

K.T.S.P. MANDAL'S

**HUTUTMA RAJGURU MAHAVIDYALAYA,
RAJGURUNAGAR TAL-KHED, DIST-PUNE 410 505**

DEPARTMENT OF COMPUTER SCIENCE

F.Y.Bsc (Computer Science)

Semester- II

Software Testing

Subject – CS-362 -Software Testing

According to New Syllabus w.e.f. 2021-2022

Prof. Pallavi G. Darakhe

DEPARTMENT OF COMPUTER SCIENCE

Chapter 3. Levels Of Testing

Levels of Software Testing

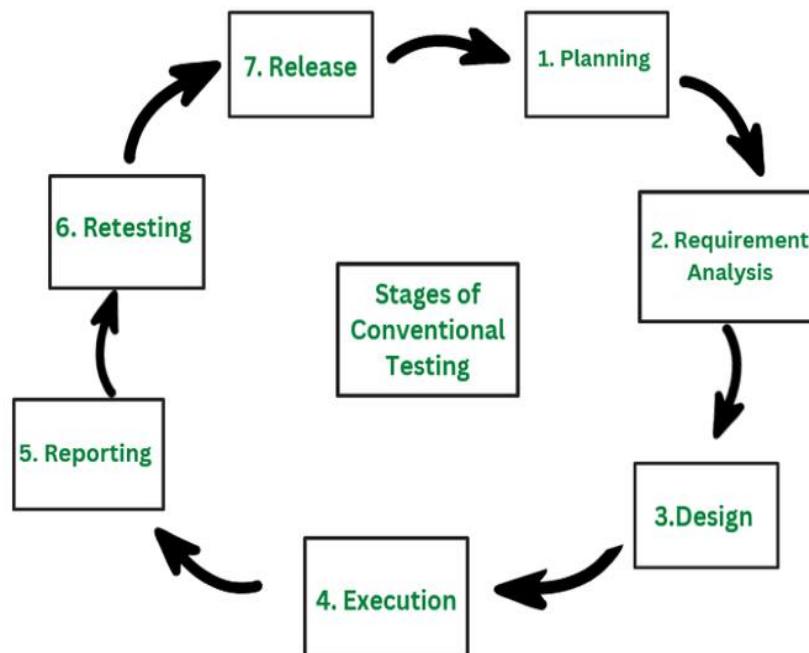
Software Testing is an activity performed to identify errors so that errors can be removed to obtain a product with greater quality. To assure and maintain the quality of software and to represent the ultimate review of specification, design, and coding, Software testing is required. There are different levels of testing :

1. Unit Testing: In this type of testing, errors are detected individually from every component or unit by individually testing the components or units of software to ensure that they are fit for use by the developers. It is the smallest testable part of the software.
2. Integration Testing: In this testing, two or more modules which are unit tested are integrated to test i.e., technique

interacting components, and are then verified if these integrated modules work as per the expectation or not, and interface errors are also detected.

3. **System Testing**: In system testing, complete and integrated Softwares are tested i.e., all the system elements forming the system are tested as a whole to meet the requirements of the system.
4. **Acceptance Testing**: This is a kind of testing conducted to ensure that the requirements of the users are fulfilled before its delivery and that the software works correctly in the user's working environment.

TEST STRATEGIES FOR CONVENTIONAL SOFTWARE

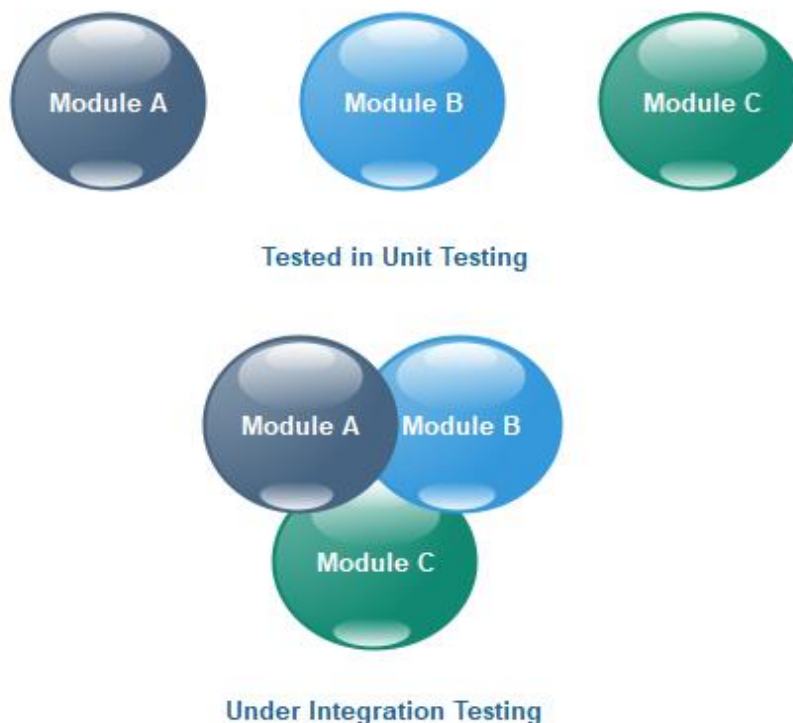


Unit Testing is a software testing technique using which individual units of software i.e. group of computer program modules, usage procedures, and operating procedures are tested to determine whether they are suitable for use or not. It is a testing method using which every independent module is tested to determine if there is an issue by the developer himself. It is correlated with the functional correctness of the independent modules. Unit Testing is defined as a type of software testing where individual components of a software are tested. Unit Testing of the software product is carried out during the development of an application. An individual component may be either an individual function or a procedure. Unit Testing is typically performed by the developer. In SDLC or V Model, Unit testing is the

first level of testing done before integration testing. Unit testing is a type of testing technique that is usually performed by developers. Although due to the reluctance of developers to test, quality assurance engineers also do unit testing.

Integration testing

Integration testing is the second level of the software testing process comes after unit testing. In this testing, units or individual components of the software are tested in a group. The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units. Unit testing uses modules for testing purpose, and these modules are combined and tested in integration testing. The Software is developed with a number of software modules that are coded by different coders or programmers. The goal of integration testing is to check the correctness of communication among all the modules.

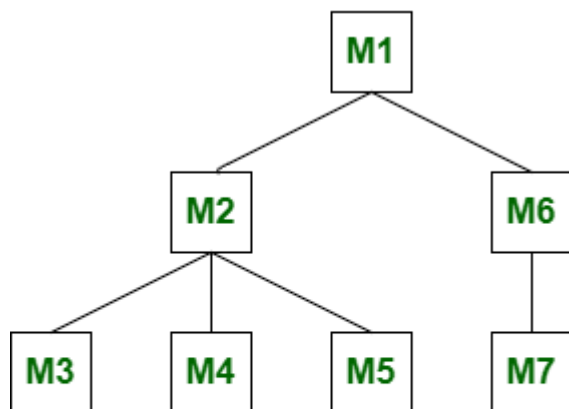


Top-down integration. Top-down integration testing is an incremental approach to construction of the software architecture.

