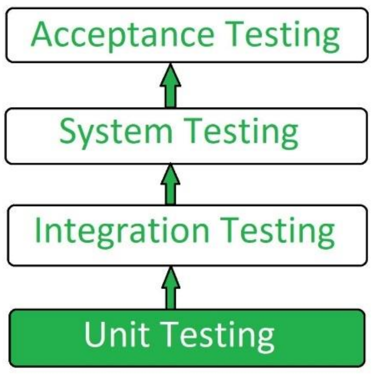
Software Testing

Chapter 3: Levels of software testing

Unit Testing:

- Unit testing involves the testing of each unit or an individual component of the software application.
- It is the first level of functional testing.
- The aim behind unit testing is to validate unit components with its performance.
- In testing hierarchy, Unit testing is the first level of testing done before integration testing.
- Unit testing is such a type of testing technique that is usually performed by developers.
- Unit testing tools: Jtest, Junit, NUnit, EMMA, PHPUnit, etc.
- There are 2 types of Unit Testing: Manual, and Automated.
- Unit tests are automated and are run each time the code is changed to ensure that new code does not break existing functionality.
- Unit tests are designed to validate the smallest possible unit of code, such as a function or a method, and test it in isolation from the rest of the system.



The objective of Unit Testing is:

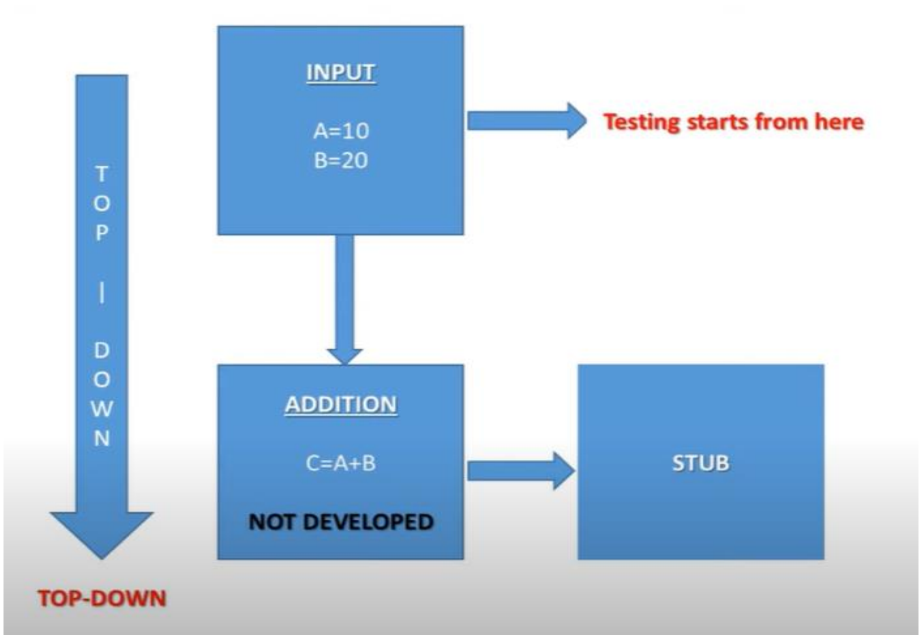
- To isolate a section of code.
- To verify the correctness of the code.
- To test every function and procedure.
- To fix bugs early in the development cycle and to save costs.
- To help with code reuse.

Stubs and Drivers:

- The Stubs and Drivers are considered as elements which are equivalent to modules that could be replaced if modules are in their developing stage, missing or not developed yet, so that necessity of such modules could be met.
- Drivers and stubs simulate features and functionalities, and have ability to serve features that a module can provide.
- This reduces useless delay in testing and makes the testing process faster.
- Stubs are mainly used in Top-Down integration testing while the Drivers are used in Bottom-up integration testing, thus increasing the efficiency of testing process.

