Functional Testing

Functional Testing, also known as black box testing, is a type of software testing that verifies whether an application meets its functional requirements. It focuses on testing the business logic, user interactions, and expected outputs rather than the internal code structure.

Purpose of Functional Testing

- Ensures that the application behaves as expected.
- Validates that all functional requirements are met.
- Detects functional defects in the application.
- Ensures user satisfaction by testing real-world scenarios.

Example of Functional Testing

Test Case	Steps	Expected Result
Test Scenario 1: Valid Login		
Login with valid credentials	1. Open login page 2. Enter valid username and password 3. Click "Login"	User is successfully logged in
Test Scenario 2: Invalid Password		
Login with invalid password	1. Open login page 2. Enter correct username and wrong password 3. Click "Login"	Error message: "Invalid password"
Test Scenario 3: Blank Fields		
Login with empty fields	1. Open login page 2. Click "Login" without entering anything	Error message: "Username and password required"

1. Boundary Value Analysis (BVA)

BVA is a black-box testing technique that focuses on testing the boundaries (edges) of input values. The idea is that most errors occur at **extreme boundaries** rather than at the center. Instead of testing a large range of values, we test the minimum, just below minimum, just above minimum, maximum, just below maximum, and just above maximum.

Example: Suppose we are testing an application that only allows users between 18 and 60 years old to register.

Test Case	Input Age	Expected Result
Below Minimum	17	Error: "Invalid age"
Minimum Value	18	Success
Just Above Minimum	19	Success
Just Below Maximum	59	Success
Maximum Value	60	Success
Above Maximum	61	Error: "Invalid age"

2. Robust Testing

It is an extension of Boundary Value Analysis (BVA). In Robust Testing, we also test extreme invalid values (e.g., negative numbers, blank input, special characters). It ensures the application handles unexpected inputs gracefully.

Example: Username Field (3–15 characters, only letters)

Test Case	Input Age	Expected Result
Below Minimum	"ab"	Error: "Too short"
Valid Minimum	"abc"	Success
Just Above Minimum	"abcd"	Success
Just Below Maximum	"abcdefghijklmno"	Success
Maximum Length	"abcdefghijklmno"	Success
Above Maximum	"abcdefghijklmnop"	Error: "Too long"
Robust Cases	"" (blank), "12345", "@!$#%"	Error: "Invalid characters"

3. Equivalence Class Partitioning (ECP)

ECP divides input values into valid and invalid partitions.
We assume that if one test case in a partition works, all others in that partition will work too.
It helps in reducing redundant test cases.

Example: Online Store Discount based on the order amount, for example: Orders below Rs 1000 gets no discount, orders between Rs 1000 to Rs 5000 get 10% discount & orders above Rs 5000 get 20% discount.

Equivalence Partitions:

Partition	Example Values (Rs)	Expected Result
nvalid (Below Min)	500	No Discount
Valid (1000 - 5000)	2000, 3000	10% Discount
Valid (Above 5000)	6000	20% Discount
Invalid (Negative values)	-50	Error

4. Decision Table Testing

A decision table is a way to represent complex business rules with multiple conditions.
Each row represents a test case covering different combinations of inputs and expected outputs.
It is useful for applications with complex logic and multiple conditions.

Example: Loan Approval System
A bank approves a loan if:

- Applicant has a Good Credit Score (≥700)
- Applicant has a Stable Job (Yes/No)

Test Case	Credit Score >= 700?	Has stable job?	Loan Approved?
1	Yes	Yes	Yes
2	Yes	No	Yes
3	No	Yes	No
4	No	No	No

5. Cause-Effect Graphing

It is a graphical representation of inputs (causes) and expected outcomes (effects).
Used to identify dependencies and relationships between inputs and outputs.
Helps in designing efficient test cases.
Example: ATM Withdrawal
A bank allows an ATM withdrawal only if:

1. The account has a sufficient balance.
2. The ATM card is valid.

Cause (Input)	Effect (Output)
Sufficient Balance	Money is dispensed
Insufficient Balance	Error: "Insufficient Funds"
Valid Card	Transaction proceeds
Expired Card	Error: "Invalid Card"