

Introduction to Automation Testing: A Guide for Beginners

When it comes to any type of technology the prime goal is that it should be efficient in its work and should not be too expensive and tough to maintain. And so when any developer builds any application or software the aim is to make a flawless product to enhance the user experience, for which any product has to go through multiple testings and have to meet some parameters set by the client.

Now to ensure unbiased and clean testing, which does not take a lot of time and money, simulations have been created to run these tests, which is also known as [automation testing](#). It follows some set of scripts and runs those simulations on the software for some outcome which is further matched with the expected results to ensure the product is working as it should be. And so automation testing not only decreases human efforts but also increases efficiency and saves a lot of time and any margin of error.

This article is a comprehensive guide for everything one should know about automation testing and how to use it.

Table of Contents

- [What is Automation Testing?](#)
- [What are the Components of Automation testing?](#)
- [Difference between Manual Testing and Automation Testing](#)
- [What are the Benefits of Automation Testing?](#)
- [Best practices for Automation testing](#)
- [What are the Challenges in Automation Testing?](#)
- [Conclusion](#)

What is Automation Testing?

In the field of software, different types of simulations play a vital role. These simulations are nothing but automation done through some scripts and software tools to make a process automatically work without any human effort. This also saves a lot of time and increases the accuracy of the work and its efficiency.

What are the Components of Automation testing?

The following are the components of automation testing:

Test Automation Framework:

It is simply the set of rules and guidelines according to which the scripts of automation testing are designed. It also provides all the software tools needed to ensure the test cases are built properly and work effectively. It makes the whole process more structured, ensuring consistency, reusability, and scalability of the simulations created.

Test Scripts:

To ensure the software's behaviour meets expectations, a set of instructions is written in a programming or scripting language for automation testing. These scripts are known as test scripts, and they are then implemented by the automation testing tools.

Test Data:

To execute the automation test cases, the required inputs are taken as test data. Validating the different scenarios is essential, as is ensuring that the software handles and maintains all the inputs properly and detects potential issues.

Types of Test Data are:

- Static Test Data
- Dynamic Test Data
- Valid Test Data
- Invalid Test Data
- Boundary Test Data
- Random Test Data

Automation Tools:

Some applications are needed to execute the automation test scripts and test data and integrate them with other required tools. These applications are known as Automation tools. These tools are specifically designed to manage all the data and report results, which is crucial in order to cover all the testing activities.

Difference between Manual Testing and Automation Testing

The difference between manual and automated testing is that manual testing requires test execution step by step without tools, while automated testing uses automation tools and frameworks to perform tests automatically.

Factors	Manual Testing	Automation Testing
Execution Speed	As each of the tests is executed manually, it takes more time.	The usage of automation tools makes the execution faster.
Accuracy	Vulnerable to errors and inconsistency.	More accurate and consistent as the automation tools run the same steps multiple times.
Cost	In the long term, manual work costs are way higher.	Higher setup costs, but in the long term, the costs reduce drastically.
Test Coverage	Due to time constraints, a very limited number of test cases can be executed with human efforts.	A large number of test cases can be executed quickly within a very short time.
Suitability	Suitable on occasions like exploratory testing where human judgment is crucial.	Suitable for regression testing, load testing, and repetitive tasks, as these do not need any human judgment.
Test Script Development	No test scripts are required, as all the tests are executed manually.	Test scripts are developed in programming or scripting languages for automation.
Flexibility	More flexible in handling dynamic changes and special test scenarios.	Less flexible as any changes would require the test scripts to be updated accordingly.
Scalability	To execute a large number of test cases, a large number of human efforts are required, making it less scalable.	Highly scalable as the test cases can be executed in parallel and across different environments simultaneously.
Maintenance	Easier to maintain, but each test case has to be re-executed for every new test cycle.	To ensure the test scripts stay updated, development and

		maintenance have to be performed regularly.
--	--	---

What are the Benefits of Automation Testing?

Here are the benefits of automation testing:

- **Increased Efficiency and Speed:**
 - Automation testing runs all the test cases in a very short time allowing quick feedback and a faster release cycle.
 - It allows multiple test cases to run parallelly in different environments at the same time, ensuring quality as well, which saves a lot of time.
- **Improved Accuracy and Consistency:**
 - These test cases get executed in a very precise manner in each test cycle, leaving no margin for errors which are likely possible in manual testing.
 - To provide consistent outcomes, automation testing ensures that test scripts are unchanged until they are manually changed. This provides reliability to the whole testing process.
- **Cost Savings in the Long Term:**
 - Human efforts are needed in automation testing to create the test scripts. Once these test scripts are executed, the complete testing process works properly with almost no human intervention, which saves a lot of time and money.
 - Once the automation test cases are developed, they can be used a number of times without any limitations. Just by changing the desired elements within the script, it can also be used for multiple different scenarios
- **Enhanced Test Coverage:**
 - Automation allows the developers to execute the test cases on a very large scale, also paying attention to the extreme edge cases which might get missed by a human.
 - Doing changes in test scripts is very common, though Automation ensures quickly that the new changes do not affect the existing functionality in any negative way which might disturb the structure of the desired software or product.