Modules are integrated by moving downward through the control hierarchy, beginning with the main control module (main program). Modules subordinate to the main control module are incorporated into the structure in either a depth-first or breadthfirst manner. Referring to the following figure, depth-first integration integrates all components on a major control path of the program structure.

Example of Top Down Integration Testing

In the top-down integration testing, if the depth-first approach is adopted then we will start integration from module M1. Then we will integrate M2, then M3, M4, M5, M6, and at last M7.



Program Structure

In the top-down integration testing, if the breadth-first approach is adopted, then we will integrate module M1 first, then M2, and M6. Then we will integrate module M3, M4, M5, and at last M7.

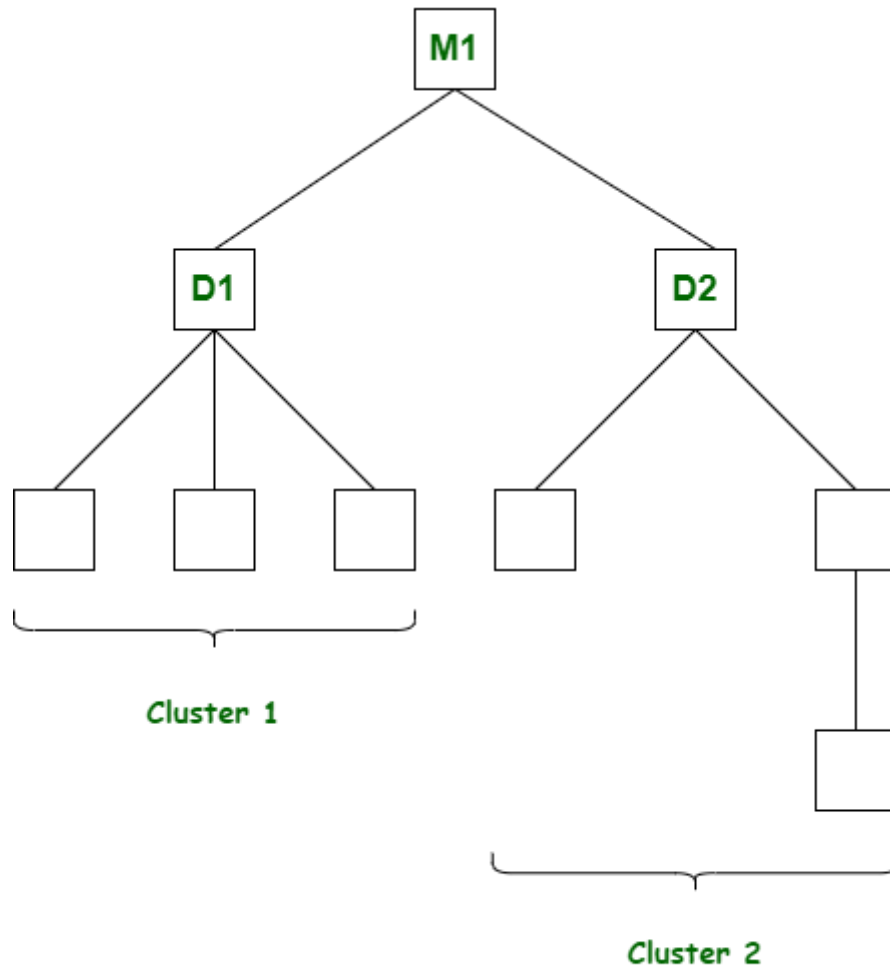
Bottom Up Integration Testing

Bottom-up Testing **known as** is a type of incremental [integration testing](#) approach in which testing is done by integrating or joining two or more modules by moving upward from bottom to top through the control flow of the architecture structure. In these, low-level modules are tested first, and then high-level modules are tested. This type of testing or approach is also inductive reasoning and is used as a synthesis synonym in many cases. Bottom-up testing is user-friendly testing and results in an increase in overall software

development. This testing results in high success rates with long-lasting results.

Example of Bottom-Up Integration Testing

In the last, modules or components are combined to form cluster 1 and cluster 2. After this, each cluster is tested with the help of a control program. The cluster is present below the high-level module or driver. After testing, the driver is removed and clusters are combined and moved upwards with modules.



Bottom Up Integration Testing

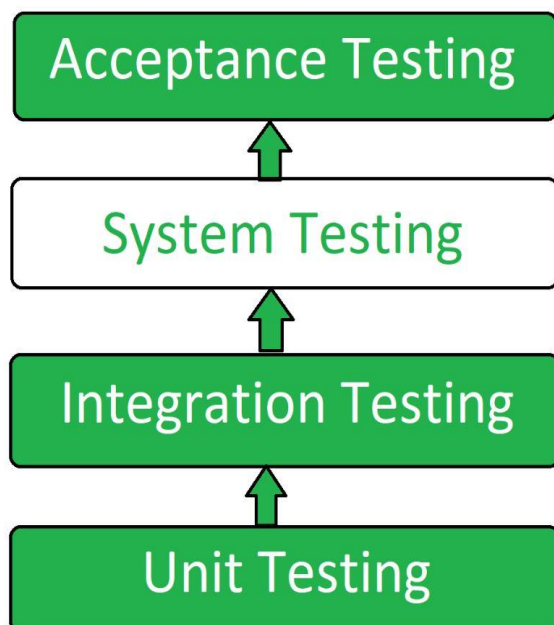
System Testing

INTRODUCTION:

System testing is a type of software testing that evaluates the overall functionality and performance of a complete and fully integrated software solution. It tests if the system meets the specified requirements and if it is suitable for delivery to the end-users. This

type of testing is performed after the integration testing and before the acceptance testing.

System Testing is a type of [software testing](#) that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements. In system testing, integration testing passed components are taken as input. The goal of integration testing is to detect any irregularity between the units that are integrated together. System testing detects defects within both the integrated units and the whole system. The result of system testing is the observed behavior of a component or a system when it is tested. **System Testing** is carried out on the whole system in the context of either system requirement specifications or functional requirement specifications or in the context of both. System testing tests the design and behavior of the system and also the expectations of the customer. It is performed to test the system beyond the bounds mentioned in the [software requirements specification \(SRS\)](#). System Testing is basically performed by a testing team that is independent of the development team that helps to test the quality of the system impartial. It has both functional and non-functional testing. **System Testing is a black-box testing.** System Testing is performed after the integration testing and before the acceptance testing.



System Testing Process: System Testing is performed in the following steps:

- **Test Environment Setup:** Create testing environment for the better quality testing.
- **Create Test Case:** Generate test case for the testing process.
- **Create Test Data:** Generate the data that is to be tested.
- **Execute Test Case:** After the generation of the test case and the test data, test cases are executed.

Types of System Testing:

- **Performance Testing:** Performance Testing is a type of software testing that is carried out to test the speed, scalability, stability and reliability of the software product or application.
- **Load Testing:** Load Testing is a type of software Testing which is carried out to determine the behavior of a system or software product under extreme load.
- **Stress Testing:** Stress Testing is a type of software testing performed to check the robustness of the system under the varying loads.
- **Scalability Testing:** Scalability Testing is a type of software testing which is carried out to check the performance of a software application or system in terms of its capability to scale up or scale down the number of user request load.
- **Acceptance testing**
- Acceptance testing is formal testing based on user requirements and function processing. It determines whether the software is conforming specified requirements and user requirements or not. It is conducted as a kind of Black Box testing where the number of required users involved testing the acceptance level of the system. It is the fourth and last level of software testing.



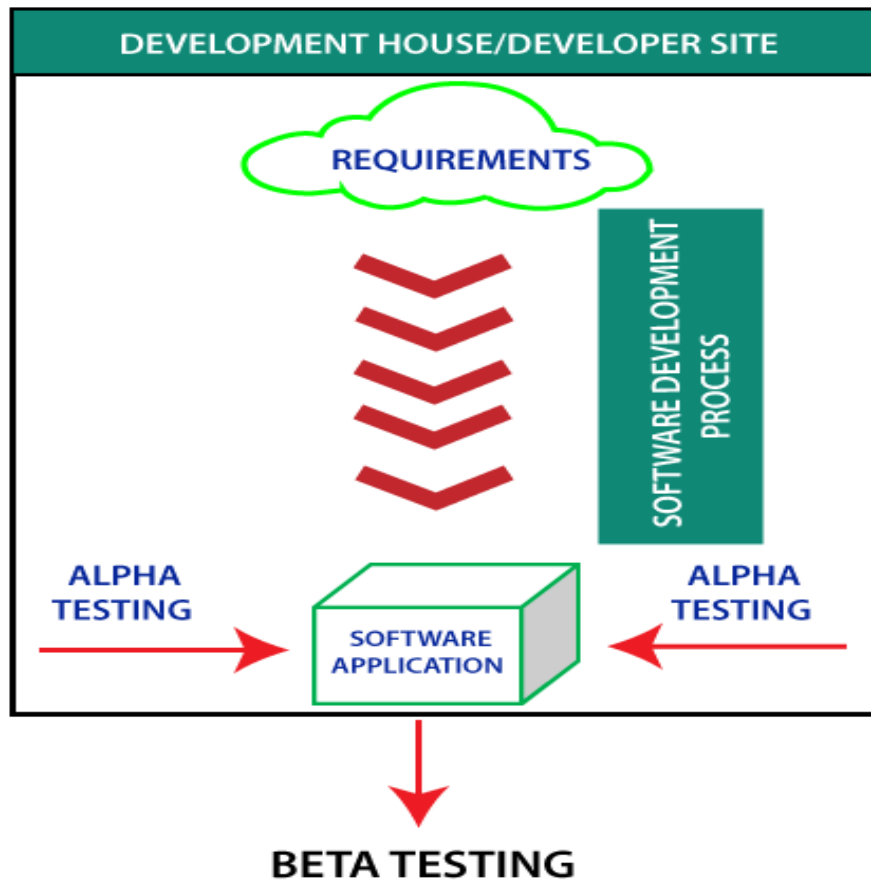
-
- User acceptance testing (UAT) is a type of testing, which is done by the customer before accepting the final product. Generally, UAT is done by the customer (domain expert) for their satisfaction, and check whether the application is working according to given business scenarios, real-time scenarios.
- In this, we concentrate only on those features and scenarios which are regularly used by the customer or mostly user scenarios for the business or those scenarios which are used daily by the end-user or the customer.
- However, the software has passed through three testing levels (Unit Testing, Integration Testing, System Testing) But still there are some minor errors which can be identified when the system is used by the end user in the actual scenario.

• Alpha Testing



- Alpha testing is conducted in the organization and tested by a representative group of end-users at the developer's side and sometimes by an independent team of testers.
- Alpha testing is simulated or real operational testing at an in-house site. It comes after the unit testing, integration testing, etc. Alpha testing used after all the testing are executed.

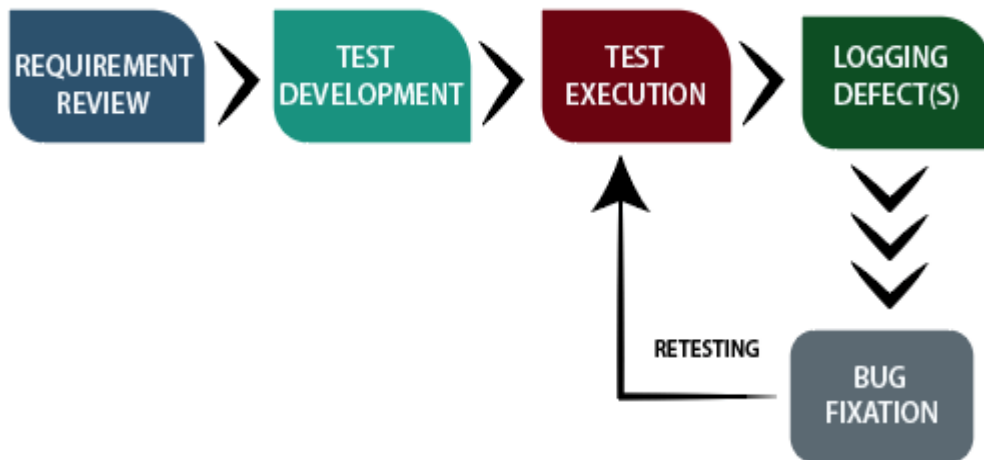
- It can be a white box, or Black-box testing depends on the requirements - particular lab environment and simulation of the actual environment required for this testing.



Alpha testing follows the following process:

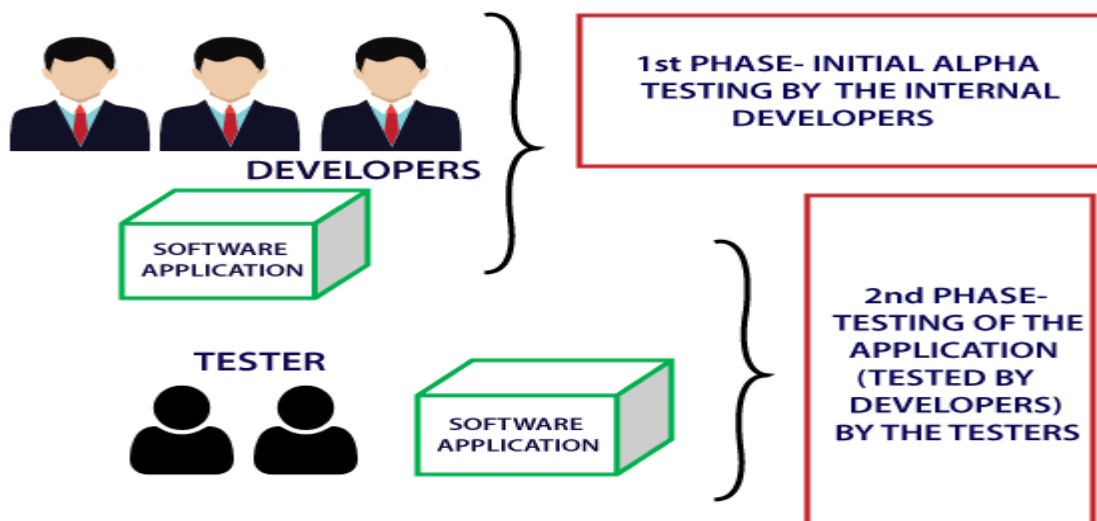
1. **Requirement Review:** Review the design of the specification and functional requirement
2. **Test Development:** Test development is base on the outcome of the requirement review. Develop the test cases and test plan.
3. **Test case design:** Execute the test plan and test cases.
4. **Logging Defects:** Logging the identified and detected bug found in the application.
5. **Bug Fixation:** When all the bugs are identified and logged, then there is a need to fix the bug.
6. **Retesting:** When all the issues are solved, and fixed retesting is done.

