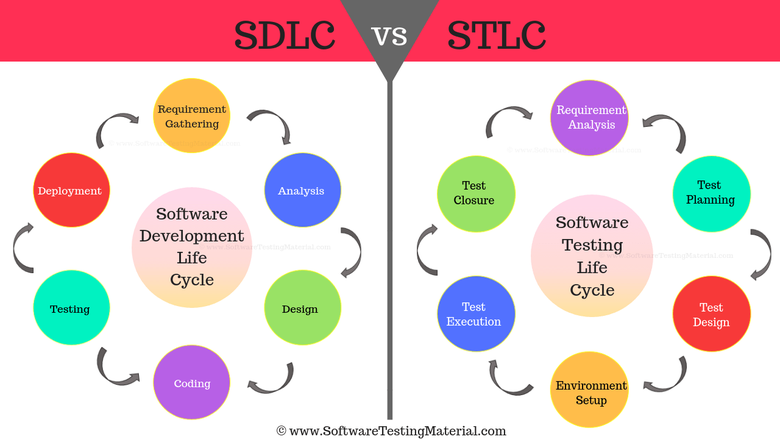
**Overall Introduction of the Software Testing**

**Introduction to Software Testing:**

**What is Software Testing?**

**Software testing is a systematic process of evaluating a software application to identify any defects or discrepancies between expected and actual results. The goal of testing is to ensure that the software meets specified requirements, functions as expected, and is of high quality.**



**What is Lifecycle?**

The life cycle means stages of testing which a QA/Tester performs on the software during the entire Testing Process.

**WHAT IS SOFTWARE TESTING LIFE CYCLE (STLC)?**

STLC is a series of certain phases executed to perform the Software Testing or to make sure that the required software/application or any website is as functional as required.

There are 6 different phases in the Software Testing Life Cycle. STLC is a testing process which is executed in a sequence, in order to meet the quality goals. STLC is followed by the QA team.

STLC is mainly similar to the SLDC. STLC can also run alongside the SDLC.

**Main Objectives of STLC.**

STLC helps the Testers to decide priorities for what steps to be carried out for testing, and when to accomplish the testing. Each of the phases mentioned below has some Entry Criteria and Exit Criteria and on the basis of that, the Testing Team can decide whether to move forward to the next phase of the Life cycle or not.

\* **Entry Criteria** is a minimum set of conditions that should be considered **before starting the testing.**

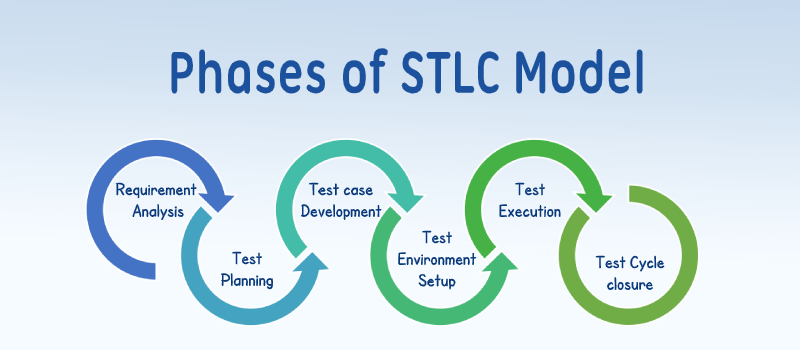
\* **Exit Criteria** is also a minimum set of conditions that are to be completed **in order to stop the testing.**

**Phases of STLC:-**

So in this process, each step is carried out in a systematic manner. There are different phases in STLC which are given below. The testing activities start from the requirements analysis phase and goes through all the phases one by one before completing the Test cycle closure phase.

Each phase varies from organization to organization.

But the most common phases are as below:-



**Requirement Analysis:-**

In this phase, the QA Team coordinates with the Developer/Client to understand the requirements for the particular project in detail and understand what the team will be testing.

The test team understands the requirements from the testing point of view for testable points.