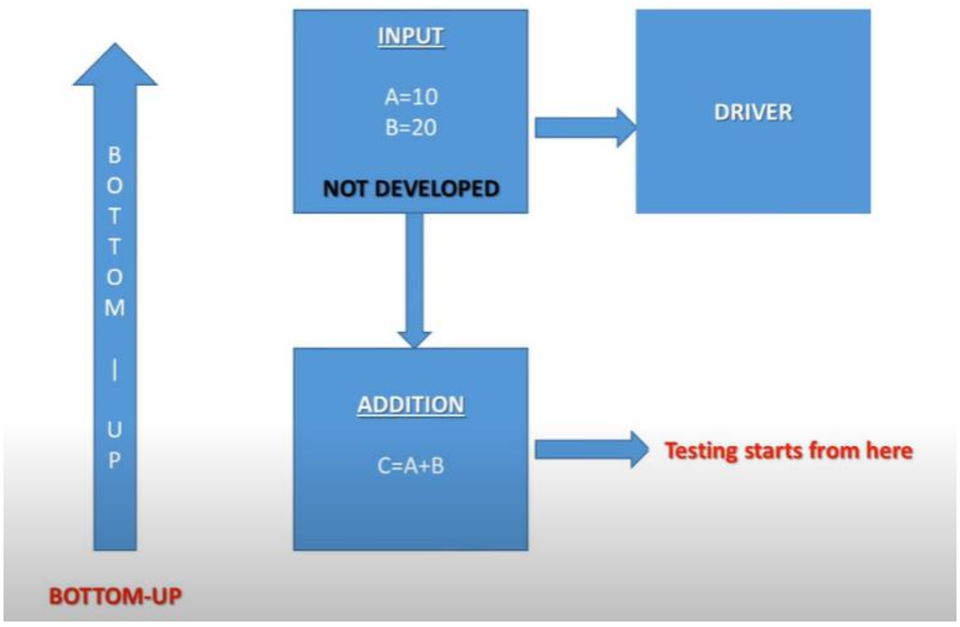
Stubs:

- Stubs are developed by software developers to use them in place of modules, if the respective modules aren't developed, missing in developing stage, or are unavailable currently while Top-down testing of modules.
- A Stub simulates module which has all the capabilities of the unavailable module.
- Stubs are used when the lower-level modules are needed but are unavailable currently.
- However, it executes like an actual module and is mainly used to test modules.



Drivers:

- Drivers serve the same purpose as stubs, but drivers are used in Bottom-up integration testing.
- Drivers are also used when some modules are missing and unavailable at time of testing of a specific module because of some unavoidable reasons, to act in absence of required module.
- Drivers are used when high-level modules are missing and can also be used when lower-level modules are missing.
- Generally, drivers are bit complex as compared to the stubs.



Ex : Suppose, you are told to test a website whose corresponding primary modules are, where each of them is interdependent on each other, as follows:

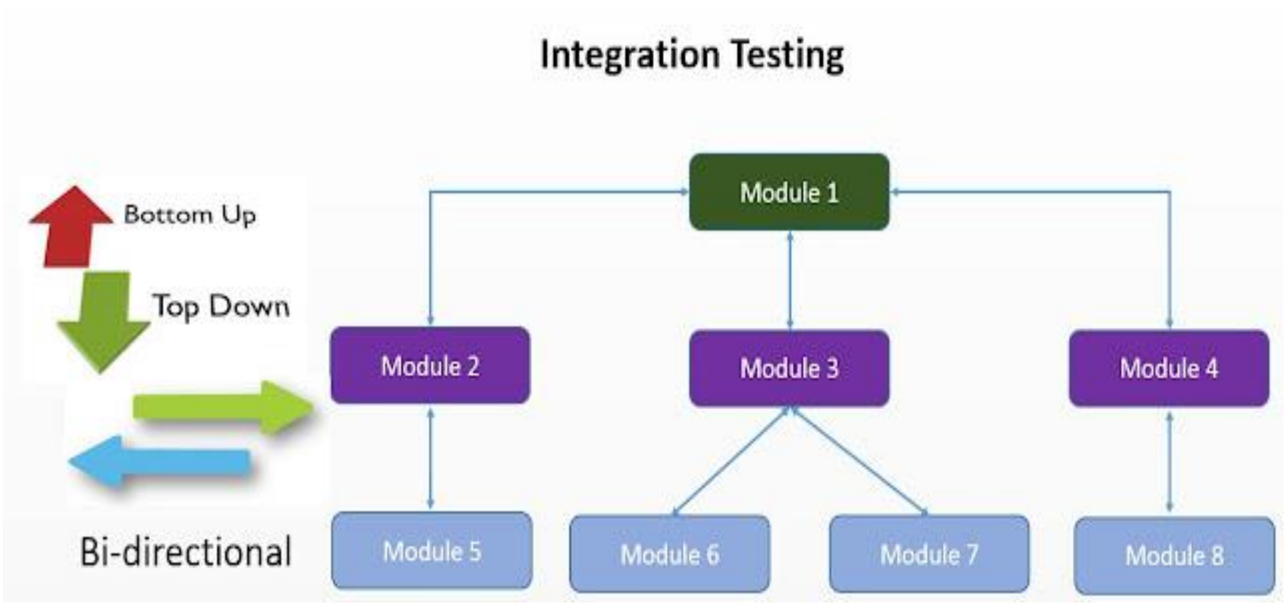
- ✓ Module-A : Login page website,
- ✓ Module-B : Home page of the website
- ✓ Module-C : Profile setting
- ✓ Module-D : Sign-out page

- Assume Module-A is developed. As soon as it's developed, it undergoes testing, but it requires Module-B, which isn't developed yet.
- So in this case, we can use the Stubs or Drivers that simulate all features and functionality that might be shown by actual Module-B. So, we can conclude that Stubs and drivers are used to fulfill the necessity of unavailable modules.
- Similarly, we may also use Stubs or Drivers in place of Module-C and Module-D if they are too not available.