ALPHA TESTING



What are the phases of alpha testing?

Alpha testing ensures that the software performs flawlessly and does not impact the reputation of the organization; the company implements final testing in the form of alpha testing. This testing is executed into two phases.



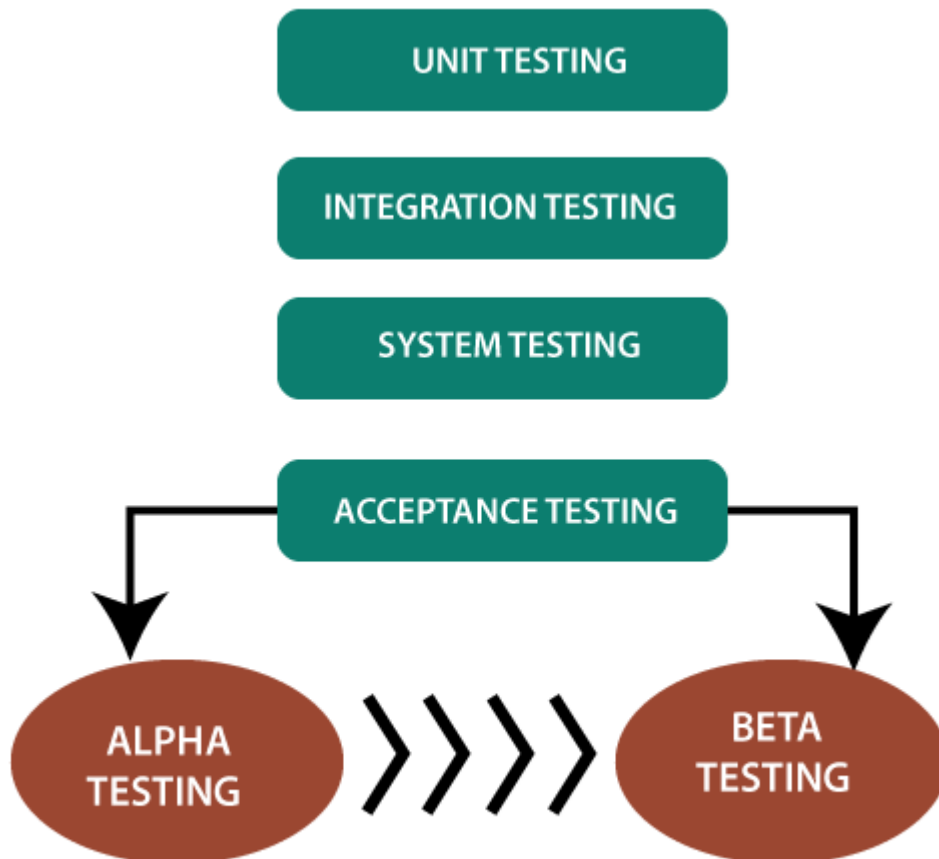
There are two phases of alpha testing.

First Phase: In-house developers or software engineers do the first phase of testing. In this phase, the tester uses a hardware debugger or hardware-aided debugger to catch bugs quickly. During the

alpha testing, a tester finds a lot of bugs, crashes, missing features, and docs.

Second Phase: The second phase involves the quality assurance staff performs the alpha testing by involving black box and white box techniques.

When to perform alpha testing?



Alpha testing is user acceptance testing. Alpha testing performed once the product has gone through stages of testing and prepared for release. It is executing before beta testing, which is also a part of acceptance testing and can define as field testing. During this testing, we can make changes in the software to improve its quality and functionality. Alpha testing done from the developer's site where independent developers can monitor and record user experience and make necessary changes to enhance the performance.

What are the reasons to perform Alpha Testing?

Alpha testing is the final stage of the testing. Alpha testing is an essential and popular testing technique that helps the team to deliver quality and useful software. This testing performed before the release of the product. Alpha testing can define as the first round of independent testing that ensures that the software run as per the requirement plan.

Reasons for alpha testing are:

- Refines the software product by finding and rectifying bugs that weren't discovered through previous tests.
- Alpha testing allows the team to test the software in a real-world environment.
- One of the reasons to do alpha testing is to ensure the success of the software product.
- Alpha testing validates the quality, functionality of the software, and effectiveness of the software before it released in the real world.

Features of Alpha Testing?

- Alpha testing is a type of acceptance testing.
- Alpha testing is happening at the stage of the completion of the software product.
- Alpha testing is in the labs where we provide a specific and controlled environment.
- Alpha testing is in-house testing, which is performed by the internal developers and testers within the organization.
- There is not any involvement of the public.
- Alpha testing helps to gain confidence in the user acceptance of the software product.
- With the help of black box and white box technique, we can achieve the alpha testing.
- Alpha testing ensures the maximum possible quality of the software before releasing it to market or client for beta testing.

- Developers perform alpha testing at developer's site; it enables the developer to record the error with the ease to resolve found bugs quickly.
- Alpha testing is done after the unit testing, integration testing, system testing but before the beta testing.
- Alpha testing is for testing the software application, products, and projects.

What are the advantages of Alpha Testing?

Advantages of alpha testing are:

- One of the benefits of alpha testing is it reduces the delivery time of the project.
- It provides a complete test plan and test cases.
- Free the team member for another project.
- Every feedback helps to improve software quality.
- It provides a better observation of the software's reliability and accountability.

What are the disadvantages of Alpha Testing?