Below are the Types of Requirements:-

=> Business Requirements

=> Architectural Requirements

=> System & Integration Requirements

**Test Planning:-**

Test Planning is known as the most important phase on Software Testing Life Cycle.

In this, the QA Manager will decide the estimate for the cost and the testing effort required to test a particular project.

QA Manager will also prepare the Test Plan also known as Test Strategy and will finalize it.

**Test Case Development:-**

This is the phase where the testing team makes a sheet of the detailed test cases. The Test data is also prepared by the testing team if required. The Test cases and Test data created are then reviewed by the QA Manager and then reworked as well.

\*  **Points to be kept in mind while creating Test Cases:-**

=> Test cases are needed to be kept simple.

=> They are prepared by keeping the End-Users in mind.

=> Test Case repetition needs to be avoided.

**Test Environment Setup:-**

Test Environment Setup is a vital part of the STLC.

A Test Environment is a setup of Software and Hardware for the testing teams to execute test cases.

\* **Steps to Setup the Test Environment:-**

=> To set up a Test Environment, a list of required Software and Hardware needs to be prepared after analyzing the requirements.

=> Setup the Test Environment.

=> Execute the Smoke Test Cases.

**Test Execution:-**

Test Execution is the process to compare the expected and actual results. The test plans and test cases prepared by the testing team will be carrying out the Test Execution.

During this, if any Bugs are found, they will be assigned back to Development to rectify/or to be modified via any Bug Tracking Tool. When bugs will be fixed, retesting would be performed by the tester.

**Test Cycle Closure:-**

A Test Team meeting will be arranged to discuss and analyze the testing strategies which are needed to be followed to prevent the blockers in the future,  which are occurred in the current cycle.

**Why is Software Testing Important?**

**Quality Assurance: Testing ensures that the software meets quality standards and is free of critical defects before it is released.**

**Risk Mitigation: Identifying and fixing defects early in the development process reduces the risk of issues in production.**

**Customer Satisfaction: High-quality software leads to better customer satisfaction and user experience.**

**Cost Savings: Fixing defects after release is more expensive than identifying and fixing them during the development phase.**

**Key Concepts in Software Testing:**

**Testing Levels:**

**Unit Testing: Testing individual components or modules in isolation.**

**Integration Testing: Verifying interactions between integrated components.**

**System Testing: Evaluating the entire system's functionality.**

**Acceptance Testing: Ensuring the software meets user requirements.**

**Testing Types:**

**Manual Testing: Testers manually execute test cases without using automation tools.**

**Automated Testing: Test cases are executed using automated testing tools and scripts.**

**Functional Testing: Verifies that the software functions as expected.**

**Non-Functional Testing: Assesses non-functional aspects (e.g., performance, security).**

**Testing Techniques:**

**Black Box Testing: Focuses on the functionality without knowledge of internal code.**

**White Box Testing: Examines internal code structures and logic.**

**Gray Box Testing: Combines elements of both black and white box testing.**

**Test Planning and Design:**

**Test Planning: Defining the scope, objectives, resources, and schedule for testing.**

**Test Case Design: Creating detailed test cases based on requirements and specifications.**

**Defect Life Cycle:**

**Defect Logging: Recording identified defects.**

**Defect Triaging: Prioritizing and assigning defects for resolution.**

**Defect Closure: Verifying that reported defects have been fixed.**

**Test Execution and Reporting:**

**Test Execution: Running test cases and recording results.**

**Defect Reporting: Communicating identified defects to development teams.**

**Test Summary Reports: Providing an overview of testing activities and outcomes.**

**Incomplete Requirements: Testing without clear requirements can be challenging.**

**Time and Resource Constraints: Limited time and resources impact testing thoroughness.**

**Changing Requirements: Frequent changes can lead to scope creep and additional testing efforts.**

**Conclusion:**

**Effective software testing is a critical component of the software development lifecycle. It ensures the delivery of high-quality software that meets user expectations, minimizes risks, and contributes to overall project success. A well-structured testing process helps identify and address defects early, reducing the likelihood of issues in production and improving customer satisfaction.**

**Manual Testing**

**What is Mobile Manual Testing?**

**Mobile manual testing is a process of evaluating and validating the functionality, usability, and performance of mobile applications manually without the use of automated testing tools. This testing approach involves human testers interacting with the mobile application on various devices to ensure that it meets specified requirements and provides a seamless user experience.**

**Key Aspects of Mobile Manual Testing:**

**Device Compatibility:**

**Testing is performed on a variety of mobile devices, considering different screen sizes, resolutions, operating systems (iOS, Android), and device configurations.**

**User Interface (UI) Testing:**

**Evaluation of the app's user interface to ensure it aligns with design guidelines, provides a consistent look and feel, and offers an intuitive user experience.**

**Functional Testing:**

**Verification of the application's functionality to ensure that all features work as intended, and that user interactions produce the expected outcomes.**

**Installation and Uninstallation:**

**Testing the installation process to ensure the app can be installed and launched successfully. Uninstallation testing ensures the app is removed cleanly from the device.**

**Usability Testing:**

**Assessment of the app's usability, including navigation, user interactions, and overall user satisfaction. This ensures the app is user-friendly and meets user expectations.**

**Performance Testing:**

**Evaluation of the app's performance under different conditions, such as varying network speeds, low memory situations, and battery usage. This helps identify potential performance bottlenecks.**

**Security Testing:Inspection for potential security vulnerabilities, such as data leakage, unauthorized access, or other security threats that could compromise user data.**

**Localization and Internationalization Testing:**

**Testing the app's compatibility with different languages, cultures, and regions to ensure it can be used effectively by a diverse user base.**

**Interoperability Testing:**

**Verification of the app's compatibility with other applications, mobile devices, and services to ensure seamless integration.**

**Challenges in Mobile Manual Testing:**

**Device Fragmentation:**

**The wide variety of mobile devices and operating system versions makes it challenging to test on all possible combinations.**

**Network Conditions:**

**Testing under various network conditions to ensure the app functions well in both high-speed and low-speed connectivity scenarios.**

**Platform-specific Challenges:**

**Addressing platform-specific nuances and guidelines, especially when testing on iOS and Android devices.**

**Conclusion:**

**Mobile manual testing is an essential part of the software testing process, ensuring that mobile applications are thoroughly examined for functionality, usability, and performance across diverse devices. Human testers play a crucial role in identifying issues that may not be easily detectable through automated testing, contributing to the overall quality and success of mobile applications.**

**Mobile Testing**

**Web Automation :**

## Automation testing

Automation testing uses some specific tools to execute the test scripts without any human interference. In automation testing, the test automation engineer writes the test scripts or uses the automation testing tools to execute the application. Whereas, in manual testing, the test engineer write test cases and implement the software based on written test cases. It is the most acceptable way to enhance the efficiency, productivity, and test coverage of Software testing.

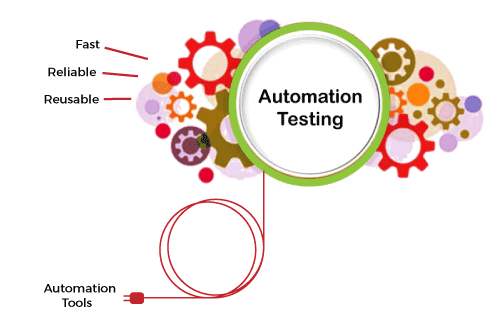
In automation testing, test engineers can execute repeated tasks, whereas repeated execution of tasks in manual testing is a tedious process.

With the help of an automation testing tool, we can easily approach the test data, handle the test implementation, and compares the actual output against the expected outcome.

Automation testing mainly focuses on reducing manual human activity with devices or systems. It increases the efficiency, effectiveness, and coverage of software testing. It saves time and effort.

For the proper implementation of automation testing, there is a requirement of considerable money and investment. There are various automation testing tools available in the market, such as - selenium, Appium,etc,..

