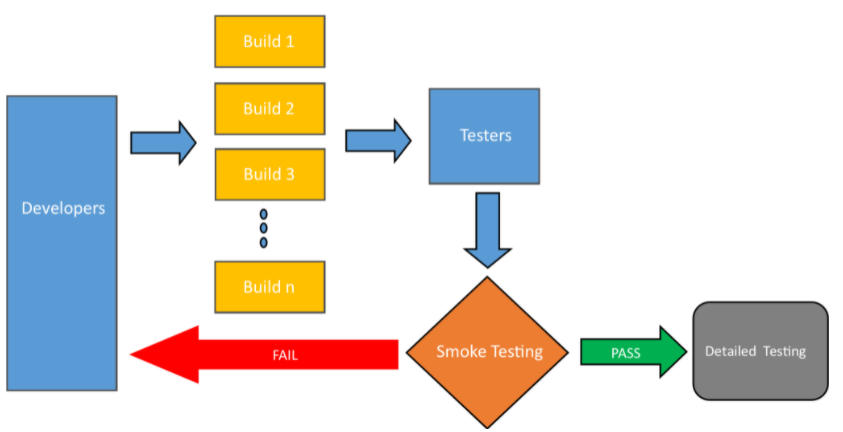
# What is Software Testing?

Software Testing is the process of **verifying** and **validating** whether the software program works as expected. And the purpose of Software Testing is to find the defects in Software, with an intention of improving the quality of the Software before it gets released into the market.

# What is Smoke Testing?

Smoke testing is a type of testing which is performed initially on the builds, to verify whether the critical or major functionalities of the Software application are working. If these tests pass, the testers will move to the detailed tests and if these tests fail, the testers will reject the build and move it back to the developers for fixing.



# What is Regression Testing?

Regression testing is generally needed when the below things are done:

* Addition of new feature
* Enhancement of existing features
* Bug fixes (Functional / Performance Bug fixes)
* Regression testing plays a major role in the project following Agile Development model
* Regression testing is performed to verify that the new code changes have not introduced any side effects in working functionalities

# What is Sanity Testing?

Sanity testing is a subset of Regression testing which is performed to verify whether the minor bug fixes or minor changes (which are done on the stable software) are working fine

# What is Black Box Testing?

Testing with i/p data 🡺 Executable program 🡺 o/p

* Black box testing a method of Software testing, which the testing team will perform without knowing the internal structure/design/implementation of the Application under test.
* Independent Testing teams having no programming knowledge will perform Black Box Testing.
* Black Box testing is generally performed as System Testing and Acceptance Testing.
* Black Box testing focuses on verifying the visual functioning of the Software using the Requirements and Specifications provided by the Business clients.

 The following are the different testing techniques involved in Black Box Testing:

* + Equivalence Partitioning
  + Boundary Value Analysis
  + Decision Tables
  + State Transition Testing

 Black box testing is not limited to Functional testing, instead, it involves the below types of testing:

* + Functional Testing – To verify the Requirements and Specifications against the Application under test
  + Non-Functional Requirements like performance testing, usability testing, accessibility testing etc. can be performed.

# What is White Box Testing?

White box testing is a method of Software testing, which tests the code and internal program structure of the Software. White box testing is also known as:

* Clear box testing
* Transparent box testing
* Glass box testing
* Structural testing
  + **Usually done by developers**

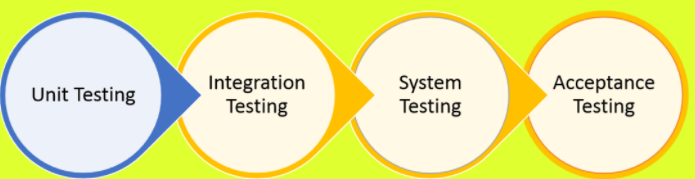
# What is Grey Box Testing?

Grey Box Testing is primarily useful for Integration Testing and Penetration Testing.