Disadvantages of alpha testing are:

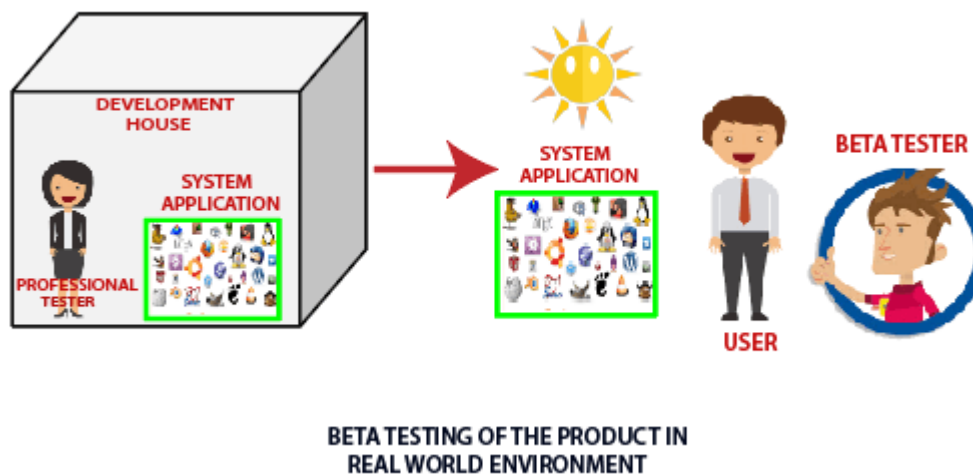
- Alpha testing does not involve in-depth testing of the software.
- The difference between the tester's tests the data for testing the software and the customer's data from their perspective may result in the discrepancy in the software functioning.
- The lab environment is used to simulate the real environment. But still, the lab cannot furnish all the requirements of the real environment such as multiple conditions, factors, and circumstances.

Beta Testing

Beta testing is a type of User Acceptance Testing among the most crucial testing, which is performed before the release of the software.

Beta Testing is a type of Field Test. This testing performs at the end of the *software* testing life cycle. This type of testing can be considered as external user acceptance testing. It is a type of salient testing. Real users perform this testing. This testing executed after the alpha testing. In this the new version, beta testing is released to a limited audience to check the accessibility, usability, and functionality, and more.

Beta testing is the last phase of the testing, which is carried out at the client's or customer's site.



Features of beta testing

Testing of the product performs by the real users of the software application in the real environment. Beta version of the software is released to a restricted number of end-users to obtain the feedback of the product quality. Beta testing reduces the risk of failure and provides the quality of the product through customer validation. It is the final testing before shipping the product to the customers. Beta testing obtains direct feedback from the customers. It helps in testing to test the product in the customer's environment.

Features of beta testing are:

- Beta testing used in a real environment at the user's site. Beta testing helps in providing the actual position of the quality.
- Testing performed by the client, stakeholder, and end-user.

- Beta testing always is done after the alpha testing, and before releasing it into the market.
- Beta testing is black-box testing.
- Beta testing performs in the absence of tester and the presence of real users
- Beta testing is performed after alpha testing and before the release of the final product.
- Beta testing generally is done for testing software products like utilities, operating systems, and applications, etc.

What is a beta version of the software?

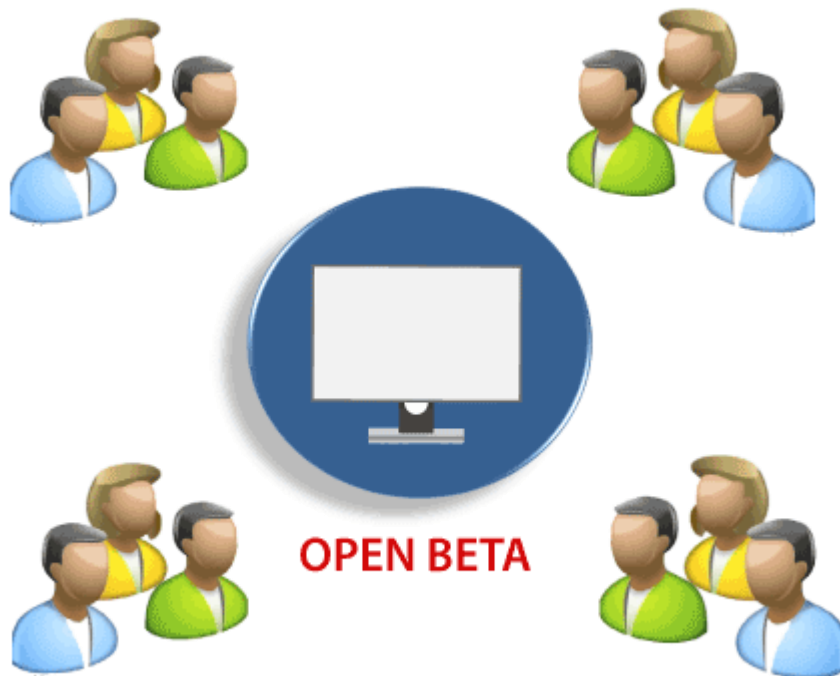
The beta version of the software is delivered to a restricted number of users to accept their feedback and suggestions on quality improvement. Hence, there are two types of beta version:

1) Closed beta version: Closed beta version, also known as a private beta, it is released to a group of selected and invited people. Those people will test the software and evaluate their features and specifications. This beta version represents the software which is capable of delivering value, but it is not ready to be used by everyone. Because it shows the issues like lack of documentation or missing vital features.



2) Open beta version: Open beta is also known as a public beta. The open beta opened to the public. Any user as a tester can assess the beta version to provide the relevant feedback and reviews. Open beta

version improves the quality of the final release. This version helps to find the various undetected errors and issues.



The beta testing process orients this beta version.

The process of beta testing follows the following steps:

1. **Planning:** Like another testing process, beta testing also supports proper planning. In this stage, the team prepares a testing strategy and defines the goal of testing. In this case, the team establishes the need of users for testing, duration, and necessary details related to the process.
2. **Participant Recruitment:** This is the second stage of the beta process in which the team recruits a group of selected end-users for testing. This group can change as per the requirement of the organization and the product.
3. **Product Launch:** When a team of users (testers) recruited. The beta version of the product is launched or installed at the client or user side, and users will test the product for quality assurance.

4. **Collect and Evaluate Feedback:** When the testing finished, developers will collect the feedback provided by the testers and evaluate it. In the end, based on the feedback, issues, and bugs are fixed and resolved by the responsible individual team.
5. **Closure:** When all the problems fixed and the organization meets the exit criteria, beta testing achieved, and the rewards offered to the testing team.

Types of beta testing

Beta testing has six types. Each type has different aspects of the software. All these help developers to improve the quality of the software and allow them to deliver a product that offers excellent user experience. Here are the different types of beta testing:

1. **Open Beta Testing:** Open beta testing involves testing the software product by a large number of people before the final release. The organization decides to make a software product open to the public before releasing the product. Open Beta includes the extensive participation of the public to use and evaluate software product accordingly. Users report the bug to the organization, along with a suggestion to improve the quality of the software.
2. **Closed Beta Testing:** Opposite to the open beta testing. Closed beta testing performed by the selective and limited number of persons. The organization recruits these. In this testing software product is not open to the public.
3. **Traditional Beta Testing:** In this testing, a software product delivered to the target market, and the feedback from the users collected. This type of testing assistance the beta testing, the quality of the software is improved, and developers can make the changes.
4. **Public Beta Testing:** This type of testing is similar to open testing. Public beta testing also allows the product is delivering to the end-users worldwide, with the aid of various online channels available in the world. From this, the feedback and evaluated data

also collected and based on the requirement changes, and the development team implements modifications.

