

Manual Testing

1. Testing the functionality or Behaviour of the application manually according to the customer requirement is called as Manual Testing.
2. It is used in the case of constructor overloading
3. Time Taken is more
4. Human-driven
5. Highly effective
6. Maintenance is less in Manual Testing
7. Easier to adapt changes if requestment is changing
8. Highly effective testing for UI Testing

Automation Testing

1. Testing the functionality or Behaviour of the application with the help of tools like Selenium,QTPwith a scripting language like java,python,javascript,ruby etc according to the customer requirement is called as Automation Testing.
2. It is used in the case of inheritance
3. Time Taken is less to complete task
4. Tool-driven
5. May not be effective for every Testing Types.
6. Maintenance is more in Automation Testing
7. Not easy to adapt changes since scripts has to be changed
8. Not so highly effective for UI Testing since it cannot check the look and feel of application, colour etc ¹



Black box Testing

1. Testing the application without looking at the code
2. Testers or end user/customer do this
3. To do this testing testers follow some techniques like Equivalence class partitioning, Boundary value analysis etc
4. The name of this testing is black because code will not be visible

White Box Testing

1. Testing each line of the code is called as white box testing
2. Developers do this
3. To do this testing developers use some techniques like Path Testing, Code Coverage etc
4. The name of this testing is white because code will be visible

ReTesting

1. ReTesting is done to make sure the fixed bug is working fine as per the customer requirement.
2. Can be done by both developer as well as Testers
3. Takes very less time
4. ReTesting should be done manually
5. The scope of ReTesting is less

Regression Testing

1. We do this to make the changes like adding a feature, deleting a feature, fixing a bug or any code modification is not impacting the unchanged features of the application.
2. Will be done only by the testers
3. Takes more time.
4. Regression should be done by Automation Testing
5. The scope of regression Testing is more usually full application.

Test Scenario Vs Test Cases

Test Scenario

1. High level documentation
2. By looking into this it will be hard to test the application until and unless you have good project knowledge
3. It tell what to test
4. Test Scenario doesn't have data that you need to enter to an application to test
5. Test Scenario is just a short sentence which says what to test
6. Test Scenarios are derived from Requirement
7. Need less time to prepare
8. With the help of test scenarios we validate the functionality of an application
9. Required less resource to write and execute compared with Test Cases

Test Cases

1. Detailed Documentation
2. By looking into test cases you can easily test any application, no matter you have good project knowledge or not
3. It tells how to test
4. It has all exact data that you need to enter
5. Test cases have test scenario, steps to reproduce, precondition, expected output, actual output, result, Bug Id, Automated or not
6. Test Cases is derived from Test Scenarios
7. Need more time to prepare
8. Here we can check or validate if test scenarios is correct or not.
9. Required more resource to write and execute

B2B VS B2C

B2B

1. Stand for Business to Business.
2. Traffic will be less on these apps.
3. Longer Release cycle

B2C

1. Stands for Business to Customer.
2. Traffic will be more on these apps.
3. Shorter release cycle

Service Based Company

1. They gives services to others
2. Example:

Accenture,Deloitte Consulting,IBM
Global Services,Salesforce, Tech
Mahindra,Wipro,Infosys etc

Product based Company

1. They develop their own product
2. Examples:

Google,Flipkart,Apple,Tesla,Samsung,
swiggy,ola,uber etc

Positive Testing

1. It will be performed as per the requirements.
2. It will be done first
3. It doesn't cover all possible test cases.
4. It takes less time.
5. It can be done by anyone in the testing team.

Negative Testing

1. It will not be performed as per the requirements.
2. It will done later.
3. It can help us to cover more possible test cases.
4. It takes more time.
5. It will be done by professionals or experienced testers.

Static Testing Vs Dynamic Testing

Static Testing

1. Performed at the early stage of development.
2. Here we don't execute the software
3. Cost of fixing the issue is less
4. We do this to avoid bugs that may arise later
5. We do this before build is ready.
6. Examples: Code review, Walkthrough, Test Plan, Test Cases
7. Can be done by both Dev and Tester.

Dynamic Testing

1. Performed at the later stage of the development.
2. Here we execute the software
3. Cost of fixing the issue is more.
4. We do this to find bugs
5. We do this after build is ready.
6. Examples: Component Testing, Integration Testing, System Testing etc
7. Can be done by both dev and tester but it is tester responsibility.

Verification Vs Validation

Verification

1. Verifying CRS,SRS,HLD & LLD and to check whether it is according to the requirement or not is called as Verification
2. Following activities are involved in Verification: Reviews, Meetings and Inspections.
3. Execution of code does not come under Verification.
4. Verification is carried out before the Validation.
5. Following items are evaluated during Verification: Plans, Requirement Specifications, Design Specifications, Code, Test Cases etc,
6. Cost of errors caught in Verification is less than errors found in Validation.
7. Done by Developer

Validation

1. Testing the functionality of an application by executing test cases is called as Validation
2. Following activities are involved in Validation: Testing like Black box testing and all functional and non functional testing.
3. Execution of code is comes under Validation.
4. Validation activity is carried out just after the Verification.
5. Following item is evaluated during Validation: Actual product or Software under test.
6. Cost of errors caught in Validation is more than errors found in Verification.
7. Done by Tester