Software Testing Levels

Software Testing - Levels

Overview of Software Testing Levels.

Before release, any software product goes through a testing process to make sure that it is working properly as it is supposed to be. There are various levels (stages) of soft testing. But, generally, software testing is divided into four levels: Unit Testing: Unit testing (or component testing) is the first stage in the testing process. A unit is the smallest testable and independent component of the software. Unit testing is normally performed by software developers before handing software over to testers for executing other formal tests.

The main purpose of unit testing is to make sure that every unit of the software is working as designed.In addition, as the cost of fixing bugs in the late stages is much higher than in the early stages, unit testing helps reduce cost by finding problems early. White box testing method is normally used at this stage. Integration Testing: The next stage is integration testing and in this stage, all units are integrated together and tested. It is a form of testing in the testing process performed to detect defects in the interactions and the interfaces between the integrated units. Black box testing technique is usually used at this stage, but in many cases, both Black and White testing method are combined together.

There are four main integration testing approaches, namely: Big bang: In the big bang approach, All components or modules are integrated together to complete a software system and then integration testing is performed. Top-down: In this approach, the process is carried out from the highest level modules to the lower level ones. This method requires Test stubs, Stub is a module which will temporarily substitute for submodules if they are not available for integration during the early phases. Bottom-up: Bottom-up testing is an approach where the lowest level components are tested first, and then progressively higher level components are tested. Drivers are required in this method to simulate the main modules which are not available in the early phases.Sandwich: Sandwich testing (or Hybrid approach) is a form of integration testing which is the combination of Top-down and Bottom-up testing.System Testing: After integration testing, the fully integrated application to check that whether the system meets its software requirements specifications (SRS).

System Testing is a type of black box testing method thus the knowledge of internal code is not required. This testing is performed in an environment that closely resembles production environment in order to provide more reliable and efficient outcoming. It is an important phase

as it helps evaluate the functional, business and technical requirements. There are various types of system testing: Usability Testing - Focuses on the user's ease to operate and use and checks that the user interface are user friendly.Load Testing - Load Testing is necessary to know that a software solution will perform under real-life loads.Regression Testing- Type of software testing that seeks to uncover software errors after changes to the program (e.g., bug fixes or new functionality) have been made, by retesting the program. This testing process is performed by the testing teams.Recovery Testing - Testing technique which evaluates how well a system recovers from crashes, hardware failures, or other catastrophic problems. This testing process is performed by the testing teams.

Migration Testing - Migration testing is done to ensure that the software can be moved from older system infrastructures to current system infrastructures without any issues.Functional Testing - Type of black box testing that bases its test cases on the specifications of the software component under test. This testing process is performed by testing teams.Performance Testing: To test that the performance of the system and to make sure system does not break while operating with available resources.Security Testing: To test that the system is secured enough to protect it from unintended users.User Acceptance Testing: User acceptance testing (or Beta, End-User Testing or Application) is the final testing in the software testing process. In this phase, end users test the software to make sure it can handle required tasks in real-world scenarios as per specifications. If the software product passes this stage, it means that end user has accepted it and it is ready to go live. Following are the different types of user acceptance testing: Alpha & Beta Testing: aim to validate the fitness-for-use of the system by the business user. The user acceptance test is performed by the users and application managers.

Contract Acceptance Testing: a developed software is tested against certain criteria and specifications which are predefined and agreed upon in a contract. The project team defines the relevant criteria and specifications for acceptance at the same time when the team agrees on the contract itself.Regulation Acceptance Testing: also known as Compliance Acceptance Testing, examines whether the software complies with the regulations. This includes governmental and legal regulations.Operational Acceptance Testing: also known as Operational Readiness Testing or Production Acceptance Testing, these test cases ensure there are workflows in place to allow the software or system to be used. This should include workflows for backup plans, user training, and various maintenance processes and security checks.Black Box Testing: analyses certain functionalities without letting testers see the internal code structure. Black Box Testing is part of User Acceptance Testing, because Black Box Tests share the same principles as UAT.Factors affecting test scope.Project size.Complexity of project. Budget.Time scope for project.Human resources.Why test at different levels.Software development naturally split to phases.Easily track bugs.Ensures a working subsystem/ component/ library.Software reuse more practical.