

### 8.1 Testing Process

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. In simple words, testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

According to ANSI/IEEE 1059 standard, Testing can be defined as - A process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item.

#### General characteristics of strategic testing

- To perform effective testing, a s/w team should conduct technical reviews
- Testing begins at the component level and work toward the integration of entire computer based system.
- Different testing techniques are appropriate at different points in time.
- Testing is done by developer of s/w and for large projects by an independent test group.
- Testing and debugging are different activities, but debugging must be accommodated in any test strategy.

In other words software testing is a verification and validation process

Verification: “Does the product meet its specification?” The set of activities that ensure the s/w correctly implements a specific function or algorithm.

Validation: “Does the product perform as desired?” The set of activities that ensure that the s/w has been built is traceable to customer requirement.

### 8.2 Some important terminologies

#### Test case

A structured test script that describes how a function or feature should be tested, including test steps, expected results preconditions and post conditions.

#### Test Scenario

A sequence of activities performed in a system, such as logging in, signing up a customer, ordering products, and printing an invoice. You can combine test cases to form a scenario especially at higher test levels.

### **Mocking**

Mocking means creating a fake version of an external or internal service that can stand-in for the real one, helping your tests run more quickly and more reliably. When our implementation interacts with an object's properties, rather than its function or behavior, a mock can be used.

### **Stubbing**

Stubbing, like mocking, means creating a stand-in, but a stub only mocks the behavior, but not the entire object. This is used when your implementation only interacts with a certain behaviour of the object.

## **8.3 Unit testing**

- Unit testing is the first level of functional testing in order to test any software. In this, the test engineer will test the module of an application independently or test all the module functionality is called unit testing.
- The primary objective of executing the unit testing is to confirm the unit components with their performance.
- Here, a unit is defined as a single testable function of software or an application. And it is verified throughout the specified application development phase.
- If some code which is dealing with the read and write function on database we cannot test that as a unit because the database is linked with it now.

## **8.4 Integration testing**

Upon completion of unit testing, the units or modules are to be integrated which gives rise to integration testing. The purpose of integration testing is to verify the functional, performance, and reliability between the modules that are integrated.

Integration Strategies:

- Big-Bang Integration
- Top Down Integration

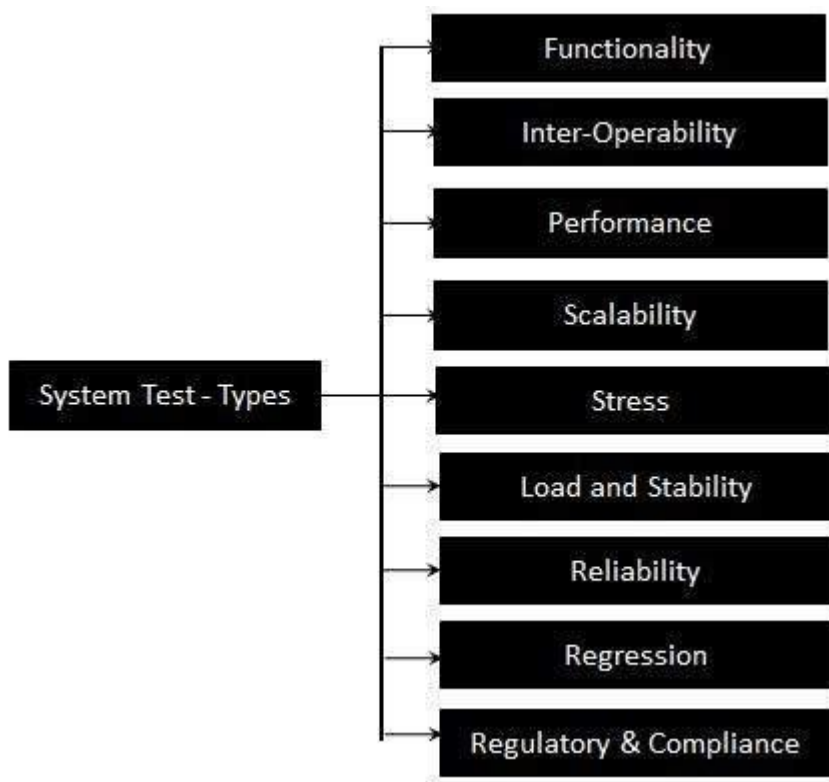
- Bottom Up Integration
- Hybrid Integration

### **8.5 System testing**

System Testing (ST) is a black box testing technique performed to evaluate the complete system the system's compliance against specified requirements. In System testing, the functionalities of the system are tested from an end-to-end perspective.

