**SOFTWARE TESTING**

**TESTING:**

Software testing is the process of checking the quality, functionality, and performance of a software product before launching. To do software testing, testers either interact with the software manually or execute test scripts to find bugs and errors, ensuring that the software works as expected. Software testing is also done to see if business logic is fulfilled, or if there are any missing gaps in requirements that need immediate tackles.

**TYPES OF SOFTWARE TESTING:**

Different types of software testing can be classified into multiple categories based on test objective, test strategy, and deliverables. Currently, there are major software testing types that Quality Assurance professionals frequently use, including:

Functional Testing:

A type of software testing to verify whether the application delivers the expected output.

Non- Functional Testing:

It is a type of software testing to verify whether the non-functional aspects of an application (e.g., stability, security, and usability) are working as expected.It

Approaches To Software Testing:

Manual Testing:

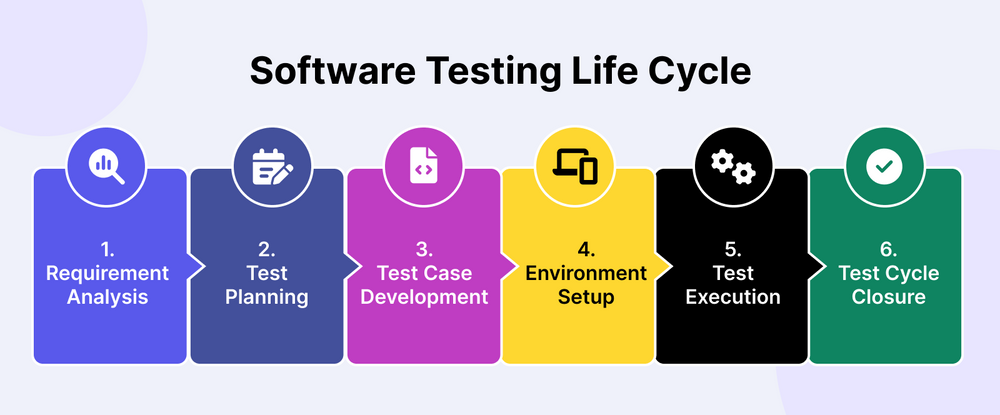
Testing software manually by humans without any automation tool or script.

Automation Testing:

Testing software using tools or scripts that automatically interact with the software. The human tester only needs to execute the script and let it do the rest of the testing.

Software Testing Process:

Many software testing initiatives follow a process commonly known as Software Testing Life Cycle (STLC). The STLC consists of 6 key activities to ensure that all software quality goals are met, as shown below:

**1.Requirement Analysis:**

In this stage, software testers work with stakeholders involved in the development process to identify and understand test requirements. The insights from this discussion, consolidated into the Requirement Traceability Matrix (RTM) document, will be the foundation to build the test strategy.

What a developer and a tester understand from the requirement analysis stage is different. A developer focuses on translating requirements into code, including architecture, design methods, and technologies, while a tester examines the code's testability. They determine how the code can be broken down into smaller parts, scenarios, and test cases. Ensuring testability is crucial during the design phase to avoid ambiguous requirements that can lead to invalid software tests.

After that, testers and developers have to collaborate to understand the feasibility of implementing business requirements. If these requirements can not be met within the given constraints, limitations, or resources, they will need to discuss with the business side (either the Business Analyst, Project Manager, and/or the client) to make adjustments or seek alternative solutions.

2.Test Planning:

After thorough analysis, a[test plan](https://katalon.com/resources-center/blog/test-plan) is created. Test planning involves aligning with relevant stakeholders on the test strategy:

* **Test objectives**: Define attributes like functionality, usability, security, performance, and compatibility.
* **Output and deliverables**: Document the test scenarios, test cases, and test data to be produced and monitored.
* **Test scope**: Determine which areas and functionalities of the application will be tested (in-scope) and which ones won't (out-of-scope).
* **Resources**: Estimate the costs for test engineers, manual/automated testing tools, environments, and test data.
* **Timeline**: Establish expected milestones for test-specific activities along with development and deployment.
* **Test approach**: Assess the testing techniques (white box/black box testing), test levels (unit, integration, and end-to-end testing), and test types (regression, sanity testing) to be used.

For a greater degree of control over the project, software testers can add a Contingency plan to adjust the variables in case the project moves in an unexpected direction.

**3. Test Case Development**

After defining the scenarios and functionalities to be tested, we'll write the test cases. For manual test cases, test management tools like Xray can be used to note down details of what was performed, the results, findings, and suggestions for developers to reproduce those bugs. For automated tests, intuitive UIs provided by tools like Katalon, Ranorex, or TestComplete are available. Open-source options like Selenium, Cypress, and Playwright are also popular for building custom frameworks.

**4. Test Environment Setup**

This step can be done in parallel with Test Case Development. A test environment is the software and hardware configurations under which the application is tested, including a database server, front-end running environment, browser, network, hardware, etc. QA teams will plan resource usage for Test Environment development. This practice ensures efficient resource allocation. Here is a quick checklist of items you need to consider when setting up test environment:

* Verify hardware specifications (CPU, RAM, storage).
* Check software dependencies (operating system, libraries, frameworks).
* Validate network requirements (firewall rules, ports, connectivity).
* Ensure the test environment is separate from the production environment.
* Use network segregation, virtualization, or dedicated hardware.
* Identify relevant test scenarios and data requirements.
* Create representative test data sets.
* Consider data privacy and security regulations.
* Install necessary operating systems and software.
* Set up databases, web servers, and other required components.
* Configure network settings, firewalls, and security measures.
* Establish recovery mechanisms to restore the environment to a known state.
* Implement regular backups to prevent data loss or configuration issues.
* Establish recovery mechanisms to restore the environment to a known state.
* Implement regular backups to prevent data loss or configuration issues.

**5. Test Execution**

With clear objectives in mind, the QA team writes test cases, test scripts, and prepares necessary test data for execution.

Tests can be executed manually or automatically. Manual testing is suitable when human insights and judgment are required, while automation testing is preferable for repetitive flows with minor adjustments. Once the tests are executed, any defects found are tracked and reported to the development team, who promptly resolve them.

**6. Test Cycle Closure**

This is the final phase of Software Testing. Software testers will gather to analyze what they found from the tests, evaluate the effectiveness, and document key takeaways for future reference. It is crucial to regularly [evaluate your QA team’s software testing process](https://katalon.com/resources-center/blog/questions-evaluate-software-testing-process) to remain in control of all testing activities across all phases of the STLC.