Automation Testing Tools?

An automation testing tool is a piece of software that lets you define testing tasks and then takes over to perform the tests for you. Ideally, this happens with minimal human oversight.

There is a wide range of automation testing tools to test across multiple platforms, including smartphones, **mobile devices**, and desktop computers. Through **testing automation**, performance testing can run without human intervention, which provides the ability to test code 24 hours a day if needed, speeding up the test execution process.

But that's not the only thing automation testing tools can do. Let's take a closer look at why using an automated testing tool is often the best approach.

why using an automated testing tool is often the best approach.

Benefits Of Automation Testing

Automated testing has many benefits over manual testing, but some cases really exemplify its worth, including:

- Multilingual sites
- Test cases that are repeated
- Tedious cases
- Cases that are broad and expansive

While these tests can be performed by a human, they are quite complex and are therefore prone to errors. For example,

That said, there are some instances where manual testing is better, including:

- New test cases that have not yet been executed manually
- Test cases where the criteria are always changing
- Test cases that are not routine

In these instances, you can see why it would be beneficial to have a pair of human eyes on the testing. For example, the first time a test code is written, it should be run manually to ensure that it delivers the expected result. Once this is verified, it can then be used as an automated solution.

In the cases where automation testing is appropriate, you'll see some specific benefits, including:

- Speed
- Wider test coverage

- Consistency
- Cost savings
- Frequent and thorough testing
- Faster time to market

Now that you know when to use an automation tool and the reasons why you should, let's look at how to choose the right tool for your needs.

9 Types Of Automation Testing

Generally, there are two types of testing. Functional testing tests the real-world applications of the software while non-functional testing tests different software requirements, like security and data storage.

categories, and some of them may overlap. The types of automated testing include:

1. Unit Testing

Unit testing is testing small, individual components of the software. It's the first stage of testing, and while it's usually done manually, it can be automated, so I wanted to include it here.

2. Smoke Tests

A smoke test is a functional test that determines whether or not a build is stable. It verifies the function of essential features to make sure the program can endure further testing. The name comes from the idea that this test prevents the program from catching fire if it's not ready for additional testing.

3. Integration Tests

These functional tests make sure that all of the individual pieces of software test are well when operating as a whole.

4. Regression Tests

Regression tests are both functional and non-functional, ensuring that no part of the software has regressed after changes are made.

5. API Testing

The application programming interface or API acts as the conduit between all the other systems that your software needs to function. It's usually tested after software development to make sure that everything is working together as it should.

6. Security Tests

Security tests are also functional and non-functional. Their purpose is to check everything for security weaknesses that can be exploited.

7. Performance Tests

Non-functional **performance tests** evaluate stability and responsiveness. They ensure that the software can handle stress and deliver a better and more reliable user experience.

8. Acceptance Tests

Acceptance tests are functional tests that try to determine how end-users will respond to the final product. This test must be passed successfully before the product can be released to end-users.

Linear Framework

This type is sometimes called Record and Playback. Testers create a test script for each test case. It's a very basic approach that's more suited to a small team that doesn't have a lot of experience with test automation.

Modular Based Framework

This framework organizes each test case into small, independent modules. Each one has a different scenario, but they are all handled by the framework's single master script. This approach is very efficient, but a lot of planning is required, and it's best used by testers who have experience with automation testing tools.

Top 7 Automation Testing Tools

60% of North American project teams report that they struggle to apply test automation at appropriate levels for Agile development. Employing the appropriate automation testing tool is one way to make sure your team doesn't hit this roadblock.

There are countless automation testing tools out there. Which one is right for you depends on what you're working on and what you need the tool to do. Here are a few of my favorites: