02-06-2023

**INTRODUCTION TO METHODS OF SOFTWARE TESTING**

Software testing methodologies are the various strategies or approaches used to test an application to ensure it behaves and looks as expected. These encompass everything from front to back-end testing, including unit and system testing.

1. **Functional Testing**

Functional testing is a type of software testing that focuses on verifying the functionality of the system or software application. It ensures that the software meets the specified requirements and performs the intended tasks correctly. Some common techniques used in functional testing include:

1. Unit Testing:

* Unit testing is the first level of testing and is often performed by the developers themselves.
* It is the process of ensuring individual components of a piece of software at the code level are functional and work as they were designed to.
* Developers in a test-driven environment will typically write and run the tests prior to the software or feature being passed over to the test team.
* Unit testing can be conducted manually, but automating the process will speed up delivery cycles and expand test coverage.
* Unit testing will also make debugging easier because finding issues earlier means they take less time to fix than if they were discovered later in the testing process.
* TestLeft is a tool that allows advanced testers and developers to shift left with the fastest test automation tool embedded in any IDE.

1. Integration Testing:

* After each unit is thoroughly tested, it is integrated with other units to create modules or components that are designed to perform specific tasks or activities.
* These are then tested as group through integration testing to ensure whole segments of an application behave as expected (i.e., the interactions between units are seamless).
* These tests are often framed by user scenarios, such as logging into an application or opening files.
* Integrated tests can be conducted by either developers or independent testers and are usually comprised of a combination of automated functional and manual tests.

1. System Testing:

* System testing is a black box testing method used to evaluate the completed and integrated system, as a whole, to ensure it meets specified requirements.
* The functionality of the software is tested from end-to-end and is typically conducted by a separate testing team than the development team before the product is pushed into production.

1. Acceptance Testing:

* Acceptance testing is the last phase of functional testing and is used to assess whether or not the final piece of software is ready for delivery.
* It involves ensuring that the product is in compliance with all of the original business criteria and that it meets the end user’s needs.
* This requires the product be tested both internally and externally, meaning you’ll need to get it into the hands of your end users for beta testing along with those of your QA team.
* Beta testing is key to getting real feedback from potential customers and can address any final usability concerns.

1. **Non-Functional Testing:**

Non-functional testing focuses on evaluating the non-functional aspects of the software, such as performance, reliability, security, and usability. It ensures that the software meets the desired quality attributes and provides a satisfactory user experience. Some common types of non-functional testing include:

1. Performance Testing:

Performance testing is a non-functional testing technique used to determine how an application will behave under various conditions. The goal is to test its responsiveness and stability in real user situations. Performance testing can be broken down into four types:

* **Load testing** is the process of putting increasing amounts of simulated demand on your software, application, or website to verify whether or not it can handle what it’s designed to handle.
* **Stress testing** takes this a step further and is used to gauge how your software will respond at or beyond its peak load. The goal of stress testing is to overload the application on purpose until it breaks by applying both realistic and unrealistic load scenarios. With stress testing, you’ll be able to find the failure point of your piece of software.
* **Endurance testing**, also known as soak testing, is used to analyze the behavior of an application under a specific amount of simulated load over longer amounts of time. The goal is to understand how your system will behave under sustained use, making it a longer process than load or stress testing (which are designed to end after a few hours). A critical piece of endurance testing is that it helps uncover memory leaks.
* **Spike testing** is a type of load test used to determine how your software will respond to substantially larger bursts of concurrent user or system activity over varying amounts of time. Ideally, this will help you understand what will happen when the load is suddenly and drastically increased.

1. Security Testing:

* With the rise of cloud-based testing platforms and cyber-attacks, there is a growing concern and need for the security of data being used and stored in software.
* Security testing is a non-functional software testing technique used to determine if the information and data in a system is protected.
* The goal is to purposefully find loopholes and security risks in the system that could result in unauthorized access to or the loss of information by probing the application for weaknesses.
* There are multiple types of this testing method, each of which aimed at verifying six basic principles of security:
* Integrity
* Confidentiality
* Authentication
* Authorization
* Availability
* Non-repudiation

1. Usability Testing:

* Usability testing is a testing method that measures an application’s ease-of-use from the end-user perspective and is often performed during the system or acceptance testing stages.
* The goal is to determine whether or not the visible design and aesthetics of an application meet the intended workflow for various processes, such as logging into an application.
* Usability testing is a great way for teams to review separate functions, or the system as a whole, is intuitive to use.

1. Compatibility Testing:

* Compatibility testing is used to gauge how an application or piece of software will work in different environments.
* It is used to check that your product is compatible with multiple operating systems, platforms, browsers, or resolution configurations.
* The goal is to ensure that your software’s functionality is consistently supported across any environment you expect your end users to be using.

1. Reliability Testing:

* Reliability testing is a type of non-functional testing that focuses on evaluating the software's ability to perform consistently and reliably over a period of time or under specific conditions.
* The goal of reliability testing is to identify any potential issues or weaknesses that may lead to failures, crashes, or system downtime.
* By conducting reliability testing, organizations can gain confidence in the software's stability and ensure a smooth user experience.

1. Stress Testing:

* Stress testing is a type of non-functional testing that evaluates the performance and stability of a software system under extreme or stressful conditions.
* The purpose of stress testing is to identify the breaking points, limitations, and weaknesses of the software, as well as to determine how it recovers from failure.

1. Reliability Testing:

* Reliability testing is a type of non-functional testing that focuses on assessing the software's ability to perform consistently and reliably over a period of time or under specific conditions.
* The objective of reliability testing is to identify potential failures, weaknesses, and areas of improvement to ensure a stable and dependable software system.