Test Automation Advantages and Disadvantages

Test Automation, when done correctly can have many advantages and be very beneficial to the project and organization.

There are however some pitfalls or disadvantages of test automation that we need to be aware of.

Test Automation Advantages

Confirmation of the known

Automated checks are a great way of confirming that the application still functions properly after changes made to it. It is possible that when a new feature is added to an application or a bug is fixed, it impacts the functionality of the working software, i.e. a regression bug is introduced.

By running a set of automated regression checks when the application is updated, we can identify any new bugs introduced as a result of the changes.

The key information here is to run automated checks as often as the application is upgraded. You may not need to run the full set of automated suites, but a quick smoke regression test should be enough to pick up any major issues.

Quick feedback

Another big advantage of automated checks are that they can provide a quick feedback on the health of the application as soon as the application is updated. This is important for the development team, as they should fix anything which is broken before moving on to code other stuff, so a quick feedback is important.

Please note that this quick feedback can only be achieved with unit tests and API tests where the UI is not involved. If we check functionality from UI or at system level, it can take a long time to complete before we can get the results of the tests which is one of the disadvantages of automated testing, so it is best to have few automated checks at the UI layer.

Fast execution of checks

Although automated checks can take a while to script, when we execute them they are generally fast and can go through various steps much quicker than a human being. Therefore, they help with providing a quick feedback to the development team. This is especially true of a data driven test where the same test(s) are executed multiple times but with different data sets.

Frees up the time of the testers

Regression tests which need to be run on a regular basis, when automated, they free up the time of the testers, so they can focus more on interesting scenarios and exploratory testing.

By the same token, when implemented properly, automated checks can run automatically with minimum or no supervision or manual intervention.

The development team can contribute

Automated checks are usually written in the same language as the application under test. For this reason the responsibility of writing, maintaining and executing the tests becomes a shared responsibility and everyone in the development team can contribute, not only testers.

Test Automation Disadvantages

False sense of quality

Beware of passing tests! This is specifically important in verifying functionality at UI or System level. An automated check only checks what is been programmed to check. All automated checks in a test suite can happily pass, but there could be major flaws undetected, because the automated check was not supposed to “look” for those failures.

Solution: ensure you design good test cases before automating them. In system level or UI automation, make sure to check for important elements as many things can happen as a result of an action. Also complement automated checks with manual / exploratory testing.

Not reliable

Automated checks can fail due to many factors

If automated checks keep failing due to issues other than genuine bugs, they can raise false alarms. Automated checks can break because a minor UI change was implemented, or a service is down or there are network issues which are not relevant to the application under test but could impact the automated checks.

It is best to aim for minimum number of failures during each run so you can get a reliable feedback from the automated checks.

Solution: Where possible / applicable, use stubs to overcome issues with connectivity or changes in the 3rd party systems so the automated checks would be independent of any downstream failures.

Automation is not testing

Unfortunately, many people mistake “Test Automation” with Testing and once they have the tools to automate the testing, they want to “automate all the tests” and get rid of so called manual testers or QAs.

The truth is that Testing is an exploration exercise; you need domain knowledge, be inquisitive with a mind to focus on learning the behavior of the application and apply proper test techniques to be able to spot anomalies in the software.

Testing is not just executing a set of pre-defined test steps and comparing the actual results with expected results; that can be the job of automated checks. But to properly test an application, a human intelligence is always required.

Solution: Understand that for a successful delivery of a project you require both automated and manual testing. One is not a replacement for the other; complement automated checks with manual / exploratory testing.

Maintenance Time and Effort

You have to accept the fact that if you have automated checks in place, you will need to spend the time upgrading the relevant tests as the application is upgraded. If the regression packs are not kept up to date, you start seeing failing tests that fail due to upgrades rather than identifying real bugs, as well as tests that are no longer applicable.

Embarking on test automation is not a one-off effort. In order to get the most benefit from the automated checks, they have to be kept up to date and relevant, which requires time and resource.

Solution: Because the maintenance factor is an ongoing activity, make sure you design in a way to reduce the time you spend updating the tests. Use reusable modules, separate the tests from the framework and use good design principles to alleviate the maintenance effort.

Slow feedback

Occasionally automated checks can take a long time to script depending on the complexity of the test. When a functionality is ready to test, sometimes it is quicker to do a manual check rather than waiting first to automate the test and then run the test and check the results.

Also, in terms of UI and System level automated checks, they can take a long time to complete so the results of the overall testing may not be known for several hours in some cases.

Solution: Try to automate the tests alongside development so that when development is complete, you can run the automated tests against the new functionality.

Also, separate the automated checks in different packs, such as a smoke pack (which should take no longer than 15 minutes) which checks the most important functionalities of the application as soon as the application is updated and run the full regression pack (which can take several hours) overnight, so you have the results ready by next morning.

Not many bugs found

Majority of the bugs seem to be found by “accident” or when performing exploratory testing. This is probably due to the fact that during each exploratory testing session we could be testing the application in different ways thus finding new loopholes through the application.

Automated regression checks on the other hand always follow a given path and sometimes with the same set of test data. This reduces the chance of finding new defects in the application. Also the number of regression bugs seem to be lower than new-feature bugs.

Solution: Try to build randomness in the scenario and data. Trying different paths with different data each time can reveal potential issues.