5. **Technical Beta Testing:** Technical beta testing is also an essential type of beta testing. This testing involves delivering the software product to the internal groups of the organization. However, the data and feedback provided by the employees of the organization.
6. **Focused Beta Testing:** This type of testing focused on monitoring and evaluating a specific feature or component of the software. In focused beta testing, the software released to the market and user's experience assessed and collected to make the required changes.
7. **Post-Release Beta Testing:** In this testing, the product delivered to the market for the use of the end-users. Their feedback, reactions, and experience are collect for the future release of the software.

Acceptance testing is the final phase of the testing, which combines both alpha and beta testing to ensure that the product released flawlessly. Beta testing performed at the user's end. This testing always performed after the alpha testing, but before the product released to the market. In this stage, the product is expected to be 90% to 95% completed.

Any product undergoing to beta test should be reviewed for the entire checklist before launching it.

Some of them are:

- All the component of the product is ready to start this testing.
- Documentation which is going to end-user should be kept ready - Setup, installation, usage, Uninstallation should be in detail.
- The product management team should review that all the functionality is in good condition.
- Procedure to collect bugs, feedback, etc. should be identified before publishing it.

Performance Testing

Performance Testing is a type of software testing that ensures software applications perform properly under their expected workload. It is a testing technique carried out to determine system performance in terms of sensitivity, reactivity, and stability under a particular workload.

Performance testing is a type of software testing that focuses on evaluating the performance and scalability of a system or application. The goal of performance testing is to identify bottlenecks, measure system performance under various loads and conditions, and ensure that the system can handle the expected number of users or transactions.

Regression Testing

Regression testing is a black box testing techniques. It is used to authenticate a code change in the software does not impact the existing functionality of the product. Regression testing is making sure that the product works fine with new functionality, **bug** fixes, or any change in the existing feature. Regression testing is a type of **software testing**. Test cases are re-executed to check the previous functionality of the application is working fine, and the new changes have not produced any bugs.

Regression testing can be performed on a new build when there is a significant change in the original functionality. It ensures that the code still works even when the changes are occurring. Regression means Re-test those parts of the application, which are unchanged. Regression tests are also known as the Verification Method. Test cases are often automated. **Test cases** are required to execute many times and running the same test case again and again manually, is time-consuming and tedious too.

Load/Stress Testing

Stress Testing is a type of software testing that verifies stability & reliability of software application. The goal of Stress testing is measuring software on its robustness and error handling capabilities under extremely heavy load conditions and ensuring that software

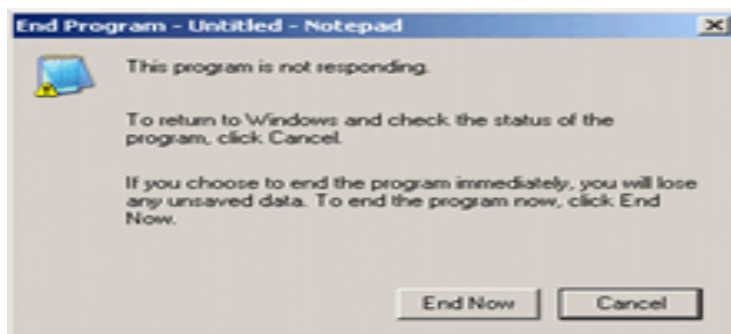
doesn't crash under crunch situations. It even tests beyond normal operating points and evaluates how software works under extreme conditions.

***That is some
Stress !!!***



Stress Testing is also known as [Endurance Testing](#). Under Stress Testing, AUT is be stressed for a short period of time to know its withstanding capacity. A most prominent use **of stress testing is to determine the limit, at which the system or software or hardware breaks**. It also checks whether the system demonstrates effective error management under extreme conditions.

The application under testing will be stressed when 5GB data is copied from the website and pasted in notepad. Notepad is under stress and gives 'Not Responded' error message.



Security Testing

Security testing is an integral part of software testing, which is used to discover the weaknesses, risks, or threats in the software application and also help us to stop the nasty attack from the outsiders and make sure the security of our software applications.

The primary objective of security testing is to find all the potential ambiguities and vulnerabilities of the application so that the software

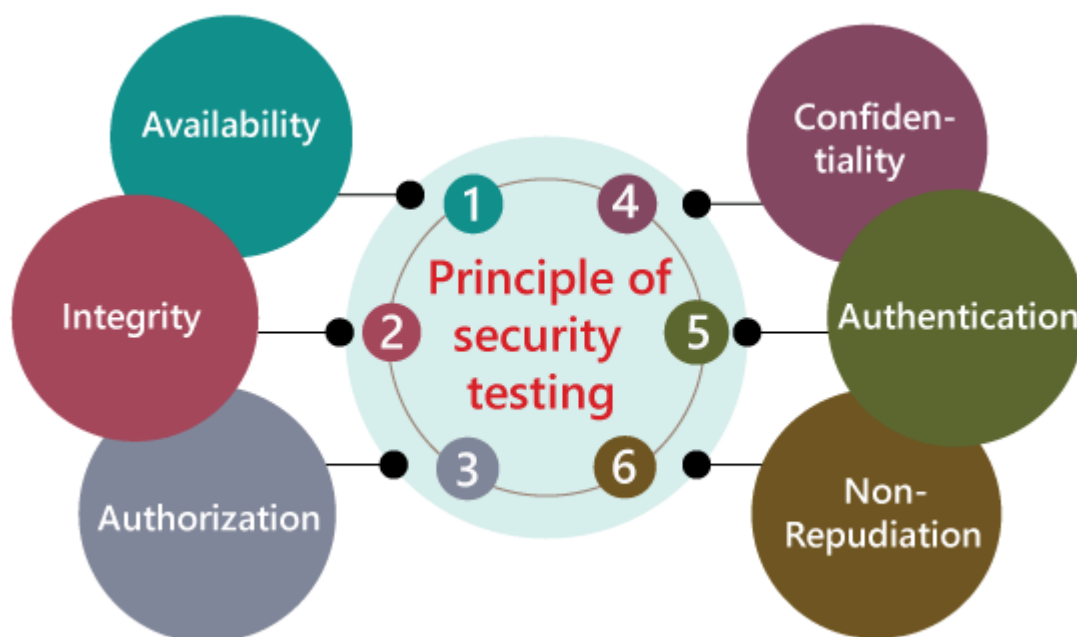
does not stop working. If we perform security testing, then it helps us to identify all the possible security threats and also help the programmer to fix those errors.

It is a testing procedure, which is used to define that the data will be safe and also continue the working process of the software.

