Testing

Software testing is a critical element of the software development cycle. The testing is essential for ensuring the Quality of the software developed and represents the ultimate view of specification, design, and code generation. Software testing is defined as the process by which one detects the defects in the software. Testing is a set of activities that work towards the integration of entire computer-based system.

Testing Objectives

1. Efficient testing helps preventing defects and that helps in provided an error-free application

2. To evaluate work products such as requirements, user stories, design, and code

3. To find failures and defects. Defects should be identified as early in the test cycle as possible

4. To verify whether all specified requirements have been fulfilled

5. To provide enough information to stakeholders to allow them to make informed decisions, especially regarding the level of quality of the test object

6. To validate whether the test object is complete and works as the users and other stakeholders expect

7. To reduce the level of risk of inadequate software quality

8. To build confidence in the level of quality of the test object

White Box Testing

White-box testing techniques analyse the internal architecture, the data structures used, the internal design, the code structure, and the behaviour of the software, in addition to functionality like black-box testing.

Software development can be tested at the system, integration, and unit levels. Verifying that an application's working flow is one of White box testing's fundamental objectives. It is comparing a sequence of specified inputs to desired or expected outputs to identify bugs when a particular input does not provide the desired outcome.

Black Box Testing

Black box testing entails evaluating a system without being aware of how it operates within. A tester inputs data and monitors the output produced by the system being tested. This allows for the identification of the system's response time, usability difficulties, and reliability concerns as well as how the system reacts to expected and unexpected user activities.

Because it tests a system from beginning to end, black box testing is a potent testing method. A tester can imitate user action to check whether the system fulfills its promises, much as end users "don't care" how a system is programmed or designed and expect to get a suitable response to their requests. A black box test checks all pertinent subsystems along the route, including UI/UX, the web server or application server, the database, dependencies, and integrated systems.

Unit Testing

Individual software components or components are tested as part of a type of software testing known as unit testing. The goal is to confirm that each piece of software code operates as intended. Developers perform unit testing while creating an application (the coding phase). Unit tests isolate a specific piece of code and validate its accuracy. A singular function, method, procedure, module, or object might be considered a unit.

Unit testing is the initial level of testing carried out prior to integration testing in the SDLC, STLC, and V Model. Unit testing is a type of White Box testing that is often carried out by the developer. However, in the real world, QA engineers also perform unit testing because of time constraints or developers' resistance to testing.

Integration Testing

Software components are logically connected and tested as a unit in a type of testing called integration testing. Multiple software modules created by various programmers make up a typical software project. This level of testing's objective is to find issues with how various software modules interact when they are combined.

The main goal of integration testing is to examine how effectively these modules communicate data. The name "I & T" (Integration and Testing), "String Testing," and occasionally "Thread Testing" are also used to describe it.