

# How to avoid 7 common mistakes in test automation

January 14, 2020 [Amit Sikka](#)

Test automation is increasingly being adopted by organizations to take over the tedious, time- and resource-intensive daily testing of systems and applications. However, there are a handful of errors a tester can make that may not only cause the test automation to fail over time, but can also impact its ROI.

Here are some of the most common test automation mistakes and ways to avoid them:

## 1. Wrong Tool Selection

One of the primary reasons software test automation projects fail, or are unable to yield the desired level of efficiency, is selection of the test automation tool. While there are a variety of factors that lead to poor decision-making and tool selection, the most common are as follows:

- The testing requirements of the application under test (AUT) are not analyzed thoroughly.
- The test tool requirements are not laid out clearly.
- The skillset or readiness of the test team is not accurately assessed.
- Tool vendor and capability evaluation is not done or is done poorly.
- The cost benefit analysis is not performed, or the tool was selected solely because it was an 'open source'.

Organizations can easily avoid above mistakes by enforcing a thorough tool and vendor evaluation process. Teams that are going to use a tool for automation should also have a part in the evaluation process for smooth adoption. While it's true that one tool rarely meets all your needs, a detailed evaluation and selection process can ensure that most of your critical needs are met.

## **2. Insufficient Test Validation**

Data validation is a critical aspect of testing. Test engineers often make the mistake of not validating a scenario at all levels. For example, a functionality may appear to be working perfectly fine when investigated at the user interface (UI) level. But behind the scenes, at the database level, it may not be ensuring the expected data-integrity and therefore can lead to major failures in the system.

To avoid such errors, test automation scripts must be designed to validate functionality at all levels – not just at the UI level. Restricting the validation to only visible UI elements (buttons, texts, hyperlinks, combo boxes, etc.) will always be prone to bug leakage in AUT production.

## **3. Ignoring the CI/CD Pipeline**

The objective of continuous integration and continuous delivery (CI/CD) is to speed up the software release cycle. It enables teams to continually integrate any small incremental code changes, test them quickly, and make them available to end users. test automation is a critical step in the whole CI/CD pipeline.

Often teams miss integrating test automation with the CI/CD pipeline, which ultimately means that they are not leveraging the full value of automated tests or the CI/CD tool. Test teams must create automated build acceptance, smoke and/or limited regression test suites, and integrate them with their CI/CD pipeline to quickly deliver quality releases to the market.

## **4. Automation Maintenance**

Any test automation requires regular maintenance, whether it's handled automatically by the tool or via manual updates. Many professionals in this space either do not consider the effort for maintenance when selecting an automation tool, the automated scripts are not designed with an objective of keeping maintenance overhead low or maintenance is completely ignored for a prolonged period – all this effectively shrinks the test automation coverage and progress.

While selecting the automation tool, one must keep the maintenance overhead in mind. A tool that offers self-healing capability is obviously better than the one that doesn't. Even with self-healing tools, test teams must keep up with any incoming application changes and ensure that the automation scripts are kept up-to-date.

## **5. Attempting to Replace Manual Testing**

It's easy to think that automated testing can solve all testing problems. The reality is that automated testing is in no way a replacement for manual testing. Some things are better verified by a human eye. For example, usability and exploratory testing are better left to manual test teams. These teams must be clear on what should or should not be automated and when to run the automated test cycles. Automation should always be considered as a helping hand to manual testers.

## **6. Record/Play Trap**

Most of the modern automation tools offer 'record and play' features that allow users to quickly create automated scripts for their scenarios. While this is exciting, the trap is that these automation scripts are built with static data, which is generally not reusable. Additionally, tools do not record any validations that are done through the naked eye. Moreover, every time there is a change in the recorded scenario, test engineers will have to re-record it. Re-recording can wipe out any dynamic data, which would then need to be imported again manually.

Most test automation tools offer built-in record and play features that allow users to essentially translate user actions on the application under test into automated test scripts. However, it gives a false impression that one can automate their test scenarios in very little time, which is most definitely not the case.

This feature should be used for creating the base script only and also for training the novice automation engineers. For robust, long lasting scripts one must avoid this record and play, while selecting a tool or automating with a long-term view.

## **7. Misplaced Priorities**

Lastly, test automation users often have misplaced priorities, from automating tests that are not often repeated and low-value use cases first, to automating tests for an application functionality that has not been stabilized or is not as critical. Automation engineers generally tend to automate the simplest test cases ahead of others in order to show good progress.

There are many ways to automate tests and tools available to help ensure precision and speed up time-to-market. Test automation is critical and should be consistent with your business operations. However, do not rely on short-term wins. Align your business with the right test automation tool(s) and experts to increase test coverage and ROI.

---



**Amit Sikka** is the associate director and testing practice head of AgreeYa Solutions, as well as an active software testing evangelist. He has over 20 years of professional experience, with a specialization in test process assessment, automation, mobility, performance engineering and enterprise application testing for various multi-industry global clients.