**TYPES OF SOFTWARE TESTING**

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 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.

Software Testing can be broadly classified into two types,

**Manual Testing:**

Manual testing includes testing software manually, i.e., without using any automation tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behavior or bug. There are different stages for manual testing such as unit testing, integration testing, system testing, and user acceptance testing.

Testers use test plans, test cases, or test scenarios to test software to ensure the completeness of testing. Manual testing also includes exploratory testing, as testers explore the software to identify errors in it.

**Automation Testing:**

 Automation testing, which is also known as Test Automation, is when the tester writes scripts and uses another software to test the product. This process involves the automation of a manual process. Automation Testing is used to re-run the test scenarios quickly and repeatedly, that were performed manually in manual testing.

Apart from regression testing, automation testing is also used to test the application from a load, performance, and stress point of view. It increases the test coverage, improves accuracy, and saves time and money when compared to manual testing.

Software testing techniques can be majorly classified into two categories:

**Black Box Testing:**

The technique of testing in which the tester doesn’t have access to the source code of the software and is conducted at the software interface without any concern with the internal logical structure of the software is known as black-box testing.

**White-Box Testing:**

The technique of testing in which the tester is aware of the internal workings of the product, has access to its source code, and is conducted by making sure that all internal operations are performed according to the specifications is known as white box testing.

**SOFTWARE REQUIREMENTS SPECIFICATION**

In software testing, SRS stands for Software Requirements Specification. The SRS is a crucial document that serves as a foundation for the entire software development and testing process. It outlines the functional and non-functional requirements of the software system to be developed. Here's a closer look at the significance of SRS in software testing:

**1.Requirement Understanding:** The SRS document provides a clear and detailed description of what the software is supposed to do. This helps all stakeholders, including developers and testers, understand the project's scope and objectives.

**2.Test Planning:** Test planning involves creating a test strategy and test plan based on the requirements specified in the SRS. Testers use the SRS to identify what needs to be tested, the test cases to be created, and the test environments to be set up.

**3.Traceability:** SRS documents are used to establish traceability between requirements and test cases. Each requirement specified in the SRS should have corresponding test cases to ensure that all requirements are adequately tested.

**4.Requirement Validation:** Testers can review the SRS to identify ambiguities, inconsistencies, and gaps in the requirements. This helps in refining the requirements and ensuring that they are complete and well-defined.

**5.Test Design:** The SRS serves as a basis for designing test cases, test scenarios, and test data. Testers can map the SRS requirements to specific test conditions and expected outcomes.

**6.Test Execution:** During the actual testing phase, testers refer to the SRS to ensure that they are testing the software according to the specified requirements. Any discrepancies between the software's behavior and the SRS are identified as defects.

**7.Requirement Change Management:** As the project progresses, requirements may change or evolve. The SRS document helps in managing these changes by providing a reference point for evaluating the impact of changes on testing efforts.

**8.Validation and Verification:** The SRS is used to validate that the delivered software meets the stated requirements and to verify that the requirements have been correctly implemented.

**9.Documentation and Reporting:** Testers use the SRS to document their test cases, test results, and any defects found during testing. It also assists in generating test reports to communicate the testing progress and status to stakeholders.

**10.Compliance and Auditing:** In regulated industries such as healthcare or finance, the SRS serves as a critical document for compliance and auditing purposes. It demonstrates that the software was tested in accordance with the specified requirements.

In summary, the Software Requirements Specification (SRS) is a fundamental document in software testing that plays a central role in understanding, planning, designing, and executing tests. It ensures that the software meets the specified requirements and helps in managing changes and communication with stakeholders throughout the testing process.



**TYPES OF TEST PLAN:**

A test plan is a document that outlines the scope, approach, resources, and schedule for testing activities within a software development project. Test plans can vary in complexity and depth depending on the project's size and requirements. There are several types of test plans, each serving a specific purpose. Here are some common types of test plans:

**1. Master Test Plan (MTP):** The Master Test Plan provides an overall view of the testing strategy for an entire project. It includes high-level details such as the testing objectives, scope, schedule, resources, and test deliverables. It may also define testing standards and procedures to be followed throughout the project.

**2. System Test Plan:** This plan focuses on testing the entire software system or application to ensure that it functions correctly as a whole. It often includes functional, integration, and system-level testing activities.

**3. Acceptance Test Plan (ATP):** The Acceptance Test Plan is typically created by the client or end-users. It outlines the acceptance criteria and tests that must be performed to determine whether the software meets the user's needs and is ready for production use.

**4. Integration Test Plan:** This plan focuses on testing the interfaces and interactions between various components or modules of the software. It verifies that the integrated components work together as intended.

**5. Regression Test Plan:** A regression test plan outlines the tests that need to be run when changes are made to the software to ensure that new updates have not introduced any unintended side effects or defects.

**6.Security Test Plan:** For applications that handle sensitive data or require high-security measures, a security test plan is crucial. It outlines the security testing strategy, including penetration testing and vulnerability assessments.