Technical versus non-technical skills in test automation

Dorothy Graham
Software Testing Consultant

info@DorothyGraham.co.uk

SUMMARY
In this paper, I discuss the role of the testers and test automators in test automation. Technical skills are needed by test automators, but testers who do not have technical skills should not be prohibited from writing and running automated tests.

Keywords
Tester, test automator, test automation, skills.

1. INTRODUCTION
Test automation is a popular topic in software testing, and an area where a number of organizations have had good success. Tests that may take days to run manually can be executed in hours, running overnight and at weekends, with greater accuracy and repeatability. Tests can be run more often, giving immediate feedback for new builds.

Yet despite the obvious potential, many organizations are still struggling to achieve good benefits from automation. I believe that one reason for this is the role of the “test automator”. There is a common misperception that testers should take on this role. This paper explains why this may not be the best solution.

It is popular for testers to be encouraged to develop programming skills. For example at EuroStar 2012, a keynote speaker advised all testers to learn to code. I don’t agree with this, and this paper, originally written for the CAST conference 2010, explains why.

2. TERMS
I will start by defining the terms I use in this paper.

Test automation: the computer-assisted running of software tests, i.e. the automation of test execution.

Test automator: A person who builds and maintains the testware associated with automated tests. [4]

Tester: A person who identifies test conditions, designs test cases and verifies test results. A tester may also build and execute tests and compare test results. [4]

Testware: The artifacts required to plan, design and execute tests, such as documentation, scripts, inputs, expected outcomes, set-up and clear-up procedures, files, databases, environments, and any additional software or utilities used in testing. [4]

3. TEST AUTOMATION SKILLS
3.1 Existing perceptions

The automation of test execution is a popular application of computer technology to itself. There are a number of books about test automation. [1,2,3,4,7,8,10,11,12] Many of them do not appear to mention skills needed (or it was not obvious if they did). There is a general perception that testers must be or become technical, i.e. programmers, if they are to become involved in automation, although there are a few exceptions that mention a distinction between testers and automators.

Linda Hayes in her useful booklet on automation [7] says: “… developing test scripts is essentially a form of programming; for this role, a more technical background is needed.” She distinguishes between “Test Developers” i.e. testers, and “Script Developers”, which is part of the role of a test automator.

Dustin et al in [3] says: “When people think of AST [Automated Software Testing], they often think the skill set required is one of a ‘tester’, and that any manual tester can pick up and learn how to use an automated testing tool. Although the skills [of a tester] … are still needed to implement AST, a complement of skills similar to the broad range of skill sets needed to develop the software product itself is needed.” (p 225)

A paper by Mosaic [13] mentions three roles: “Manual Test Engineer”, “Automation Test Engineer” and “Lead Automator”. In this model, the design of tests (i.e. the tester’s role) is done by both test engineers; the automation work (i.e. test automator’s role) is done by the lead automator and automation test engineer. The key distinction is who designs the tests, which in my view is best done by the tester, but collaborating with the test automator for tests that are to be automated.

3.2 Is test automation a technical task?

The answer to this question depends on what you include as part of “test automation”. If you view it as the direct use of a test execution tool, i.e. writing, editing and running scripts written in the tool’s scripting language, then it is a technical task, and programming (i.e. scripting) skills are needed.

Another technical aspect of test automation is the design of the testware architecture – the structure and relationship of all of the items of testware that comprise the artefacts required for automated tests to successfully run. The design of the testware architecture is a critical aspect for successful test automation, and the skills needed for this include technical expertise, as well as knowledge of how the tests are to be used. The person who designs the testware architecture may be called a test automator, test architect, or lead automator.

3.3 Constructing automated tests is not entirely a technical process

The construction of the automation architecture, and the scripts and other testware that will be used to run automated tests is a technical task, but automated testing is not just the structure of the architecture and scripts.

The whole purpose of test automation is to make it possible to run tests with minimal human involvement in test execution (and comparison).

There is a need for testers to be able to use automated tests, both to write tests to be run automatically, and to run those tests and view the results. The tests that are to be automated could be technical tests, such as those written by developers as part of Test-Driven Development or unit or integration testing,
but system and acceptance tests can also be automated, and the testers who write those tests are not always technical (i.e. software developers).

The content of the test needs to be determined, but this is a task that is done by a tester; the implementation of the test is what is done by the automator.

