02-06-2023

**INTRODUCTION TO SOFTWARE TESTING**

Software testing can be stated as the process of verifying and validating whether a software or application is bug-free, meets the technical requirements as guided by its design and development, and meets the user requirements effectively and efficiently by handling all the exceptional and boundary cases.

The process of software testing aims not only at finding faults in the existing software but also at finding measures to improve the software in terms of efficiency, accuracy, and usability. It mainly aims at measuring the specification, functionality, and performance of a software program or application.

* Software testing is a process of identifying the correctness of software by considering its all attributes (Reliability, Scalability, Portability, Re-usability, Usability) and evaluating the execution of software components to find the software bugs or errors or defects.
* Software testing provides an independent view and objective of the software and gives surety of fitness of the software.
* It involves testing of all components under the required services to confirm that whether it is satisfying the specified requirements or not.
* The process is also providing the client with information about the quality of the software. Testing is mandatory because it will be a dangerous situation if the software fails any of time due to lack of testing.
* So, without testing software cannot be deployed to the end user.

**Why Software Testing is important?**

Software testing is crucial for ensuring the quality, reliability, usability, and security of software systems. It helps detect and fix issues early, enhances the user experience, and builds confidence among users and stakeholders.

1. **Quality assurance:** Testing helps ensure that the software meets the desired quality standards. By identifying and fixing defects, errors, and vulnerabilities early in the development process, testing helps prevent issues from reaching the end users, thereby enhancing the overall quality of the software.
2. **Bug detection:** Testing helps uncover bugs or glitches in the software. It allows developers to identify and rectify these issues before the software is released to the users, reducing the likelihood of user dissatisfaction, system failures, or potential security breaches.
3. **Reliability and stability:** Thorough testing help improve the reliability and stability of software systems. By subjecting the software to various scenarios and conditions, testers can validate its performance, robustness, and resilience, ensuring that it functions as expected under different circumstances.
4. **User experience:** Testing plays a crucial role in ensuring a positive user experience. By conducting usability testing and gathering feedback from real users, developers can identify areas of improvement and make necessary adjustments to enhance the software's usability, intuitiveness, and overall user satisfaction.
5. **Compliance and security:** Testing help assess the software's compliance with relevant industry standards, regulations, and security requirements. It helps identify vulnerabilities and weaknesses that could be exploited by malicious actors, ensuring that appropriate security measures are implemented to protect sensitive data and prevent potential breaches.
6. **Cost-effectiveness:** Identifying and fixing defects during the development phase is generally more cost-effective than addressing them after the software has been deployed. Testing helps reduce the likelihood of expensive rework, maintenance, or potential legal issues that may arise from faulty software.
7. **Customer confidence:** Thorough testing instils confidence in customers and stakeholders regarding the software's quality and reliability. A well-tested software product is more likely to meet customer expectations, resulting in higher customer satisfaction and increased trust in the software and its development team.

**When to start Testing?**

An early start to testing reduces the cost and time to rework and produce error-free software that is delivered to the client. However, in Software Development Life Cycle (SDLC), testing can be started from the Requirements Gathering phase and continued till the deployment of the software. It also depends on the development model that is being used. For example, in the Waterfall model, formal testing is conducted in the testing phase; but in the incremental model, testing is performed at the end of every increment/iteration and the whole application is tested at the end.

Testing is done in different forms at every phase of SDLC −

* During the requirement gathering phase, the analysis and verification of requirements are also considered as testing.
* Reviewing the design in the design phase with the intent to improve the design is also considered as testing.
* Testing performed by a developer on completion of the code is also categorized as testing.

**When to stop Testing?**

It is difficult to determine when to stop testing, as testing is a never-ending process and no one can claim that a software is 100% tested. The following aspects are to be considered for stopping the testing process −

* Testing Deadlines
* Completion of test case execution
* Completion of functional and code coverage to a certain point
* Bug rate falls below a certain level and no high-priority bugs are identified
* Management decision

**Software Testing can be divided into two steps:**

1. Verification: it refers to the set of tasks that ensure that the software correctly implements a specific function.
2. Validation: it refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements.