

Pitfalls of Software Test Automation

John Morrison

<http://www.techmanageronline.com>

Organizations make significant investments in Test Automation. However, few of them realize corresponding benefits from their investments. Often, automation suites turn out to be expensive to create and subsequently maintain without being effective in unearthing defects. We'll look at some of the common pitfalls encountered with test automation.

1. Quality of tests being automated.

“Garbage in, Garbage out” applies well to test automation. If your test automation is based on tests that are themselves of poor quality, then you can rest assured that your automation results will hardly be spectacular.

While analyzing issues with test automation, we must examine the underlying tests and their design. For most organizations, their test automation goal is to automate 100% of their existing manual tests. While it may be a challenge to reach this goal, especially in larger and complex projects, the usefulness of such a goal is debatable. Unless you have a good set of tests to automate, merely trying to convert all of your manual tests to automated ones, does not offer much benefit. The exercise could very likely incur significant investments of time and resources while yielding little effective returns.

2. Automation Framework and Process.

For many groups, test automation begins with the acquisition of a tool and from thereon it proceeds towards a race to automate as many tests as quickly as possible. Everyone's excited about the tool and its features, so much so that each one lets loose their creative energies on automating their set of tests quickly. During this time, some attributes of test automation such as – reusability, maintainability, scalability, not hard-coding data into the tests and such others are often ignored. These attributes unfortunately, are required for a robust and maintainable test automation suite.

An oft repeated aspect of test automation is that you must approach test automation just as you would a product that your organization produces. This implies that the same principles and practices need to apply to test automation; the same level of detailed planning, designing, architecting, process definitions, reviews, etc. need to apply. Test

automation is software that you are developing and a reason that automation can fail is because most organizations do not treat the test software they develop with the same level of care and commitment as they do to the products they sell.

Dis-organized test automation or automation done with little attention to the required attributes, results in a suite that can be a maintenance nightmare. Groups soon realize that they need more resources to both maintain the automated suites as well as to automate newer tests, resulting in ever increasing costs and little added benefit. It will be useful to remember that all automation needs to be either maintained or abandoned. Make sure your automation is easily maintainable at low cost.

3. Stability of the System being tested and the ability of the automation suite to adapt to changes to the System

There are two parts to this – one, the system or application that is being tested needs to be in a relatively stable state before you embark on an extensive automation exercise. If you try to build your automation suite when the system on which these automation tests are based upon keeps changing frequently, you are guaranteed to encounter – false test results being reported by automation runs and ever increasing maintenance activities on your suite. We must however, realize that despite calling for “relative stability”, there will be some changes and any test automation exercise must accept and be open to some changes. Having a stable product can help minimize not eliminate changes.

This is where we look at part two, which is the ability of the test automation suite to adapt to changes to the product it is testing. When test automation is not based on a sound framework that can minimize the impact of changes to the system under test, extensive maintenance activities need to be performed. This can have the added effect of affecting stakeholder perception of both the reliability and usefulness of your automation suite.

From a process stand-point, when major changes are planned to be made to the system under test, these must be reviewed to evaluate their impact on existing test suites. Any resulting test automation activity must be planned and communicated appropriately.

Having looked at some of the pitfalls of test automation, we should analyze an important element in automating tests, the people involved in automation. Who automates your tests ?

Ideally, a person who is chosen to automate tests should have sound development and testing skills. Often times, people are chosen who might have expertise in either of these areas. This can be worked upon. If your test automation engineer is predominantly from a testing background and lacks development skills, you need to check if this person can pick up the required development skills and demonstrate the necessary rigor and discipline.

Generally, test engineers who desire to be programmers, jump at the opportunity to automate tests. This provides them an opportunity to develop their programming skills. While doing this, one must make sure that these engineers are well grounded on the automation requirements and focus on delivering them rather than getting carried away with the programming task. Sometimes, you may not need to automate all your tests given the available time and resourcing. Here, the ability to focus on those areas that can provide maximum return on investment is important.

A reason for requiring development skills is the need for focus on aspects such as maintainability, re-usability, reliability and importantly, the ability of the test automation suite to easily adapt to changes to the system under test. Test automation needs to be approached with the same rigor and commitment as a regular software development project.

Sometimes, developers get assigned to do test automation. In this case, an evaluation needs to be done to ascertain if this person can gain understanding of the testing discipline and develop a healthy respect for the function. When in-house engineers aren't available, it may be the case that contractors are employed to perform test automation. The advantage here is that you are likely to get engineers with matching skill-sets.

However, one must be aware of & take appropriate steps to mitigate any maintenance challenges once the contractors have departed. Having a sound automation framework in place and reviewing conformance to stated standards and guidelines can be useful.

In some groups, new college graduates, interns or junior programmers are assigned to work on automating tests. The argument may be that these folks have spent a lot of time learning about programming in college and would be able to apply their “fresh” learnings to automating tests which is a type of programming activity. Also, these new comers often are glad to have the chance to demonstrate their competence and creativity and show interest. In some cases, it may be that the management wants to keep these junior folks engaged on some relatively lower risk activity and hence assigns automation to these engineers. If what they develop, works – good and if it doesn't it can be shelved with low cost incurred. The new comers often are not well trained on the testing function and often end up using their programming skills to creating complex test automation suites that can be hard to maintain without meeting testing requirements.

Also, in some cases there is an attempt to assign automation to the “lower-performers” from development or testing while engaging the “better performers” in doing other tasks that are deemed to be more important. These folks have poor skills, commitment and motivation in either area. The chances of success in this case is very low. Test automation requires skilled, committed and motivated folks who can work effectively with both development and testing teams.