4. TESTERS TO AUTOMATORS?

4.1 Testers become automators?

I have seen it work well to have a team of manual testers embarking on an automation project, where all (or nearly all) of the testers effectively become programmers, i.e. test programmers, or scripters. At a former colleague’s company, five out of the team of six testers went on the tool vendor’s training course and became familiar with the tool’s scripting language. One tester decided he didn’t want to become technical, so he concentrated on manual testing, but the others all became good test automators. There were two interesting side-effects of the testers’ newly acquired skillset. First, they had a lot more sympathy for the developers, as they now understood first-hand the frustrations of trying to get the computer to do what you wanted it to do. Second, they found that the developers treated them with a bit more respect, as they now also had some development skills. This led to a better relationship between the developers and testers.

Another example where it worked very well to have all of the testers become automators is described in a chapter by Lisa Crispin [2] in our forthcoming book. An agile team of 9 to 12 people were all involved in doing manual regression testing, so were highly motivated to automate 20% of their work, and everyone became involved in the automation.

4.2 A separate team of test automators?

I have seen other organizations where a separate team is set up to automate tests, leaving the testers free to concentrate on designing tests and running manual tests. As the automation team gets going, they automate tests nominated by the testers, freeing the testers from having to do those tests manually. The automation team provides a service to the testers, designing the testware architecture and structure of the tests, and assisting where needed when problems are encountered with the automated tests. For example, if an automated test fails, it could be because of a software fault (in which case the tester would have found a bug), but it could fail for a technical reason such as a problem with the environment, a missing testware item (i.e. a bug in the automated testware), or a problem with the tool itself. The tester, not being technical, will need technical assistance to identify the source of the problem.

So we have the situation where test automation does require technical skills, but we have testers who do not have those skills – can this really work? Yes it can, but it needs two key separations or layers of abstraction.

5. AUTOMATION SUCCESS NEEDS LAYERS OF ABSTRACTION

5.1 Technical Layer

Technical aspects are very important for test automation. A good testware architecture will have two layers of abstraction [6]. The technical layer will implement good software development practices for
the testware, separating the tool itself and the direct scripting of the tool from the software or scriptware that calls and uses the lower level scripts. Modularity and reuse are key factors in minimizing maintenance of automated testware. If something changes in the software, the testware will need to reflect that change. With lower levels of scripting (a recorded test or linear script being the lowest), a small change to a screen can result in making “magnetic trash” [9] of the automated tests.

If possible, the testware should be designed so that it can cope with changes in the software under test without needing any changes to the testware. If this is not possible, the effects of any change to the software being tested should be confined to only one testware artefact (or a minimum number if this is not practical).

This layer gives good maintainability to the automated test regime.

5.2 Tester Layer

If all of the testers are technical, such as developers who are doing Test-Driven Design or unit testing, then this layer is not as critical. The Tester layer of abstraction is needed when system testers or user acceptance testers want to use test automation, but do not want to become technical, i.e. programmers.

In order to achieve this, the non-technical testers must be able both to write tests (that can then be run automatically) and also to run tests, i.e. to “kick off” a set of automated tests.

If the testware architecture uses a keyword-driven approach [1,4,5,6], the testers can write tests using keywords that are related to the business knowledge or domain knowledge that they are familiar with. Yes, they do have to follow the correct syntax for the keywords, but tools enable this to be relatively easy to do, for example by providing a drop-down list of valid keywords and checking the syntax of parameters entered to the keywords.

The keywords are implemented (i.e programmed) by test automators, using the scripting language of the tool, or using any other programming language that they know and would be appropriate. The testers are not involved in the implementation of the keywords, but they are able to use them to write tests.

The testers also need to be able to select a set of tests to be run automatically. This can be implemented by the test automators to make it easy for the testers to kick off a set of tests, for example by providing options in a user-friendly interface to the automation.

The testers also need to receive and understand the results of the automated tests, and the way in which this information is communicated to them is also designed by the test automator.

This separation of the tester from the automation is needed for the automation to grow within an organization and to give long-lasting benefits and wide-spread acceptance.

6. SUMMARY AND CONCLUSION

Test automation does need technical skill – for those who are closest to the tool itself.

The skills of the tester and the skills of the test automator may be found in the same person, but it may work better to have different people performing the two roles.
The test automator’s role is critical in establishing a modular and well-structure testware architecture, separating the tool from the testware, and providing a tester-friendly interface to the testware for non-technical testers.

Not every tester can or should become a test automator. Many non-technical people are very good testers; they should be able to use test automation without needing to have technical skills. Getting to this point, however, does require good technical support, but that support does not have to be provided by the tester.

7. REFERENCES


    action=ctu&id=5550465&pv=f=pp&authToken=7sp6&authType=name&trk=pp_get Intr&link=cnt dir


