Why Automation Testing is Necessary

Reduce Cost (Of Failure)

A really strong set of test-suites that are executed repeatedly each time a change is made to the code reduces the risk of issues leaking into the field. Automated tests help in finding bugs early in the software development lifecycle, thereby reducing the risk of delivering faulty software.

At the end of the day, delivering a quality product to the market beats any other type of savings and cutbacks.

Save Time

Once the setup is in place, automated tests can be repeatedly run, reducing the time to run repetitive manual tests from weeks to hours.

Once written, the tests can be executed any number of times with no additional cost. The tests are also available 24/7, unlike manual testers!

Accuracy and Reliability

Manual testing is prone to errors because of the number of prerequisites involved in running each test. Additionally, each test may require a different execution sequence.

Manually testers are humans after all, so mistakes are to be expected. This may result in inaccurate results being propagated to the development team.

Automated tests carry out the same steps every time, with precision. The results are generally made available to everyone concerned in the least possible amount of time.

Load testing

Load testing ensures that your application can handle expected and unexpected user loads.

Automated testing can run thousands of tests simultaneously, simulating millions of users, all of which is next to impossible with manual testing.

Realistic load tests must include parameterized settings that are configurable using variables that are randomized and represent what happens in the real world.

Measure Quality Metrics

Extensions and tools available for automated testing provide features to measure a number of code quality metrics such as code coverage (i.e. the percentage of code that is actually tested), technical debt, code semantics check etc.

They are able to measure such metrics because the tests themselves co-exist with the rest of the code. This provides opportunities to measure the quality of huge code bases in a matter of minutes by parsing the source code during an automated build phase. This is not at all possible in manual testing.