Principle of Security testing

Here, we will discuss the following aspects of security testing:

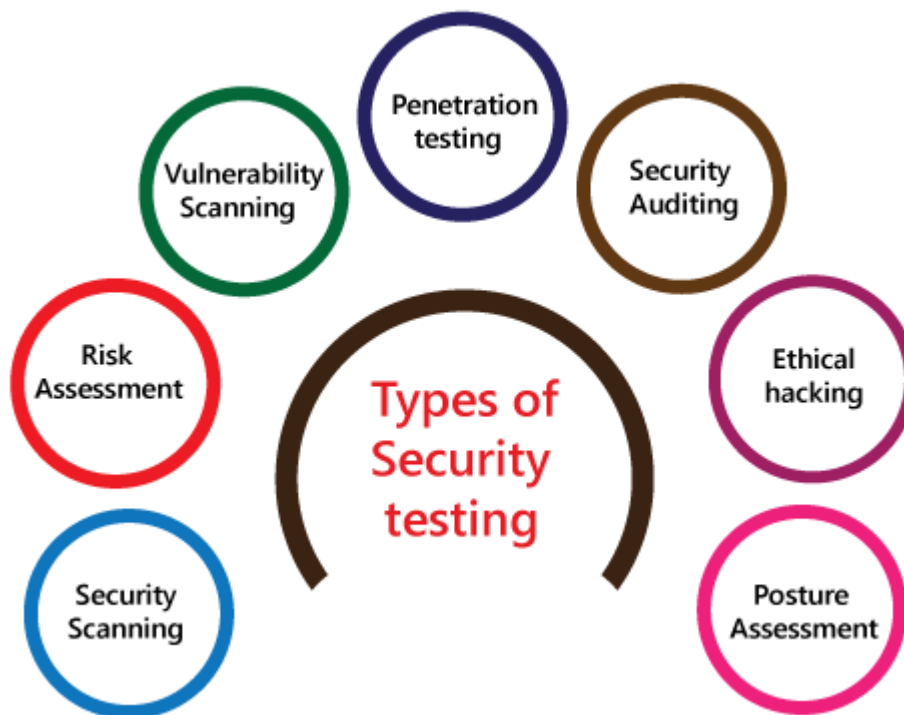
- Availability
- Integrity
- Authorization
- Confidentiality
- Authentication
- Non-repudiation



Types of Security testing

As per Open Source Security Testing techniques, we have different types of security testing which as follows:

- **Security Scanning**
- **Risk Assessment**
- **Vulnerability Scanning**
- **Penetration testing**
- **Security Auditing**
- **Ethical hacking**
- **Posture Assessment**



Security Scanning

Security scanning can be done for both [automation testing](#) and [manual testing](#). This scanning will be used to find the vulnerability or unwanted file modification in a web-based application, websites, network, or the file system. After that, it will deliver the results which help us to decrease those threats. Security scanning is needed for those systems, which depends on the structure they use.

Risk Assessment

To moderate the risk of an application, we will go for risk assessment. In this, we will explore the security risk, which can be detected in the association. The risk can be further divided into three parts, and those

are **high, medium, and low**. The primary purpose of the risk assessment process is to assess the vulnerabilities and control the significant threat.

Vulnerability Scanning

It is an application that is used to determine and generates a list of all the systems which contain the desktops, servers, laptops, virtual machines, printers, switches, and firewalls related to a network. The vulnerability scanning can be performed over the automated application and also identifies those software and systems which have acknowledged the security vulnerabilities.

Penetration testing

Penetration testing is a security implementation where a cyber-security professional tries to identify and exploit the weakness in the computer system. The primary objective of this testing is to simulate outbreaks and also finds the loophole in the system and similarly save from the intruders who can take the benefits.

Security Auditing

Security auditing is a structured method for evaluating the security measures of the organization. In this, we will do the inside review of the application and the control system for the security faults.

Ethical hacking

Ethical hacking is used to discover the weakness in the system and also helps the organization to fix those security loopholes before the nasty hacker exposes them. The ethical hacking will help us to increase the security position of the association because sometimes the ethical hackers use the same tricks, tools, and techniques that nasty hackers will use, but with the approval of the official person.

The objective of ethical hacking is to enhance security and to protect the systems from malicious users' attacks.

Posture Assessment

It is a combination of **ethical hacking, risk assessments, and security scanning**, which helps us to display the complete security posture of an organization.

Internationalization Testing

Software testing is an important part of the software development life cycle. There are different types of software testing are performed during the development of a software product/service. Software testing ensures that our developed software product/service is bug-free and delivers fulfilling the desired requirements.

The most important factor is when an IT company develops a software product/service beyond any specific region then the company has to perform a test to check if the product will work in different regions of the world or not as it will be used by different cultural people. It is also shortly known as i18n, in which 18 represents the number of characters in between I & N in the word Internationalization. Internationalization simply makes applications ready for localization.

Just imagine, your native language is Hindi and you are more comfortable with it than English. And you are opening the Amazon application for buying a brand new mobile. There you select Marathi as your preferred language since you are comfortable with it the most. Then the content and user interface will be adapted to the language “Marathi”.

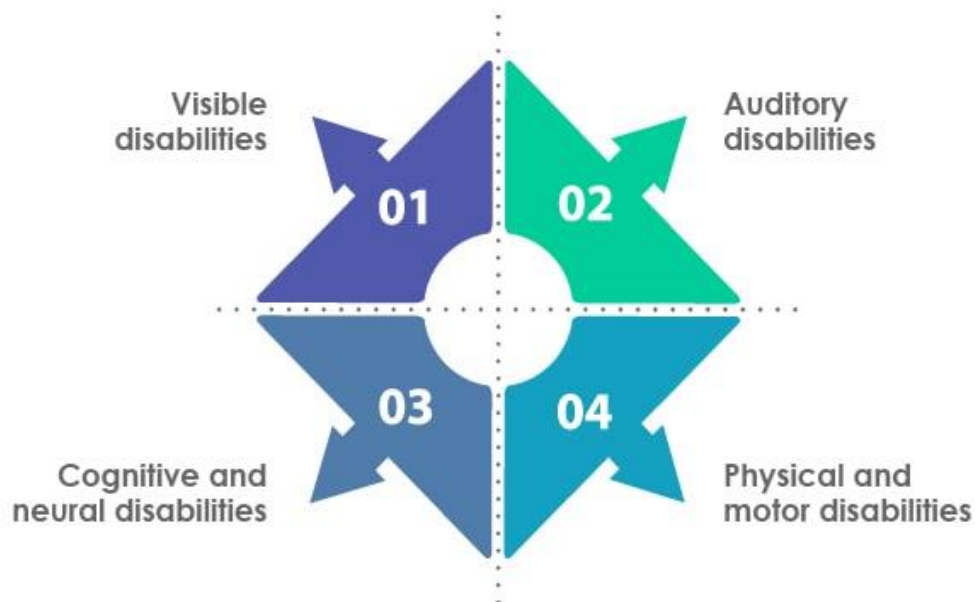
After that, the functionality and the responses of the application are not going to change. The words and visual representations are customized to your language. Along with that, you will get recommendations depending on special occasions and specific festivals in your culture and region. This customization for a specific language and region is made possible with the process of localization.

Likewise, the Amazon application now supports several languages including seven popular Indian languages. If you prefer any of the languages, the entire page will be customized in just seconds based on the language selected. This process of designing an application

for the localization to any given international language and region is called Internationalization.

Usability and Accessibility Testing

Accessibility testing is a software testing technique performed to ensure that web applications or mobile applications are accessible to everyone, including the different abled sections of society. This software testing technique verifies that an app is developed according to [web accessibility standards](#) to ensure equal access to all people, including people with specific disabilities.



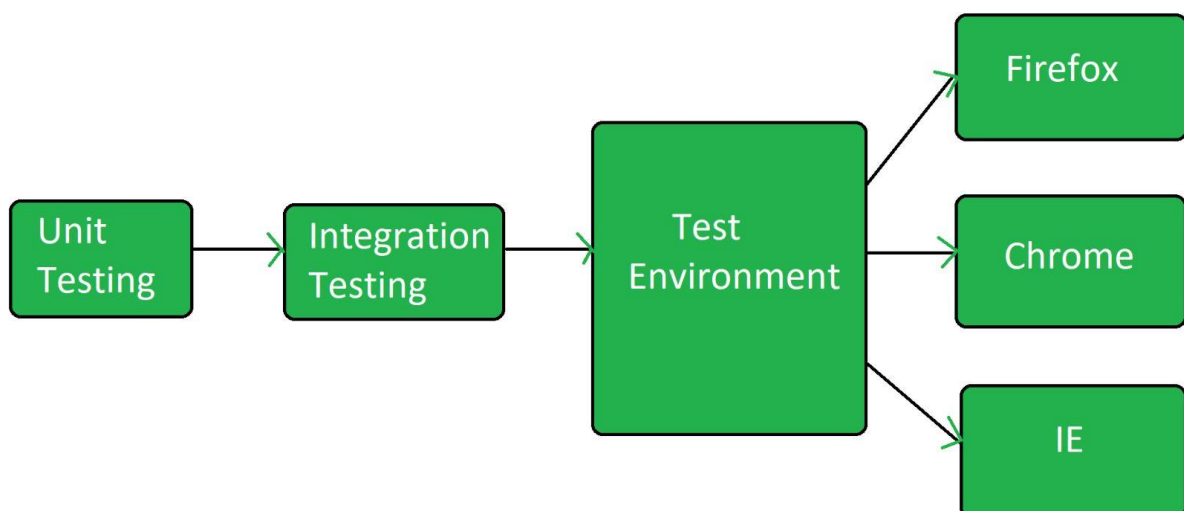
Configuration Testing

Configuration Testing is a software testing technique in which the software application is tested with multiple combinations of software and hardware in order to evaluate the functional requirements and find out optimal configurations under which the software application works without any defects or flaws.

The objectives of configuration Testing

- Validating the application to determine if it fulfills the configurability requirements

- Manually causing failures which help in identifying the defects that are not efficiently found during testing (Ex: changing the regional settings of the system like Time Zone, Language, Date time formats, etc.)
- Determine an optimal configuration of the application under test.
- Analyzing the system performance by adding or modifying the hardware resources like Load Balancers, increase or decrease in memory size, connecting various printer models, etc.
- Analyzing system Efficiency based on the prioritization, how efficiently the tests were performed with the resources available to achieve the optimal system configuration.
- Verification of the system in a geographically distributed Environment to verify how effectively the system performs. For Ex: Server at a different location and clients at a different location, the system should work fine irrespective of the system settings.
- Verifying how easily the bugs are reproducible irrespective of the configuration changes.
- Ensuring how traceable the application items are by properly documenting and maintaining the versions which are easily identifiable.
- Verifying how manageable the application items are throughout the [software development life cycle](#).



Types of Configuration Testing:
Configuration testing is of 2 types:

1. Software Configuration Testing: Software configuration testing is done over the Application Under Test with various operating system versions and various browser versions etc. It is a time-consuming testing as it takes long time to install and uninstall the various software which are to be used for testing. When the build is released, software configuration begins after passing through the unit test and integration test.

2. Hardware Configuration Testing: Hardware configuration testing is typically performed in labs where physical machines are used with various hardware connected to them. When a build is released, the software is installed in all the physical machines to which the hardware is attached and the test is carried out on each and every machine to confirm that the application is working fine. While doing hardware configuration test, the kind of hardware to be tested is spelled out and there are several computer hardware and peripherals which make it next to impossible to execute all the tests. Configuration Testing can also be classified into following 2 types:

1. **Client level testing:** Client level testing is associated with the usability and functionality testing. This testing is done from the point of view of its direct interest of the users.
2. **Server level Testing:** Server level testing is carried out to determine the communication between the software and the external environment when it is planned to be integrated after the release.

Compatibility Testing

Compatibility testing :

Compatibility testing is software testing which comes under the [non functional testing](#) category, and it is performed on an application to check its compatibility (running capability) on different platform/environments. This testing is done only when the application becomes stable. Means simply this compatibility test aims to check the developed software application functionality on various software, hardware platforms, network and browser etc. This compatibility testing is very important in product production and implementation point of view as it is performed to avoid future issues regarding compatibility.

Types of Compatibility Testing :

Several examples of compatibility testing are given below.

1. Software :

- Testing the compatibility of an application with an Operating System like Linux, Mac, Windows
- Testing compatibility on Database like Oracle SQL server, MongoDB server.
- Testing compatibility on different devices like in mobile phones, computers.

Types based on Version Testing :

There are two types of compatibility testing based on version testing

1. **Forward compatibility testing :** When the behavior and compatibility of a software or hardware is checked with its newer version then it is called as forward compatibility testing.
2. **Backward compatibility testing :** When the behavior and compatibility of a software or hardware is checked with its older version then it is called as backward compatibility testing.

2. Hardware :

Checking compatibility with a particular size of

- RAM
- ROM
- Hard Disk
- Memory Cards
- Processor
- Graphics Card

3. Smartphones :

Checking compatibility with different mobile platforms like android, iOS etc.

4. Network :

Checking compatibility with different :

- Bandwidth
- Operating speed
- Capacity

Along with this there are other types of compatibility testing are also performed such as browser compatibility to check software compatibility with different browsers like Google Chrome, Internet Explorer etc. device compatibility, version of the software and others.

How to perform Compatibility testing ?

Testing the application in a same environment but having different versions. For example, to test compatibility of Facebook application in your android mobile. First check for the compatibility with Android 9.0 and then with Android 10.0 for the same version of Facebook App.

Testing the application in a same versions but having different environment. For example, to test compatibility of Facebook application in your android mobile. First check for the compatibility with a Facebook application of lower version with a Android 10.0(or your choice) and then with a Facebook application of higher version with a same version of Android.

Advantages of Compatibility Testing :

1. It ensures complete customer satisfaction.
2. It provides service across multiple platforms.
3. Identifying bugs during development process.

Disadvantages Compatibility testing :

1. Variety of user interface.
2. Changes with respect to font size.
3. Alignment issues.
4. Issues related to existence of broken frames.
5. Issues related to overlapping of content.



