



Test Automation: Delivering Business Value

Last Updated: 31st October, 2008



Introduction

As organizations look to reduce costs across a wide range of operational disciplines, IT will become under the radar more and more. The impact of project slippage, poor quality delivery and inadequate solutions will not be tolerated. This may prove a pain-point for many organizations who have traditionally failed in these critical areas.

Testing, like all project disciplines, has areas that can be quickly improved within an organization. One of the key areas where cost-savings and project efficiencies can be generated is through the use of test automation. The ideal is to reduce time and cost and improve quality. However, to maximise the benefits across all three needs some simple principles to be followed. This paper is aimed at lifting the lid on what makes test automation a viable option for an organization and recommends some sound advice in the implementation of a test automation strategy.

Manual or Automated Testing

Essentially there are two categories of testing – manual and automated. Each has its own advantages and disadvantages. Actually, “Manual or Automated Testing” is the wrong question. There is always manual testing, and it’s absolutely essential. Therefore the question is really “Should we include automated testing or not”?

The most common reason for not deploying automation is the time required to implement it. Companies can struggle to find the time to perform existing manual tests and as

such will not have the additional time it takes to automate the scripts. The tools require financial outlay - in addition to the manual testing budget - but if utilized properly, give significant benefits. The costs of manual testing remain pretty much constant, while the cost of automated testing reduces every time the scripts are used throughout the development cycle and into testing post implementation.

Automated test tools are sold on their ability to record and replay tests, so scripts can be built in a very short space of time. However, hundreds of scripts will usually be needed which can take a significant amount of time to develop. Unfortunately these scripts often do not even play back straight after they’ve been recorded. The scripts are unstructured and difficult to maintain, with embedded data which can take no account of changes to the data in the Application Under Test (AUT). These scripts can be parameterised to make them more flexible, but this takes time and therefore additional investment.

To really make test automation work we need scripts that are reusable, robust, and reliable. More importantly, however, the scripts need to be modular in a framework. The scripts should be data driven, using external data in spreadsheets or a database that the business users can control.

To identify whether automation is appropriate for any given project we need to understand the benefits of each type of testing, and therefore when each method is most appropriate. The table below highlights this:

Benefit	Automated Testing	Manual Testing
Productivity	Automation’s greatest talent. Gives confidence that new functionality has not changed the behavior of the existing code. Takes the tedium away from repetitive tasks and assures accuracy.	Very time consuming. Mistakes can be made. As the release iterations increase, so does the size of the regression pack, meaning more time is required for regression testing every release.
Repeatability and consistency	The tests can be exactly reproduced for regression purposes but can also be exactly repeated for deeper diagnosis by the developers.	Input data is not always recorded, and the Known Start Point of the test is not always adhered to, so the exact circumstances of a failure may be difficult to reproduce manually. Moreover, the exact coverage achieved with manual testing will vary from release to release unless each test is defined as detailed steps and executed in exactly the same order every time.

Benefit	Automated Testing	Manual Testing
Speed of Execution	The scripts run as fast as the AUT will allow. Allows testers to be more productive, and to improve the test pack's coverage.	Human testers are a bit slower! Testers cannot create new or improved test cases while they are executing tests and recording the results, unless the size of the team is increased.
Maximizing Machine Resources	Tests can be run out of hours at no additional cost.	Again, it is expensive to employ a team of testers out of hours. Manual testing ties up machine resources (including shared resources such as network bandwidth) during the day.
Results Recording	Test results are automatically recorded for all outcomes. Test success evidence is available for audit purposes.	Test results recording and storage can be very time consuming, and may get missed. Commonly test results are only recorded when there is a failure. Furthermore there is always the potential for testers to mistakenly record results against the wrong tests or even intentionally "pass" manual tests without spending the time doing so.
Ad Hoc and Exploratory Testing	Not suitable for ad hoc testing (although this is a possible use for record and replay).	Ideally suited to ad-hoc and explorative testing.
Dynamic Applications	Automated scripts need a high level of maintenance for highly dynamic applications, although this can be minimized by the use of modular, reusable scripts and functions.	Easy to adapt to dynamic applications, but diligence is needed to ensure that the manual scripts are updated accordingly.

Make the most of your test automation tools

All too often businesses looking to cut costs only think of the short term. In this light, test automation tools are either dismissed as being too expensive, or bought on the strength of the supposed quick returns of record and playback. The immediate paper-savings are obviously attractive, but if the costs of manual testing versus automated testing over, say, a three year period are compared, then the actual cost savings are far greater.

Don't make false economies. For example, choosing a test automation tool because it is the cheapest could end up being far more costly.

Define your requirements, for example Management Information. If the test management tool cannot deliver

reports to your requirements, then other tools and methods will have to be employed, the cost of which has to be added to the cost of the tool. You could end up using the test management tool purely as a test asset repository. If this is the case then the reporting and repository aspects could probably be implemented at no additional cost if standard office software that already exists is used such as Windows Explorer to set up folders as the repositories, and Microsoft Excel for reporting.

Test automation should not be looked at project specifically, but as an enterprise wide asset. The most realistic way forward is to prove it on a few projects first. Pilot projects should be encouraged as a precursor to a Center of Excellence for automation where a centralized team of automation experts can be tooled and skilled up and shared amongst several projects. The experience and expertise built up can then be readily adopted by other

projects. With this mind-set it is more likely that testing will be incorporated earlier and more fully, into the software development lifecycle.

Plan your Approach to Test Automation

Make life easy. A set of manual test scripts will probably already exist. Use these as the basis for the first cut of regression tests, and for creating the smoke test scripts. When writing the manual scripts, bear in mind the possibility of automating them. For example use spreadsheets to store the data – then they can be incorporated straight into the data driven tests.

Set out guidelines and standards for the structure of the automated test scripts and the test assets.

Start small. First automate the least business-critical tests. It is better to look at those where there is certainty that the user interface is 100% stable in terms of screens and controls and amongst these, those that take lots of time because they are iterative in nature should be automated first. The manual test scripts for the first iteration become the first set of automated regression scripts for the second iteration of the AUT.

Once the initial regression pack is in place, the manual testers can concentrate on writing tests for new functionality. These tests will subsequently be incorporated into the automated regression pack, augmenting its coverage more

and more over time. To increase coverage with automated scripts more quickly, take existing scripts and adapt them for varying business scenarios.

Modularize the scripts, and create function libraries in order to eliminate code redundancy, maximise reusability and maintainability.

Incorporate error handling and reporting functions into the reusable modules. This saves coding time and makes the scripts more consistent, robust, reliable and maintainable.

Do not forget to document the scripts and function libraries. If the test engineers don't know what's available, then inconsistency and code redundancy will ensue, affecting the maintainability.

Conclusion – it makes sense to automate

With planning and proper use of automated test tools, the cost of testing can be dramatically reduced, and the resulting quality and reliability of the software will bring even more cost benefits to the business.

Used strategically, test automation will reduce the cost of testing to the business directly, by reducing the time to test, and indirectly by improving software quality and reliability across projects. It should be seen as an investment and not an overhead.