is storing of data and external functionality is filling up data in fields and submit buttons process.

-Tester test the positive and negative scenarios.
**Positive scenario means-**
If suppose we have mobile number field with 10 digit functionality then as a tester we will check field functionality by entering 10 digit number whether it works or not.

**Negative scenario means-**
If suppose we have mobile number field with 10 digit functionality then as a tester if we check with 9 digits or less as it should not accept or more than 10 digits.

-**The main focus in Black Box Testing is on the functionality of the system as a whole.**

## Black Box Testing Techniques / Test case design Techniques

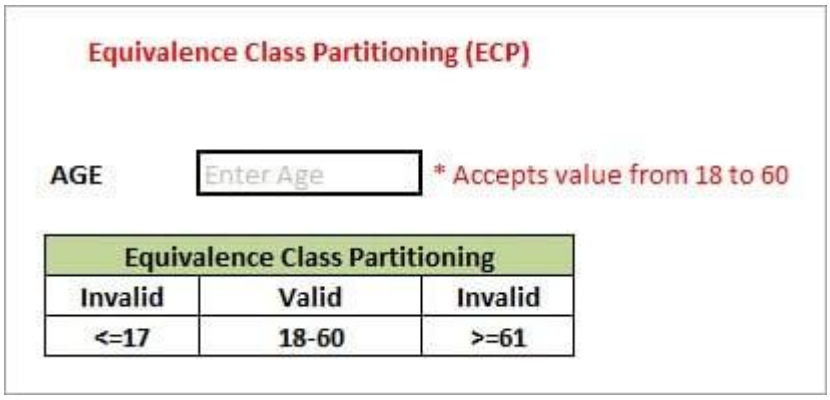
- Equivalence Partitioning
- Boundary Value Analysis
- Decision Table Testing
- State Transition Testing
- Error Guessing
- Graph-Based Testing Methods

## Equivalence Partitioning

-Equivalence Partitioning Method is also known as Equivalence class partitioning (ECP).

-It is a software testing technique or black-box testing or test case design technique by which data divided into classes or groups, and with the help of these classes of data, test cases can be made.

Example-




As present in the above image, an “AGE” text field accepts only the numbers from 18 to 60. There will be three sets of classes or groups.

**Two invalid classes will be:**

a) Less than or equal to 17.

b) Greater than or equal to 61.

One valid class will be anything between 18 to 60.

**#2) Boundary Value Analysis-**

-This technique uses to test the data set according to the boundary condition.

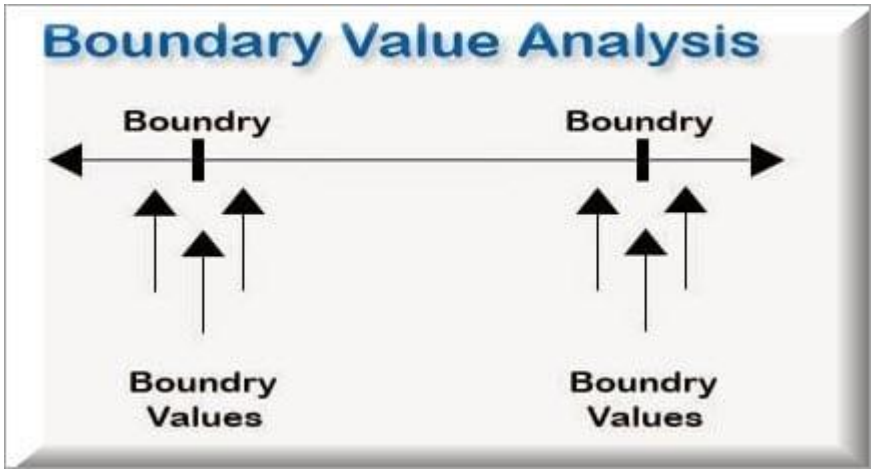
-From the name itself, we can understand that in this technique we focus on the values at boundaries itself

-In boundary value analysis both the valid inputs and invalid inputs are being tested to verify the issues.

-Example-First identify the boundary data then do the below things to find out the data by which we can test-

Invalid Boundary Check { Min-1 ; Max+1 }

Valid Boundary Check {Min; Min+1 ;Max-1;Max }




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.

## 3. Decision Table Testing

A decision table is the tabular representation of several input values, cases, rules, and test conditions.