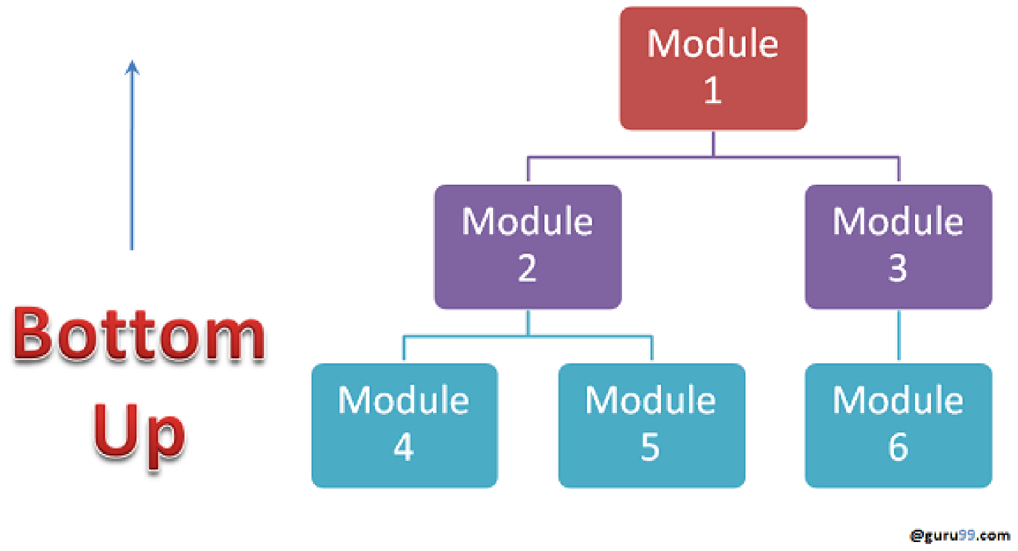
Sr No.	Stubs	Drivers
1.	Stubs are used in Top-Down Integration Testing.	Drivers are used in Bottom-Up Integration Testing.
2.	Stubs are basically known as a "called programs" and are used in the Top-down integration testing.	While, drivers are the "calling program" and are used in bottom-up integration testing.
3.	Stubs are similar to the modules of the software, that are under development process.	While drivers are used to invoking the component that needs to be tested.
4.	Stubs are basically used in the unavailability of low-level modules.	While drivers are mainly used in place of high-level modules and in some situation as well as for low-level modules.
5.	Stubs are taken into use to test the feature and functionality of the modules.	Whereas the drivers are used if the main module of the software isn't developed for testing
6.	The stubs are taken into concern if testing of upper-levels of the modules are done and the lower-levels of the modules are under developing process.	The drivers are taken into concern if testing of lower-levels of the modules are done and the upper-levels of the modules are under developing process.
7.	Stubs are used when lower-level of modules are missing or in a partially developed phase, and we want to test the main module.	Drivers are used when higher-level of modules are missing or in a partially developed phase, and we want to test the lower(sub)- module.

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.
- 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.



Bottom up Integration testing:



- In bottom-up testing, each module at lower levels are tested with higher modules until all modules are tested.
- The primary purpose of this integration testing is that each subsystem tests the interfaces among various modules making up the subsystem.
- This integration testing uses test drivers to drive and pass appropriate data to the lower-level modules.

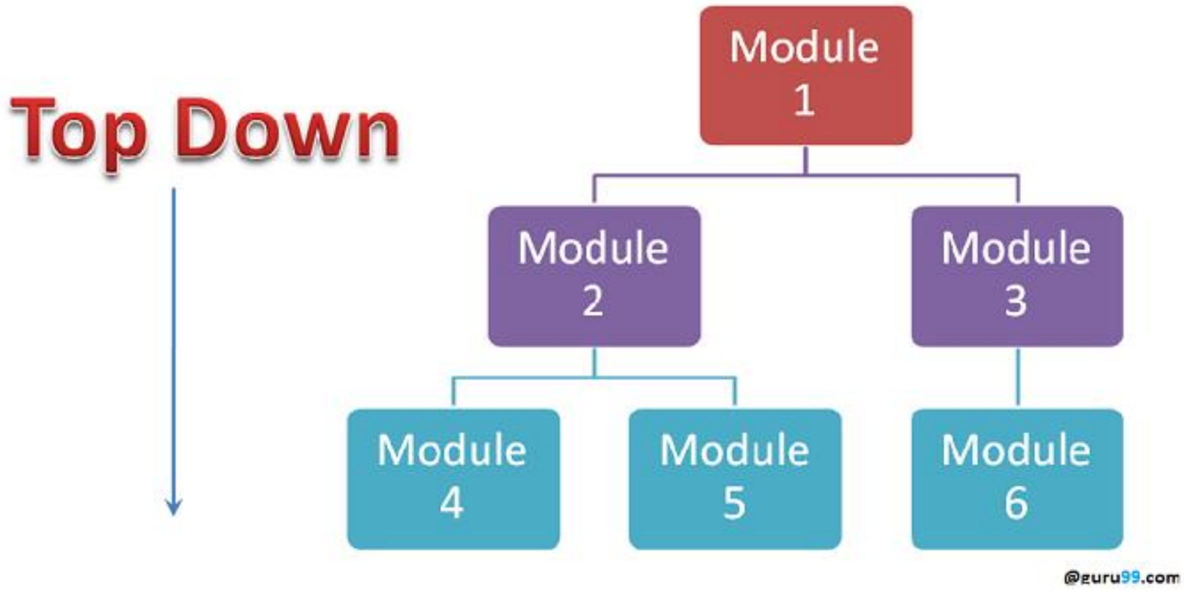
Advantages:

- In bottom-up testing, no stubs are required.
- A principal advantage of this integration testing is that several disjoint subsystems can be tested simultaneously.
- It is easy to create the test conditions.
- Best for applications that uses bottom up design approach.
- It is Easy to observe the test results.

Disadvantages:

- Driver modules must be produced.
- In this testing, the complexity that occurs when the system is made up of a large number of small subsystems.
- As Far modules have been created, there is no working model can be represented.

3. Top-Down Integration Testing



- Top-down integration testing technique is used in order to simulate the behaviour of the lower-level modules that are not yet integrated.
- In this integration testing, testing takes place from top to bottom.
- First, high-level modules are tested and then low-level modules and finally integrating the low-level modules to a high level to ensure the system is working as intended.

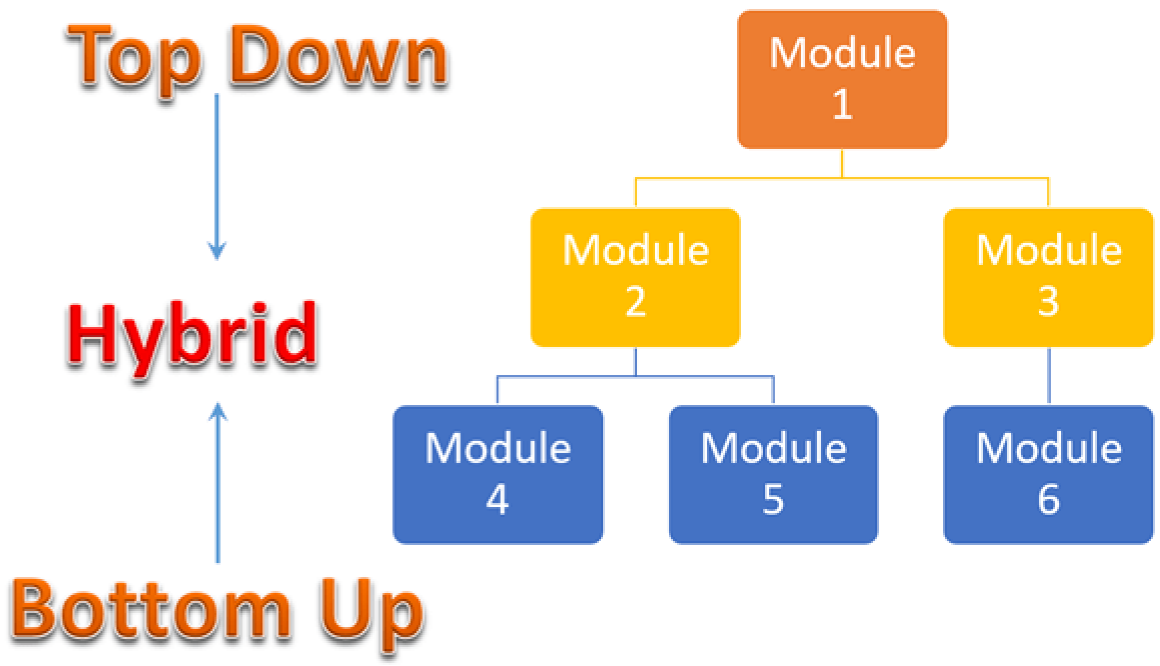
Advantages:

- Separately debugged module.
- Few or no drivers needed.
- It is more stable and accurate at the aggregate level.
- Easier isolation of interface errors.
- In this, design defects can be found in the early stages.

Disadvantages:

- Needs many Stubs.
- Modules at lower level are tested inadequately.
- It is difficult to observe the test output.
- It is difficult to stub design.