Grey Box is most suitable for testing the Web Based applications.



# What is Unit Testing?



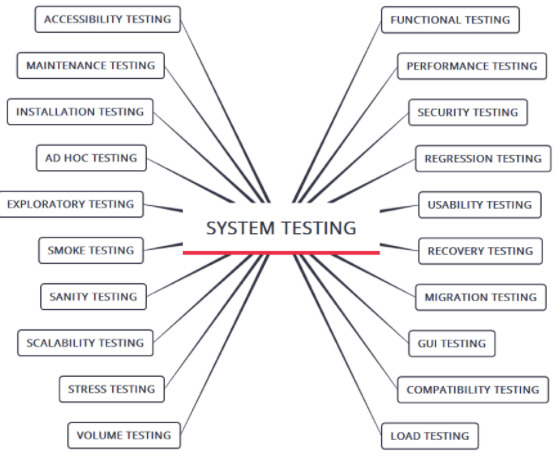
**Importance of Unit Testing**: The below are different points which state the importance of performing Unit Testing in the Software Development Life Cycle:

* The cost of the defects identified and fixed during Unit Testing is very less compared to the cost of the defects which are later identified and fixed during the later high levels of testing phases.
* With Unit Testing, defects are identified at the early stages of Software Development Life Cycle, which result in the reduction of bug fixing cost along with time.
* As part of regression testing, the Unit Tests can be automated and run every time along with the UI automated, when the changes are made to the application code

# What is Integration Testing?

Integration testing is one of the levels in Software testing which comes after Unit testing. The purpose of Integration testing is to combine the individually tested units to identify the defects emerging from their interaction

# What is System Testing?



# What is Acceptance Testing?

Acceptance Testing is the last/fourth level of Software testing, where the testing is performed by the Business Customers / End Users to confirm whether the application is ready for getting released into the market.



**Alpha Testing** – Acceptance Testing performed at the vendor site (i.e. Developer organization site) by non-project staff to confirm the applications readiness, before releasing to the Business Customers.

**Beta Testing** – After the Alpha testing is performed on the application at the vendor side, the application will be released to the Business Customers

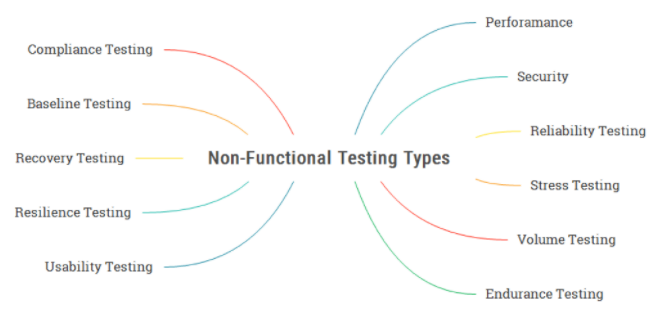
# What is Functional Testing?

Functional Testing is performed to test whether the software application is functioning according to the specified business requirements specifications. The following are verified as part of Functional Testing:

* Installation of the Application
* Application functionality
* User Interface of the Application
* API / Web Services of the Application
* Database of the Application
* Accessibility of the Application
* Proper error messages are displayed in the Application

# What is Non-Functional Testing?

Non-Functional Testing is performed to verify the non-functional characteristics of the software like performance, security, etc.



# What is Positive Testing?

 Positive Testing is an important process in Software Testing, where the testing is performed by providing valid inputs to the application and verifying whether the application is working as expected with the valid inputs.

# What is Negative Testing?

Negative Testing is an important process in Software Testing, where the testing is performed by providing invalid inputs to verify whether the application is able to handle invalid inputs properly.

# What is End-to-End Testing?

End-to-End testing is a type of testing which is performed to verify the software system as a whole. In this testing, end-to-end flows covering the multiple interconnected sub-systems will be created to test the real-world activities performed by the end users in the production/live environments. End-to-End testing is performed horizontally and vertically to increase the test coverage and increase the quality of the software system.