Testers can use decision table testing to test the results of several input combinations

Example pf login screen by Decision Base Table =

The condition is simple if the user provides correct username and password the user will be redirected to the homepage. If any of the input is wrong, an error message will be displayed.

| Conditions | Rule 1 | Rule 2 | Rule 3 | Rule 4 |
|---|---|---|---|---|
| **Username (T/F)** | F | T | F | T |
| **Password (T/F)** | F | F | T | T |

| | | | | |
|---|---|---|---|---|
| **Output (E/H)** | E | E | E | H |

Legend:

- T – Correct username/password
- F – Wrong username/password
- E – Error message is displayed
- H – Home screen is displayed

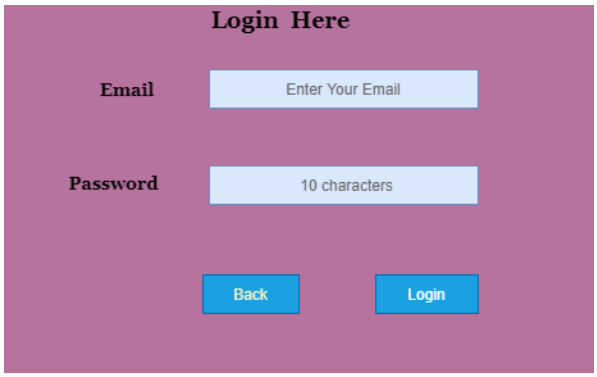
Interpretation:

- Test Case 1 – Username and password both were wrong. The user is shown an error message.
- Test Case 2 – Username was correct, but the password was wrong. The user is shown an error message.
- Test Case 3 – Username was wrong, but the password was correct. The user is shown an error message.
- Test Case 4 – Username and password both were correct, and the user navigated to homepage

**State transition testing:**

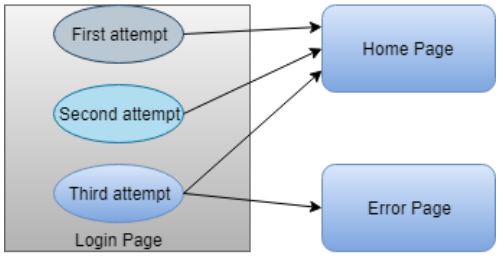
-State Transition Testing is a technique that is used to test the different states of the application.

The state of the system changes depending upon the conditions or .Accordingly tester needs to test them.




Let see in the diagram:

There is a login function of an application which provides a maximum three number of attempts, and after exceeding three attempts, it will be directed to an error page.




## State transition table

| STATE | LOGIN | VALIDATION | REDIRECTED |
|---|---|---|---|
| **S1** | **First Attempt** | **Invalid** | **S2** |
| **S2** | **Second Attempt** | **Invalid** | **S3** |
| **S3** | **Third Attempt** | **Invalid** | **S5** |
| **S4** | **Home Page** | | |
| **S5** | **Error Page** | | |

In the above state transition table, we see that state S1 denotes first login attempt. When the first attempt is invalid, the user will be directed to the second attempt (state S2). If the second attempt is also invalid, then the user will be directed to the third attempt (state S3). Now if the third and last attempt is invalid, then the user will be directed to the error page (state S5).

But if the third attempt is valid, then it will be directed to the homepage (state S4).

## 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 techniques

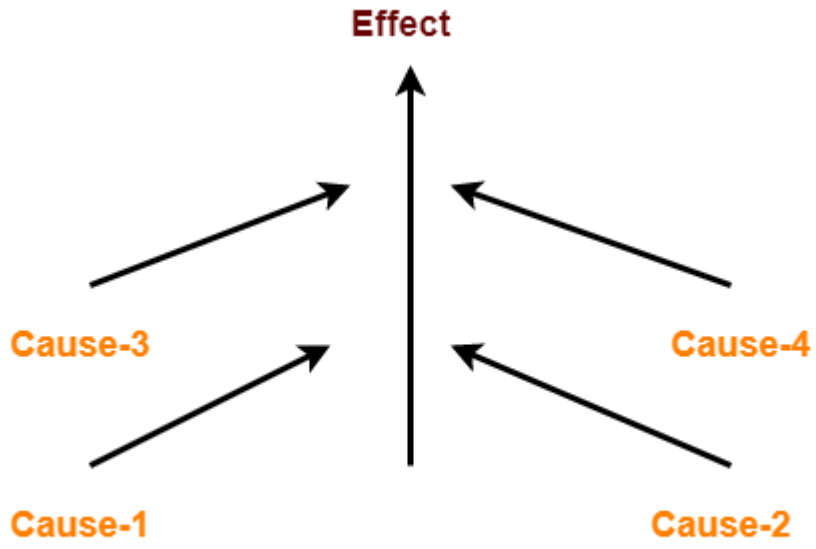
-Few common areas where we can use in error guessing:

- Divide by zero.
- Accepting the Submit button without any value.
- File upload without attachment.
- File upload with other extensions or files like PPT,PDF etc
- File upload with less than or more than the limit size.

## 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.




Cause-Effect Flow Diagram

- A “Cause” stands for a distinct input condition that fetches about an internal change in the system.
- An “Effect” represents an output condition, a system state that results from a combination of causes.

**What we are testing in functional testing –**

We check functionality and the coverage are-

1. Behavioral Coverage
2. I/P Domain Coverage
3. Error Handling Coverage
4. Back End coverage
5. Service Level Coverage
6. Calculation base coverage

## Behavioral Coverage

-In this we check the property and behavior of object.

## Example-

Property of check box - Do tick when user click

Behavior of check box – Check/Uncheck

Property of Dropdown – to show hidden list on click

Behavior of dropdown – Show / hide list

## I/P Domain coverage

-Here we check the input data with different condition and can use Boundary value analysis and Equivalence class partitioning techniques.

## Error Handling coverage

-Error handling coverage include checking whether system show error message or not as per the requirement.

-If in customer requirement, mobile number should accept the 10 digits and developer build according to its them if customer enter 2 digits and click on OK button then system should highlight text box with red color with error message “Please enter 10-digit mobile number”

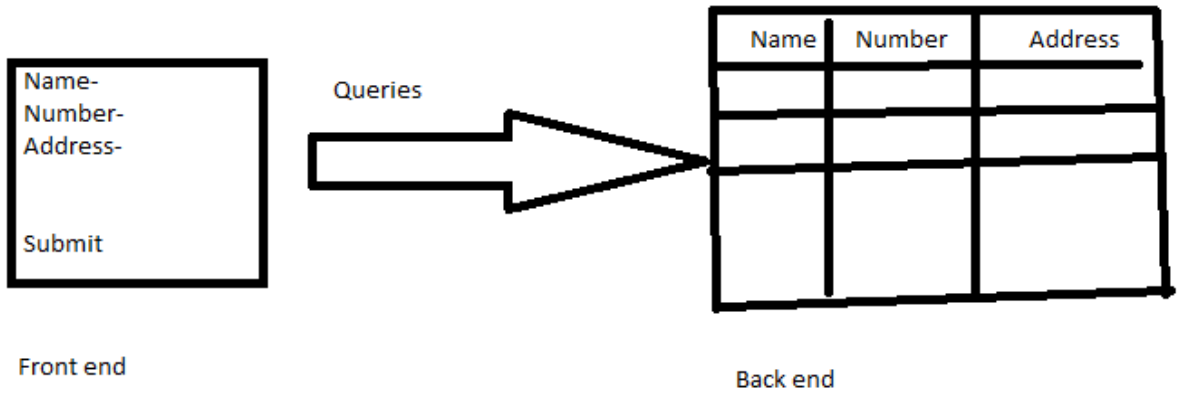
-So, this is the process of validation whether.

## Back-end coverage

-Back end of any software system is database.

-In back-end coverage dev check whether the entered information from user get stored in database or not

-We also check whether data get fetch from db or not.




## Service Level Coverage.

In SRS, BA creates functional flow diagram and accordingly tester test the sequence of function and modules.

This aspect of sequentially of functional modules get tested in service level coverage.

Quality product should be release by doing all QA activities.

## Calculation base Coverage:

Here arithmetic operation can be tested

Arithmetic operation means addition, multiplication, sub and division etc

Example-If we add 1 thing in cart and it is of 900 rs then total is 900 and again we add 2 things which are 50 rs then total is 1000.

Then remove 1 thing of 50 rs then 950 is total. This gets check in this coverage.