System Testing is usually carried out by a team that is independent of the development team in order to measure the quality of the system unbiased. It includes both functional and Non-Functional testing.

Types of System Tests:



### **8.6 Regression testing**

Regression testing a black box testing technique that consists of re-executing those tests that are impacted by the code changes. These tests should be executed as often as possible throughout the software development life cycle.

### **Types of Regression Tests:**

**Final Regression Tests:** - A "final regression testing" is performed to validate the build that hasn't changed for a period of time. This build is deployed or shipped to customers.

**Regression Tests:** - A normal regression testing is performed to verify if the build has NOT broken any other parts of the application by the recent code changes for defect fixing or for enhancement.

### **Selecting Regression Tests:**

- Requires knowledge about the system and how it affects by the existing functionalities.
- Tests are selected based on the area of frequent defects.
- Tests are selected to include the area, which has undergone code changes many times.
- Tests are selected based on the criticality of the features.

### **Regression Testing Steps:**

Regression tests are the ideal cases of automation which results in better Return On Investment (ROI).

- Select the Tests for Regression.
- Choose the apt tool and automate the Regression Tests
- Verify applications with Checkpoints
- Manage Regression Tests/update when required
- Schedule the tests
- Integrate with the builds
- Analyze the results

## **8.7 Performance Testing**

Performance testing, a non-functional testing technique performed to determine the system parameters in terms of responsiveness and stability under various workload. Performance testing measures the quality attributes of the system, such as scalability, reliability and resource usage.

### Performance Testing Techniques:

**Load testing** - It is the simplest form of testing conducted to understand the behaviour of the system under a specific load. Load testing will result in measuring important business critical transactions and load on the database, application server, etc., are also monitored.

**Stress testing** - It is performed to find the upper limit capacity of the system and also to determine how the system performs if the current load goes well above the expected maximum.

**Soak testing** - Soak Testing also known as endurance testing, is performed to determine the system parameters under continuous expected load. During soak tests the parameters such as memory utilization is monitored to detect memory leaks or other performance issues. The main aim is to discover the system's performance under sustained use.

**Spike testing** - Spike testing is performed by increasing the number of users suddenly by a very large amount and measuring the performance of the system. The main aim is to determine whether the system will be able to sustain the workload.

### Attributes of Performance Testing:

- Speed
- Scalability
- Stability
- reliability

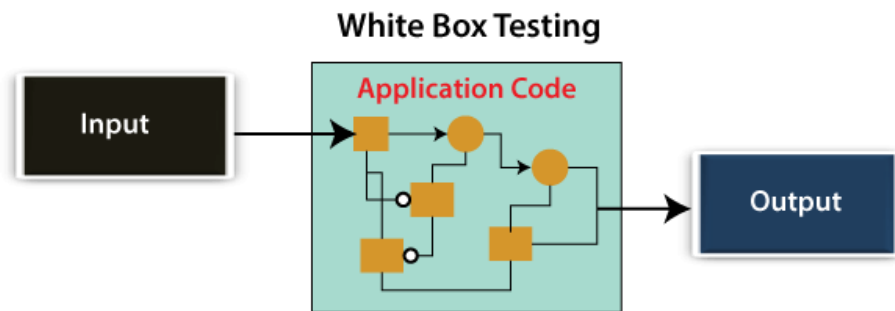
### 8.8 White box testing and black box testing

In white-box testing, the developer will inspect every line of code before handing it over to the testing team or the concerned test engineers.

Subsequently, the code is noticeable for developers throughout testing; that's why this process is known as **WBT (White Box Testing)**.

In other words, we can say that the **developer** will execute the complete white-box testing for the particular software and send the specific application to the testing team.

The purpose of implementing the white box testing is to emphasize the flow of inputs and outputs over the software and enhance the security of an application.



White box testing is also known as **open box testing**, **glass box testing**, **structural testing**, **clear box testing**, and **transparent box testing**.

### Black Box Testing

Another type of manual testing is **black-box testing**. In this testing, the test engineer will analyze the software against requirements, identify the defects or bug, and sends it back to the development team.

Then, the developers will fix those defects, do one round of White box testing, and send it to the testing team.

Here, fixing the bugs means the defect is resolved, and the particular feature is working according to the given requirement.

The main objective of implementing the black box testing is to specify the business needs or the customer's requirements.

In other words, we can say that black box testing is a process of checking the functionality of an application as per the customer requirement. The source code is not visible in this testing; that's why it is known as **black-box testing**.



### Types of Black Box Testing

Black box testing further categorizes into two parts, which are as discussed below:

- **Functional Testing**
- **Non-function Testing**

### 8.9 Acceptance testing

Acceptance testing, a testing technique performed to determine whether or not the software system has met the requirement specifications. The main purpose of this test is to evaluate the system's compliance with the business requirements and verify if it has met the required criteria for delivery to end users.

There are various forms of acceptance testing:

- User acceptance Testing
- Business acceptance Testing
- Alpha Testing
- Beta Testing

The acceptance test activities are designed to reach at one of the conclusions:

- Accept the system as delivered
- Accept the system after the requested modifications have been made
- Do not accept the system

### Acceptance Test Report - Attributes

The Acceptance test Report has the following attributes:

- Report Identifier
- Summary of Results
- Variations
- Recommendations
- Summary of To-DO List
- Approval Decision

### 8.10 Alpha and Beta testing

Alpha Testing is a type of acceptance testing; performed to identify all possible issues and bugs before releasing the final product to the end users. Alpha testing is carried out by the testers who are internal employees of the organization. The main goal is to identify the tasks that a typical user might perform and test them.

To put it as simple as possible, this kind of testing is called alpha only because it is done early on, near the end of the development of the software, and before beta testing. The main focus of alpha testing is to simulate real users by using a black box and white box techniques.



Beta Testing is performed by “real users” of the software application in “real environment” and it can be considered as a form of external User Acceptance Testing. It is the final test before shipping a product to the customers. Direct feedback from customers is a major advantage of Beta Testing. This testing helps to test products in customer’s environment.

Beta version of the software is released to a limited number of end-users of the product to obtain feedback on the product quality. Beta testing reduces product failure risks and provides increased quality of the product through customer validation.

### **KEY DIFFERENCE**

- Alpha Testing is performed by the Testers within the organization whereas Beta Testing is performed by the end users.
- Alpha Testing is performed at Developer’s site whereas Beta Testing is performed at Client’s location.
- Reliability and Security testing are not performed in-depth in Alpha Testing while Reliability, Security and Robustness are checked during Beta Testing.



- Alpha Testing involves both Whitebox and Blackbox testing whereas Beta Testing mainly involves Blackbox testing.
- Alpha Testing requires testing environment while Beta Testing doesn't require testing environment.
- Alpha Testing requires long execution cycle whereas Beta Testing requires only few weeks of execution.
- Critical issues and bugs are addressed and fixed immediately in Alpha Testing whereas issues and bugs are collected from the end users and further implemented in Beta Testing.

### **8.11 Debugging techniques, tools and approaches**

In the context of software engineering, debugging is the process of fixing a bug in the software. In other words, it refers to identifying, analyzing and removing errors. This activity begins after the software fails to execute properly and concludes by solving the problem and successfully testing the software. It is considered to be an extremely complex and tedious task because errors need to be resolved at all stages of debugging.

#### **Debugging Techniques:**

- Interactive debugging
- Print debugging (tracing)
- Remote debugging
- Post-mortem debugging

#### **Debugging Tools:**

Debugging tool is a computer program that is used to test and debug other programs. A lot of public domain software like gdb and dbx are available for debugging. They offer console-based command line interfaces. Examples of automated debugging tools include code based tracers, profilers, interpreters, etc.

Some of the widely used debuggers are:

- Radare2
- WinDbg
- Valgrind

#### **Debugging Approaches:**

The following are a number of approaches popularly adopted by programmers for debugging.

### **Brute Force Method:**

This is the foremost common technique of debugging however is that the least economical method. during this approach, the program is loaded with print statements to print the intermediate values with the hope that a number of the written values can facilitate to spot the statement in error. This approach becomes a lot of systematic with the utilisation of a symbolic program (also known as a source code debugger), as a result of values of various variables will be simply checked and breakpoints and watch-points can be easily set to check the values of variables effortlessly.

### **Backtracking:**

This is additionally a reasonably common approach. during this approach, starting from the statement at which an error symptom has been discovered, the source code is derived backward till the error is discovered. sadly, because the variety of supply lines to be derived back will increase, the quantity of potential backward methods will increase and should become unimaginably large so limiting the utilisation of this approach.

### **Cause Elimination Method:**

In this approach, a listing of causes that may presumably have contributed to the error symptom is developed and tests are conducted to eliminate every error. A connected technique of identification of the error from the error symptom is that the package fault tree analysis.

### **Program Slicing:**

This technique is analogous to backtracking. Here the search house is reduced by process slices. A slice of a program for a specific variable at a particular statement is that the set of supply lines preceding this statement which will influence the worth of that variable.