**7. TESTING**

The testing process involves development of a test plan, executing the plan and documenting the test results.

**LEVELS OF TESTING**

Levels of testing include the different methodologies that can be used while conducting Software Testing. Following are the main levels of Software Testing

* Functional Testing.
* Non-Functional Testing.

**FUNCTIONAL TESTING**

* This is a type of black box testing that is based on the specifications of the software that is to be tested.
* The application is tested by providing input and then the results are examined that need to conform to the functionality it was intended for.
* Functional Testing of the software is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

# Unit Testing

* This type of testing is performed by the developers before the setup is handed over to the testing team to formally execute the test cases.
* Unit testing is performed by the respective developers on the individual units of source code assigned areas. The developers use test data that is separate from the test data of the quality assurance team.
* The goal of unit testing is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality.

# Integration Testing

* The testing of combined parts of an application to determine if they function correctly together is Integration testing.
* In a comprehensive software development environment, bottom-up testing is usually done first, followed by topdown testing. The process concludes with multiple tests of the complete application, preferably in scenarios designed to mimic those it will encounter in customers' computers, systems and network.

# System Testing

* This is the next level in the testing and tests the system as a whole. Once all the components are integrated, the application as a whole is tested rigorously to see that it meets Quality Standards. This type of testing is performed by a specialized testing team.

**System testing is so important because of the following reasons:**

* System Testing is the first step in the Software Development Life Cycle, where the application is tested as a whole.
* The application is tested thoroughly to verify that it meets the functional and technical specifications.
* The application is tested in an environment which is very close to the production environment where the application will be deployed.
* System Testing enables us to test, verify and validate both the business requirements as well as the Applications Architecture.

# Acceptance Testing

* This is arguably the most importance type of testing as it is conducted by the Quality Assurance Team who will gauge whether the application meets the intended specifications and satisfies the clients requirements.
* More ideas will be shared about the application and more tests can be performed on it to gauge its accuracy and the reasons why the project was initiated.
* Acceptance tests are not only intended to point out simple spelling mistakes, cosmetic errors or Interface gaps, but also to point out any bugs in the application that will result in system crashers or major errors in the application.
* By performing acceptance tests on an application the testing team will deduce how the application will perform in production.

**NON-FUNCTIONAL TESTING**

* This section is based upon the testing of the application from its non-functional attributes.
* Non-functional testing of Software involves testing the Software from the requirements which are non functional in nature related but important a well such as performance, security, user interface etc.
* Some of the important and commonly used non-functional testing types are mentioned as follows:

# SOFTWARE TESTING METHODS

There are different methods which can be use for Software testing. This chapter briefly describes those methods.

# Black Box Testing

* The technique of testing without having any knowledge of the interior workings of the application is Black Box testing.
* The tester is oblivious to the system architecture and does not have access to the source code.
* Typically, when performing a black box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

# White Box Testing

* White box testing is the detailed investigation of internal logic and structure of the code. White box testing is also called glass testing or open box testing.
* In order to perform white box testing on an application, the tester needs to possess knowledge of the internal working of the code.
* The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.

# Grey Box Testing

* Grey Box testing is a technique to test the application with limited knowledge of the internal workings of an application.
* In software testing, the term *the more you know the better* carries a lot of weight when testing an application.
* Mastering the domain of a system always gives the tester an edge over someone with limited domain knowledge.
* Unlike black box testing, where the tester only tests the application's user interface, in grey box testing, the tester has access to design documents.