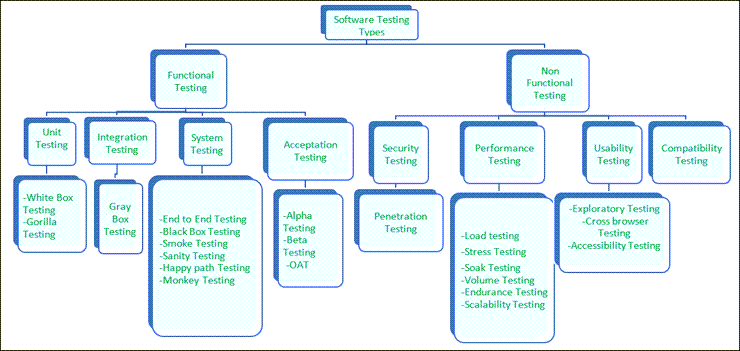
**SYSTEM TESTING**

**Types of Software Testing: Different Testing Types with Details**

We, as testers, are aware of the various types of Software Testing like Functional Testing, Non-Functional Testing, Automation Testing, Agile Testing, and their sub-types, etc.

Each type of testing has its own features, advantages, and disadvantages as well. However, in this tutorial, we have covered mostly each and every type of software testing which we usually use in our day-to-day testing life.

**Different Types of Software Testing**



**Functional Testing**

There are four main types of functional testing.

**#1) Unit Testing**

Unit testing is a type of software testing which is done on an individual unit or component to test its corrections. Typically, Unit testing is done by the developer at the application development phase. Each unit in unit testing can be viewed as a method, function, procedure, or object. Developers often use test automation tools such as NUnit, Xunit, JUnit for the test execution.

Unit testing is important because we can find more defects at the unit test level.

**For example,** there is a simple calculator application. The developer can write the unit test to check if the user can enter two numbers and get the correct sum for addition functionality.

**a) White Box Testing**

White box testing is a test technique in which the internal structure or code of an application is visible and accessible to the tester. In this technique, it is easy to find loopholes in the design of an application or fault in business logic. Statement coverage and decision coverage/branch coverage are examples of white box test techniques.

**b) Gorilla Testing**

Gorilla testing is a test technique in which the tester and/or developer test the module of the application thoroughly in all aspects. Gorilla testing is done to check how robust your application is.

**For example,** the tester is testing the pet insurance company’s website, which provides the service of buying an insurance policy, tag for the pet, Lifetime membership. The tester can focus on any one module, let’s say, the insurance policy module, and test it thoroughly with positive and negative test scenarios.

**#2) Integration Testing**

Integration testing is a type of software testing where two or more modules of an application are logically grouped together and tested as a whole. The focus of this type of testing is to find the defect on interface, communication, and data flow among modules. Top-down or Bottom-up approach is used while integrating modules into the whole system.

This type of testing is done on integrating modules of a system or between systems. **For example,** a user is buying a flight ticket from any airline website. Users can see flight details and payment information while buying a ticket, but flight details and payment processing are two different systems. Integration testing should be done while integrating of airline website and payment processing system.

**a) Gray box testing**

As the name suggests, gray box testing is a combination of white-box testing and black-box testing. Testers have partial knowledge of the internal structure or code of an application.

**#3) System Testing**

System testing is types of testing where tester evaluates the whole system against the specified requirements.

**a) End to End Testing**

It involves testing a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate.

**For example,** a tester is testing a pet insurance website. End to End testing involves testing of buying an insurance policy, LPM, tag, adding another pet, updating credit card information on users’ accounts, updating user address information, receiving order confirmation emails and policy documents.

**b) Black Box Testing**

Blackbox testing is a software testing technique in which testing is performed without knowing the internal structure, design, or code of a system under test. Testers should focus only on the input and output of test objects.

Detailed information about the advantages, disadvantages, and types of Black Box testing can be found here.

**c) Smoke Testing**

Smoke testing is performed to verify that basic and critical functionality of the system under test is working fine at a very high level.

Whenever a new build is provided by the development team, then the Software Testing team validates the build and ensures that no major issue exists. The testing team will ensure that the build is stable, and a detailed level of testing will be carried out further.

**For example,** tester is testing pet insurance website. Buying an insurance policy, adding another pet, providing quotes are all basic and critical functionality of the application. Smoke testing for this website verifies that all these functionalities are working fine before doing any in-depth testing.

**d) Sanity Testing**

Sanity testing is performed on a system to verify that newly added functionality or bug fixes are working fine. Sanity testing is done on stable build. It is a subset of the regression test.

**For example,** a tester is testing a pet insurance website. There is a change in the discount for buying a policy for second pet. Then sanity testing is only performed on buying insurance policy module.

**e) Happy path Testing**

The objective of Happy Path Testing is to test an application successfully on a positive flow. It does not look for negative or error conditions. The focus is only on valid and positive inputs through which the application generates the expected output.

**f) Monkey Testing**

Monkey Testing is carried out by a tester, assuming that if the monkey uses the application, then how random input and values will be entered by the Monkey without any knowledge or understanding of the application.

The objective of Monkey Testing is to check if an application or system gets crashed by providing random input values/data. Monkey Testing is performed randomly, no test cases are scripted, and it is not necessary to be aware  
of the full functionality of the system.

**#4) Acceptance Testing**

Acceptance testing is a type of testing where client/business/customer test the software with real time business scenarios.

The client accepts the software only when all the features and functionalities work as expected. This is the last phase of testing, after which the software goes into production. This is also called User Acceptance Testing (UAT).

**a) Alpha Testing**

Alpha testing is a type of acceptance testing performed by the team in an organization to find as many defects as possible before releasing software to customers.

**For example,** the pet insurance website is under UAT. UAT team will run real-time scenarios like buying an insurance policy, buying annual membership, changing the address, ownership transfer of the pet in a same way the user uses the real website. The team can use test credit card information to process payment-related scenarios.

**b) Beta Testing**

Beta Testing is a type of software testing which is carried out by the clients/customers. It is performed in the **Real Environment**before releasing the product to the market for the actual end-users.

Beta Testing is carried out to ensure that there are no major failures in the software or product, and it satisfies the business requirements from an end-user perspective. Beta Testing is successful when the customer accepts the software.

Usually, this testing is typically done by the end-users. This is the final testing done before releasing the application for commercial purposes. Usually, the Beta version of the software or product released is limited to a certain number of users in a specific area.

So, the end-user uses the software and shares the feedback with the company. The company then takes necessary action before releasing the software worldwide.

**c) Operational acceptance testing (OAT)**

Operational acceptance testing of the system is performed by operations or system administration staff in the production environment. The purpose of operational acceptance testing is to make sure that the system administrators can keep the system working properly for the users in a real-time environment.

**The focus of the OAT is on the following points:**

* Testing of backup and restore.
* Installing, uninstalling, upgrading software.
* The recovery process in case of natural disaster.
* User management.
* Maintenance of the software.

**Non-Functional Testing**

There are four main types of functional testing.

**#1) Security Testing**

It is a type of testing performed by a special team. Any hacking method can penetrate the system.

Security Testing is done to check how the software, application, or website is secure from internal and/or external threats. This testing includes how much software is secure from malicious programs, viruses and how secure & strong the authorization and authentication processes are.

It also checks how software behaves for any hacker’s attack & malicious programs and how software is maintained for data security after such a hacker attack.

**a) Penetration Testing**

Penetration Testing or Pen testing is the type of security testing performed as an authorized cyberattack on the system to find out the weak points of the system in terms of security.

Pen testing is performed by outside contractors, generally known as ethical hackers. That is why it is also known as ethical hacking. Contractors perform different operations like SQL injection, URL manipulation, Privilege Elevation, session expiry, and provide reports to the organization.

**Notes:** Do not perform the Pen testing on your laptop/computer. Always take written permission to do pen tests.

**#2) Performance Testing**

Performance testing is testing of an application’s stability and response time by applying load.

The word stability means the ability of the application to withstand in the presence of load. Response time is how quickly an application is available to users. Performance testing is done with the help of tools. Loader.IO, JMeter, LoadRunner, etc. are good tools available in the market.

**a) Load testing**

Load testing is testing of an application’s stability and response time by applying load, which is equal to or less than the designed number of users for an application.

**For example,** your application handles 100 users at a time with a response time of 3 seconds, then load testing can be done by applying a load of the maximum of 100 or less than 100 users. The goal is to verify that the application is responding within 3 seconds for all the users.

**b) Stress Testing**

Stress testing is testing an application’s stability and response time by applying load, which is more than the designed number of users for an application.

**For example,** your application handles 1000 users at a time with a response time of 4 seconds, then stress testing can be done by applying a load of more than 1000 users. Test the application with 1100,1200,1300 users and notice the response time. The goal is to verify the stability of an application under stress.

**c) Scalability Testing**

Scalability testing is testing an application’s stability and response time by applying load, which is more than the designed number of users for an application.

**For example,** your application handles 1000 users at a time with a response time of 2 seconds, then scalability testing can be done by applying a load of more than 1000 users and gradually increasing the number of users to find out where exactly my application is crashing.

Let’s say my application is giving response time as follows:

* 1000 users -2 sec
* 1400 users -2 sec
* 4000 users -3 sec
* 5000 users -45 sec
* 5150 users- crash – This is the point that needs to identify in scalability testing

**d) Volume testing (flood testing)**

Volume testing is testing an application’s stability and response time by transferring a large volume of data to the database. Basically, it tests the capacity of the database to handle the data.

**e) Endurance Testing (Soak Testing)**

Endurance testing is testing an application’s stability and response time by applying load continuously for a longer period to verify that the application is working fine.

**For example,** car companies soak testing to verify that users can drive cars continuously for hours without any problem.

**#3) Usability Testing**

Usability testing is testing an application from the user’s perspective to check the look and feel and user-friendliness.

**For example,** there is a mobile app for stock trading, and a tester is performing usability testing. Testers can check the scenario like if the mobile app is easy to operate with one hand or not, scroll bar should be vertical, background colour of the app should be black and price of and stock is displayed in red or green colour.

The main idea of usability testing of this kind of app is that as soon as the user opens the app, the user should get a glance at the market.

**a) Exploratory testing**

Exploratory Testing is informal testing performed by the testing team. The objective of this testing is to explore the application and look for defects that exist in the application. Testers use the knowledge of the business domain to test the application. Test charters are used to guide the exploratory testing.

**b) Cross browser testing**

Cross browser testing is testing an application on different browsers, operating systems, mobile devices to see look and feel and performance.

Why do we need cross-browser testing? The answer is different users use different operating systems, different browsers, and different mobile devices. The goal of the company is to get a good user experience regardless of those devices.

Browser stack provides all the versions of all the browsers and all mobile devices to test the application. For learning purposes, it is good to take the free trial given by browser stack for a few days.

**c) Accessibility Testing**

The aim of Accessibility Testing is to determine whether the software or application is accessible for disabled people or not.

Here, disability means deafness, colour blindness, mentally disabled, blind, old age, and other disabled groups. Various checks are performed, such as font size for visually disabled, colour and contrast for colour blindness, etc.

**#4) Compatibility testing**

This is a testing type in which it validates how software behaves and runs in a different environment, web servers, hardware, and network environment.