4. Bi-directional Integration Testing –



- A Bidirectional integration testing is also called sandwiched integration testing.
- A mixed integration testing follows a combination of top down and bottom-up testing approaches.
- The top-level modules are tested with low-level modules and the low-level modules are tested with high-level modules simultaneously.
- In top-down approach, testing can start only after the top-level module have been coded and unit tested. In bottom-up approach, testing can start only after the bottom level modules are ready.
- This sandwich or mixed approach overcomes this shortcoming of the top-down and bottom-up approaches.
- It is also called the hybrid integration testing. also, stubs and drivers are used in bidirectional/mixed integration testing.

OR

- Bidirectional means that the data is transferred between system A and system B, i.e. in both directions. For example, you can seamlessly continue your process that starts in system A in real time in system B and run it back to system A again.

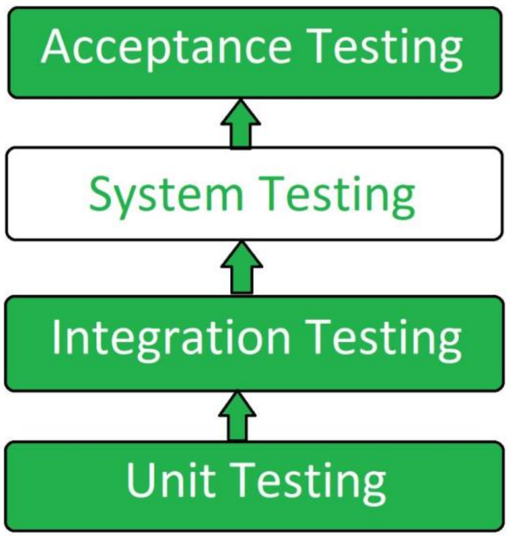
Advantages:

- Mixed approach is useful for very large projects having several sub projects.
- This Sandwich approach overcomes this shortcoming of the top-down and bottom-up approaches.
- Parallel test can be performed in top and bottom layer tests.

Disadvantages:

- For mixed integration testing, it requires very high cost because one part has a Top-down approach while another part has a bottom-up approach.
- This integration testing cannot be used for smaller systems with huge interdependence between different modules.

System Testing:



- It is a level of testing that validates the complete and fully integrated software product.
- The purpose of a system test is to evaluate the end-to-end system specifications.
- Usually, the software is only one element of a larger computer-based system.
- Ultimately, Ultimately, the software is interfaced with other software/hardware systems.
- System Testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system.
- System testing done by a professional testing agent on the completed software product before it is introduced to the market.
- **System Testing is a black-box testing**. System Testing is performed after the integration testing and before the acceptance testing.

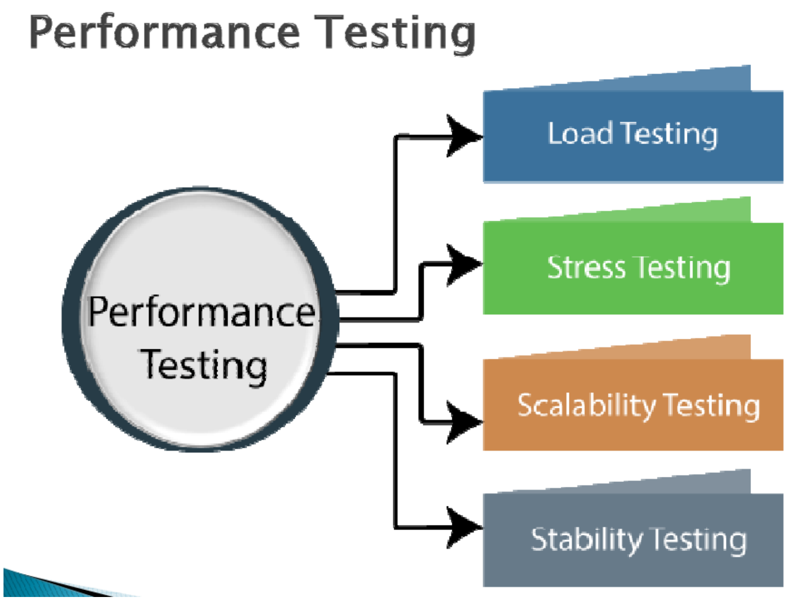
1.Recovery Testing:

- In software testing, recovery testing is the activity of testing how well an application is able to recover from crashes, hardware failures and other similar problems.
- Recovery testing Recovery testing is the forced failure of the software in a variety of ways to verify that recovery is properly performed.
- The purpose of Recovery Testing is to determine whether software operations can be continued after disaster or integrity loss.
- Recovery testing involves reverting back software to the point where integrity was known and reprocessing transactions to the failure point.

2.Security Testing:

- SECURITY TESTING is a type of Software Testing that uncovers vulnerabilities, threats, risks in a software application and prevents malicious attacks from intruders.
- The purpose of Security Tests is to identify all possible loopholes and weaknesses of the software system which might result in a loss of information, revenue, repute at the hands of the employees or outsiders of the Organization.
- It also aims at verifying 6 basic principles as listed below:
    1. Confidentiality
    2. Integrity
    3. Authentication
    4. Authorization
    5. Availability
    6. Non-repudiation

3.Performance Testing



- Performance Testing is a software testing process used for testing the speed, response time, stability, reliability, scalability and resource usage of a software application under particular workload.
- The main purpose of performance testing is to identify and eliminate the performance bottlenecks in the software application. It is a subset of performance engineering and also known as “Perf Testing”.

The focus of Performance Testing is checking a software program's

1. Speed - Determines whether the application responds quickly
2. Scalability - Determines maximum user load the software application can handle.
3. Stability - Determines if the application is stable under varying loads

Load Testing:

- Load Testing is a non-functional software testing process in which the performance of software application is tested under a specific expected load.
- It determines how the software application behaves while being accessed by multiple users simultaneously.
- The goal of Load Testing is to improve performance bottlenecks and to ensure stability and smooth functioning of software application before deployment.
- This testing usually identifies –
- The maximum operating capacity of an application
- Determine whether the current infrastructure is sufficient to run the application
- Sustainability of application with respect to peak user load
- No. of concurrent users that an application can support, and scalability to allow more users to access it.

Stress testing:

- A stress test is a type of performance test that checks the upper limits of your system by testing it under extreme loads.
- Stress tests examine how the system behaves under intense loads and how it recovers when going back to normal usage.
- Stress tests also look for memory leaks, slowdowns, security issues, and data corruption.
- Stress testing can be conducted through load testing tools by defining a test case with a very high number of concurrent virtual users.

Usability Testing:

Usability testing, a non-functional testing technique that is a measure of how easily the system can be used by end users. It is difficult to evaluate and measure but can be evaluated based on the below parameters:

- Level of Skill required to learn/use the software. It should maintain the balance for both novice and expert user.
- Time required to get used to in using the software.
- The measure of increase in user productivity if any.
- Assessment of a user's attitude towards using the software.

UAT(User Acceptance testing):

- User Acceptance Testing (UAT) is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment.
- UAT is done in the final phase of testing after functional, integration and system testing is done.
- User acceptance testing, a testing methodology where the clients/end users involved in testing the product to validate the product against their requirements.
- It is performed at client location at developer's site.

Acceptance Criteria:

Acceptance criteria are defined on the basis of the following attributes:

1. Functional Correctness and Completeness
2. Data Integrity
3. Data Conversion
4. Usability
5. Performance
6. Timeliness
7. Confidentiality and Availability
8. Installability and Upgradability
9. Scalability
10. Documentation

Alpha testing and Beta testing:

- Alpha and Beta testing are the Customer Validation methodologies (Acceptance Testing types) that help in building confidence to launch the product, and thereby results in the success of the product in the market.
- Even though they both rely on real users and different team feedback, they are driven by distinct processes, strategies, and goals.
- These two types of testing together increase the success and lifespan of a product in the market.
- These phases can be adapted to Consumer, Business, or Enterprise products.

Alpha testing:

- This is a form of internal acceptance testing performed mainly by the in-house software QA and testing teams.
- Alpha testing is the last testing done by the test teams at the development site after the acceptance testing and before releasing the software for the beta test.
- Alpha testing can also be done by the potential users or customers of the application. But still, this is a form of in-house acceptance testing.

Beta testing:

- This is a testing stage followed by the internal full alpha test cycle. This is the final testing phase where the companies release the software to a few external user groups outside the company test teams or employees.
- This initial software version is known as the beta version.
- Most companies gather user feedback in this release.
- In short, beta testing can be defined as– the In short, beta testing can be defined as testing carried out by real users in a real environment.

Accessibility Testing:

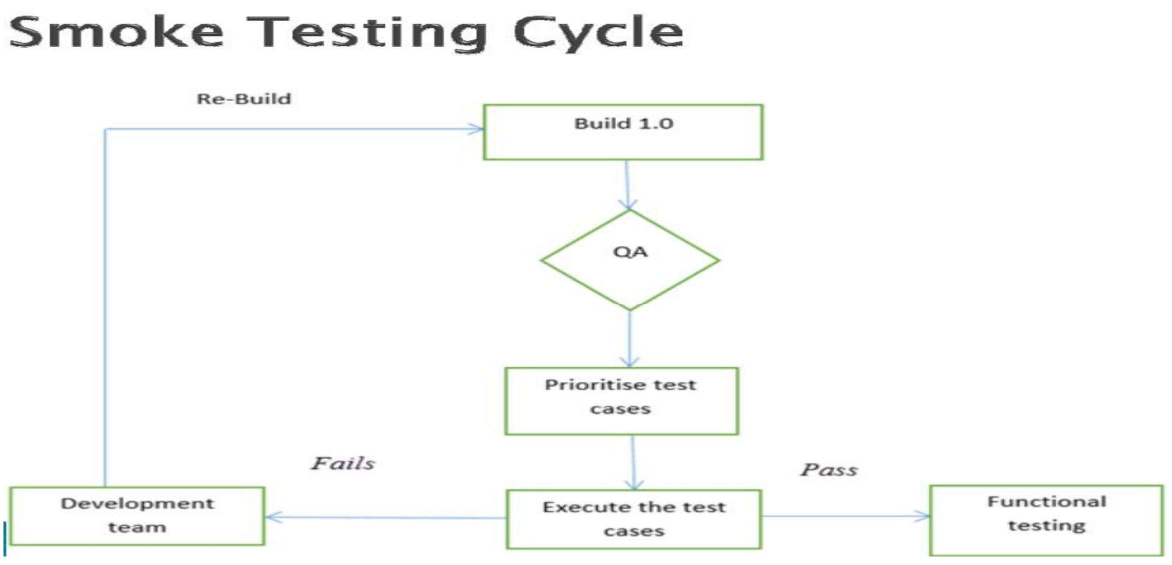
Accessibility Testing is defined as a type of Software Testing performed to ensure that the application being tested is usable by people with disabilities like hearing, colour blindness, old age and other disadvantaged groups. It is a subset of Usability Testing.

People with disabilities use assistive technology which helps them in operating a software product. Examples of such software are:

- Speech Recognition Software - Speech Recognition Software - It will convert the spoken word to text, which serves as input to the computer.
- Screen reader software - Used to read out the Used to read out the text that is displayed on the screen
- Screen Magnification Software- Used to enlarge Screen Magnification Software the monitor and make reading easy for vision impaired users.
- Special keyboard made for the users for easy Special keyboard typing who have motor control difficulties.

Smoke Testing:

- Smoke Testing is a software testing process that determines whether the deployed software build is stable or not.
- Smoke testing is a confirmation for QA team to proceed with further software testing.
- It consists of a minimal set of tests run on each build to test software functionalities.
- Smoke testing is also known as "Build Verification Testing" or “Confidence Testing.”
- It is a simple test that shows the product is ready for testing.
- The main aim of smoke testing is to detect early major issues.



Advantages:

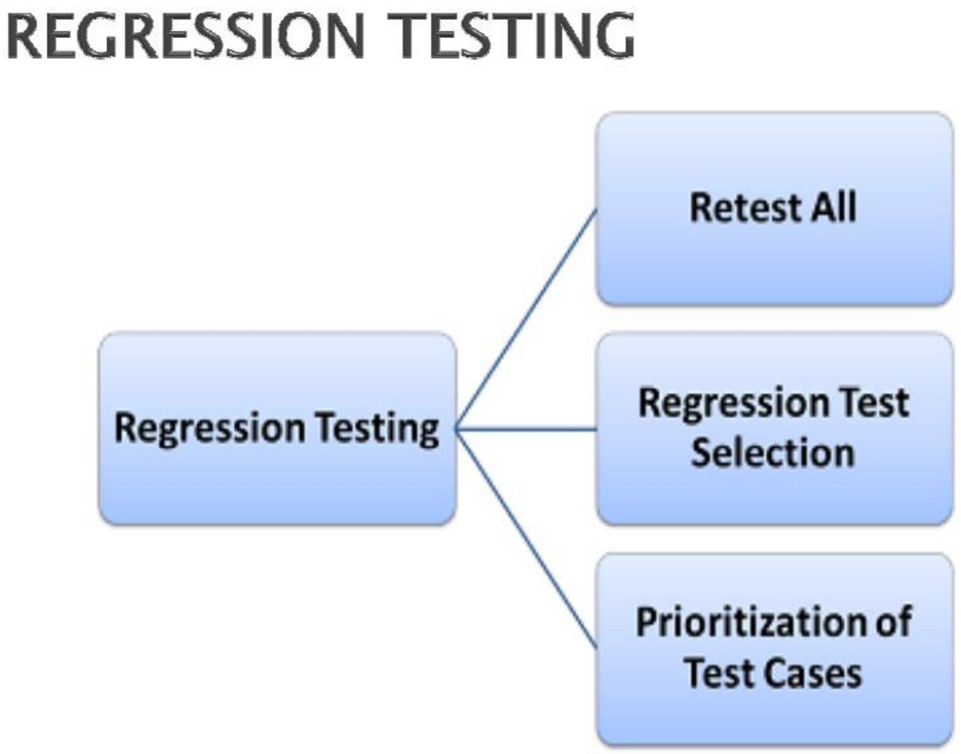
- Smoke testing is easy to perform.
- It helps in identifying defects in early stages.
- It improves the quality of the system.
- Smoke testing reduces the risk of failure.
- Smoke testing makes progress easier to access.
- It saves test effort and time.
- It makes easy to detect critical errors and helps in correction of errors.
- It runs quickly.
- It minimizes integration risks.

Sanity Testing:

- Sanity testing is a subset of regression testing.
- After receiving the software build, sanity testing is performed to ensure that the code changes introduced are working as expected.
- This testing is a checkpoint to determine if testing for the build can proceed or not.
- The main purpose of this testing is to determine that the changes or the proposed functionality are working as expected.
- If the sanity test fails, the build is rejected by the testing team to save time and money.
- It is performed only after the build has cleared the smoke test and been accepted by the Quality Assurance team for further testing. The focus of the team during this testing process is to validate the functionality of the application and not detailed testing.
- For instance, if your scientific calculator gives the result of 2 + 2 =5! Then, there is no point testing the advanced functionalities like sin 30 + cos 50.

- ➢ Both Sanity and Smoke testing are ways to avoid wasting time and effort by quickly determining whether an application is too flawed to merit any rigorous testing.
- ➢ Both smoke and sanity tests can be executed manually or using an automation tool.
- ➢ When automated tools are used, the tests are often initiated by the same process that generates the build itself.
- ➢ As per the needs of testing, you may have to execute both Sanity and Smoke Tests in the software build. In such cases, you will first execute Smoke tests and then go ahead with Sanity Testing.
- ➢ In industry, test cases for Sanity Testing are commonly combined with that for smoke tests, to speed up test execution.

REGRESSION TESTING:

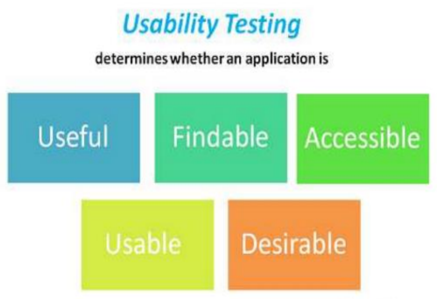


- REGRESSION TESTING is defined as a type of software testing to confirm that a recent program or code change has not adversely affected existing features.
- Regression Testing is nothing but a full or partial selection of already executed test cases which are re-executed executed to ensure existing functionalities work fine.
- This testing is done to make sure that new code changes should not have side effects on the existing functionalities.
- It ensures that the old code still works once the latest code changes are done.

Need of Regression testing:

- The Need of Regression Testing mainly arises whenever there is requirement to change the code and we need to test whether the modified code affects the other part of software application or not.
- Moreover, regression testing is needed, when a new feature is added to the software application and for defect fixing as well as performance issue fixing.

Usability Testing:



- Usability Testing also known as User Experience(UX) Testing, is a testing method for measuring how easy and user-friendly a software application is.

- Usability testing mainly focuses on user's ease of using application, flexibility of application to handle controls and ability of application to meet its objectives.

Web Based Testing:

Web Testing, or website testing is checking your web application or website for potential bugs before its made live and is accessible to general public. Web Testing checks for functionality, usability, security, compatibility, performance of the web application or website.
During this stage issues such as that of web application security, the functioning of the site, its access to handicapped as well as regular users and its ability to handle traffic is checked.

In Software Engineering, the following testing technique may be performed depending on your web testing requirements.

1. Functionality Testing of a Website:

- Functionality testing ensures a web application is working properly and correctly.
- It is performed to test the functionalities of each feature on the website.

Functional testing covers:

- **Unit Testing:**
- **Smoke Testing, Build Verification Testing and Confidence Testing:**
- **Sanity Testing:**
- **Regression Testing:**
- **Integration Testing:**
- **Usability Testing:**

2. Usability testing:

To verify how the application is easy to use with.

- Test the navigation and controls.
- Content checking.
- Check for user intuition.

https://www.interaction-design.org/literature/topics/intuitive-design

3.Interface Testing:

- Performed to verify the interface and the dataflow from one system to other.
- Testing the API requires making requests to multiple API endpoints to validate the response.

4. Database Testing:

- Test if any errors are shown while executing queries
- Data Integrity is maintained while creating, updating or deleting data in database.
- It will help to prevent data loss, save lost transaction data, and prevent unauthorized access to the information.
- Test data retrieved from your database is shown accurately in your web application

5. Compatibility testing.

- Testing for Cross-Compatibility With Browsers, Operating Systems and Mobile Devices

Compatibility testing is performed based on the context of the application.

- Browser compatibility
- Operating system compatibility
- Compatible to various devices like notebook, mobile, etc.

6. Performance Testing:

- Website application response times at different connection speeds
- Load test your web application to determine its behavior under normal and peak loads
- Stress test your web site to determine its break point when pushed to beyond normal loads at peak time.
- Test if a crash occurs due to peak load, how does the site recover from such an event

7. Security testing:

- Test unauthorized access to secure pages should not be permitted
- Restricted files should not be downloadable without appropriate access
- Check sessions are automatically killed after prolonged user inactivity
- On use of SSL certificates, website should re-direct to encrypted SSL pages.