# What is Manual Testing?

Manual Testing is a testing process, where the Software Testers manually test the Software without using any Test Automation tools

**Step 1:** Software Tester goes through the Requirements Specifications Document and creates a bunch of test cases for testing the Software Application. The following are few test cases for logging into the Facebook application:

> Login to the facebook.com using the valid username and valid password

> Login to the facebook.com using invalid username and password

> Login to the facebook.com without providing any username or password

**Step 2:** Software Tester executes the test cases manually (i.e. with hands and eyes ). The following is an example of how the Software Tester will execute the created test case for logging into the Facebook application with valid username and valid password:

Software Tester manually opens the Browser

Software Tester manually types the facebook.com URL into the address field of the opened browser

Software Tester manually types the valid username using the keys on his computer keyboard into the ‘username’ field on the facebook page

Software Tester manually types the valid password using the keys on his computer keyboard into the ‘password’ field on the facebook page

Software Tester manually clicks on ‘Login’ button on facebook page using the Computer mouse

Software Tester manually checks with his eyes, whether the tester is able to login to the application and confirms it as successful login after the Facebook Home page is displayed.

**Step 3:** Software Tester reports bugs if the executed test cases are giving different results from the specified requirements. The following is an example of defect which can be found during test execution:

> On manually entering the valid username and valid password and clicking on ‘Login’ button, if the User is not able to login to the facebook, the Software Tester will report a Bug. After checking the Bug details provided by Software Tester, Software Developer fixes the Bug. Software Tester then retests the Bug and confirms whether the Bug is fixed.

Manual Testing versus Automation Testing: The following points will help you in understanding the difference between Manual Testing and Automated Testing:

> Manual Testing: Software Testers performing testing manually without using any test automation tools.

> Automated Testing: Here Test Automation tools like UFT/Selenium/Others will be used to perform testing on the required Software.

> As Investigation/Questioning/Reasoning/Observation/Judgment on the Application for finding bugs cannot be performed by Automation Testing and is only possible with Manual Testing. Due to this reason, Automation Testing cannot replace Manual Testing.

> Testers performing Manual Testing will generally understand the Software requirements from the Requirements Specification documents, derive test cases, execute the test cases manually with an intention of finding bugs and report the bugs observed during the execution of tests to developers.

> Testers will also perform retesting to check whether the reported defects are properly fixed by the developers.

> Manual Testing is repetitive and boring, as the testers need to frequently execute the test cases again and again when any changes are made to the Software Application by the developers.

> In Manual Testing, testers need to be patient, open-minded and creative.

> In Manual Testing, testers need to test the Applications with an End User perspective for finding good defects.

> Manual Testing comes first, and based on the project requirements Automation Testing will be performed next.

# What is Automation Testing?

Test Automation is a testing process, which is performed with the help of Test Automation tools

Popular Automation tools 🡺 Selenium, QTP, Tricentis Tosca, Ranorex, Robot Framework, Coded UI, QMetry Automation Studio, Appium, Squish, Silk Test etc.

# Agile Testing – The complete guide

- 𝑾𝒉𝒚 𝑨𝒈𝒊𝒍𝒆?

\* Traditional View of Project Management

\* Traditional View of Project Implementation

\* Real requirement of Customer

\* Challenges in Software Development

- 𝑨𝒈𝒊𝒍𝒆

\* What is Agile?

\* Agile Manifesto

\* Agile Principles

\* Scrum

\* Scrum - Timebox

- 𝑺𝒄𝒓𝒖𝒎 𝑹𝒐𝒍𝒆𝒔

\* Scrum Master

\* Product Owner

\* Development Team

- 𝑺𝒄𝒓𝒖𝒎 𝑬𝒗𝒆𝒏𝒕𝒔

\* Sprint Planning Meeting

\* User Stories

\* Daily Stand-up

\* Sprint Retrospective

\* Backlog Grooming

Defect life cycle