- **Faster Time-to-Market:**
 - Faster issue and defect fixing is crucial, and so automation testing provides very quick feedback and notifications regarding any potential issues.
 - Integration with CI/CD pipelines allows for continuous testing ensuring quick actions upon every new change in the scripts.
- **Better Utilisation of Resources:**
 - Automation testing saves time and allows testers to work on test cases where human judgment is required.
 - Automated tests can be scheduled as well which ensures the proper use of hardware efficiently.
- **Detailed Reporting and Documentation:**
 - After each testing cycle, automation tools provide a detailed analysis of the test and report the issue and logs it along with screenshots which makes it easier for testers to fix the issues.
 - The detailed documentation of each testing cycle improves the traceability and auditability of automation testing.

Best practices for Automation testing

Following are the automation best practices to follow:

1. Decide Which Tests to Automate

Not all tests can be automated, as some require human judgment. Start your automation plan by identifying tests that will benefit from automation. It is advisable to automate tests with the following characteristics:

- Require repetitive actions with large amounts of data.
- Are prone to human error.
- Need to use multiple data sets.
- Extend across multiple builds.
- Must run on different platforms, hardware, or OS configurations.
- Focus on frequently used functions.

2. Divide Tasks Based on Skill

Assign test suite and case creation tasks to individuals based on their technical expertise. For example, if a test requires a proprietary tool, team members of varying skill levels can create test scripts with minimal effort. However, if using an open-source tool, automation tests will require someone with coding expertise for that specific tool.

3. Collective Ownership of Tests

Do not rely on a single tester or engineer for entire automation testing projects. The entire team must stay updated and involved to integrate automation successfully into the testing infrastructure. This ensures transparent communication and informed decisions about setting up and running the right tests.

4. Remove Uncertainty

Automation aims for consistent, accurate test results. To prevent false positives and inconsistencies, eliminate unstable tests in regression packs and ensure tests are up-to-date. Proper test planning and regular assessments of automated tests are essential.

5. Pick the Right Testing Tool or Framework

Choosing the right tool is crucial for automation testing. Consider the following:

- **Nature of Software:** For web-based applications, use tools like Selenium. For mobile-based applications, consider Appium.
- **Programmer Experience:** Select frameworks, languages, and tools that match testers' expertise, such as Java, JavaScript, Ruby, or C#.
- **Open Source or Not:** Depending on budget constraints, choose open-source tools like Selenium or Appium. Remember, open-source tools are not inferior to commercial ones; for instance, [Selenium Webdriver](#) is highly favored by automated testers globally.

7. Test on Real Devices

Websites must be tested on real devices to ensure they work under real-world conditions like low battery, incoming calls, and weak network strength. If an in-house lab is unavailable, opt for cloud-based testing that offers real devices. LambdaTest is an AI-powered test orchestration and execution platform that lets you run manual and automated tests at scale with over 3000+ real devices, browsers, and OS combinations.

8. Keep Records for Better Debugging

When tests fail, maintain records of the failures, including text and video logs, to identify the reasons for test failure. Choose tools with mechanisms for automatically saving browser screenshots at each test step. This helps detect where errors occur. For instance, BrowserStack Automate video records each test run as executed on the remote machine.

9. Use Data-Driven Tests

For analyzing multiple data points, manual testing is impractical due to the volume of data and variables. Data-driven automated tests simplify the process by using a single test and data set to work through various data parameters efficiently.

What are the Challenges in Automation Testing?

Although automated testing is significantly more efficient than manual testing, it presents several challenges, including:

1. **Learning Curve:** Most automation tools require coding knowledge, leading to a steep learning curve from test creation to maintenance.
2. **Unrealistic Goals:** Setting realistic goals and expectations for automated testing is crucial. It should be combined with other testing and analysis methods to achieve optimal software quality.
3. **Flaky Tests:** Tools like Selenium often face issues with flaky tests, where test scripts break if there are any changes in the application.
4. **Lack of Cognitive Ability:** Automated tests only check the parts they are programmed to check, failing to detect issues outside their predefined scope.

Conclusion

While manual testing has been beneficial and worked for all types of testing for years, automation testing has raised the bar, making the whole process more efficient, accurate, consistent, and time-efficient. Indeed, there are still challenges with automation testing, either due to high investment or skill set requirements, but the opportunities it provides are immensely larger and more profitable.

While putting automation testing over manual testing, it is important to acknowledge the fact that, ultimately, it all comes down to human efforts no matter what, the success of automation

testing entirely depends upon the efforts of the developers who are writing test scripts, creating the test cases and carefully planning and collaborating to make the automation possible.