### Benefits of automation testing



The advantages of automation testing are given as follows -

### Saves time

Automating the testing process helps the testing team to use less time to validate newly created features. For instance, in manual testing, there is a need to write thousand test cases for a calculator application, but automation makes the process much faster.

### Productivity improvement

As during execution, automation tests do not require human intervention, so testing an application can be done late at night, and we can get the results next morning. Software developers and testers require less time on automation testing.

### Accuracy improvement

In manual testing, there is a chance of mistakes whether you are an experienced testing engineer. The chances of errors may increase when testing a complex use case. But Automation testing reduces the chances of errors. There is good accuracy, as we will get the same result each time on performing the same test cases.

### Test suite reusability

We can reuse the test scripts in automation testing, and we don't need to write the new test scripts again and again. These test cases can be used in various ways, as they are reusable. Reusability helps to reduce the cost and also eliminate the chances of human error.

### Ability to test on various platforms

Automation testing allows the user to test the application on different web browsers and operating systems.

### Running tests 24/7

In automation testing, we can start the testing process from anywhere in the world and anytime we want. It can also be done remotely if we don't have many approaches or the option to purchase them.

### Early bug detection

By automation testing, it is easy to detect critical bugs in the initial phases of software development. It reduces the cost and helps us to spend fewer working hours to fix such problems. It increases the efficiency of the team.

### Less human resources

Automation testing requires fewer people to perform a tedious manual test. To implement the automation test script, we need a test automation engineer who can write the test scripts to automate our tests.

### Reduce the expenses

Automation testing is less expensive, as once the test scripts have been built, we can reuse them at any time without any extra cost. While manual testing is more expensive than automation, with manual monitoring, it is typical to execute experiments repeatedly.

### Scalability of test cases

In manual testing, we require the involvement of the number of people and number of hours to scale up a project. Whereas the scalability of automation testing is higher, we need adding of test executors to the testing framework.

### Consistency

Compared to manual testing, automation testing is more consistent and way faster than executing the regular monotonous tests that cannot be missed but may cause faults when tested manually.

### Fast development and delivery

Automated tests can be executed repeatedly and completed rapidly. We do not have to wait for weeks to execute the tests; few hours are enough for execution. Switching from manual to automation reduces the waiting time and boosts development.

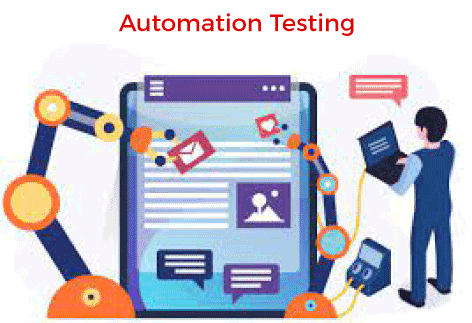
### Easily execution of lengthy and complicated test cases

Execution of bug-prone and complex test cases is easier with automation testing. Test cases with reproducible steps lead to distraction and wrong assurances on testing them manually.

Some of the other benefits of automation testing are listed as follows -

* In comparison to manual testing, automation testing requires fewer resources.
* It makes load and performance testing, stress testing, and reliability testing possible.
* It is more reliable, as it reduces the occurrence of errors. It is reliable because it tests the application with the help of tools and test scripts.
* With automation testing, test engineers are free to focus on other work.
* It improves the testing coverage as the automatic execution of test cases is faster than manual execution.
* Automation testing allows the execution of test cases in a 24x7 environment.
* It enhances the knowledge of test engineers by producing a repository of different test cases.
* Batch execution is possible using automation testing because all the written scripts can be executed simultaneously.
* Automation testing is 70% faster than manual testing.

Automation testing is increasing in software industries. When it is properly implemented, it will have a lot of benefits. It is also kept in mind what tasks are worth automating. The tasks performed a single time should not be automated because the effort needed to automate such tasks usually does not pay off.



Testing is a key area, and applying automation in it will decrease the timelines for SDLC. Using the automation testing tools helps to remove the barriers of manual testing by reducing the development cost and bugs in the application and enhancing the product's quality