



Original Software

Three Essential Steps Towards Implementing Successful Software Automation.

Colin Armitage.



Introduction

Today, the pressure on IT departments to rapidly deliver quality applications that support the business needs has never been greater. Application testing represents a key microcosm of this overall picture. On one hand the need to invest in testing has been demonstrated as essential due to a greater appreciation of the full cost of application failure in an ever increasing risk averse environment. On the other hand, effective testing takes significant time and impacts key resources both inside and outside the IT department.

Enter software test automation, originally conceived in an effort to try and resolve this conflict of quality and time that has been the scourge of even the most diligent of IT departments.

The last twenty years have seen numerous software products and testing methodologies designed and developed in an effort to attempt to offset manually executed testing. Some methodologies have come from industry sources while others have their roots in academia, what they all share is the need to invest considerable time, along with hiring expensive skills, in developing an initial library of testing assets. Unfortunately the outlay doesn't end there with a comparable level of effort needed every time the application under test is updated.

The reality is this - business applications simply do not keep still while 'test assets' are developed and redeveloped to keep up with every tiny change and update. The scripts that form the base of these tools are complicated and require specialist programming skills to ensure success. This constant re-scripting requires a lot of time and resource and IT departments struggle to cope with the re-design work needed by the very tools that were promised to cut their workload in the first place. In many organizations test automation is now an oxymoron and the solutions lie on shelves gathering dust while those who never invested rub their hands believing that by doing nothing they've become better off.

But the pressures to reduce the time to market and improve application quality will not simply go away. Do nothing is not an option so what is the alternative? Business pressures dictate that test automation offers the only realistic solution but to make sure that the 'solution' is better than the 'do nothing' alternative we need to step a significant step back.



A New Way Forward

Let's start by selecting an approach based on what IT Departments need, not on what technology has historically offered. We believe that the key requirement is to align effort with benefits while embracing the other requirements at every stage. We call this approach Crawl, Walk, and then Run to testing success.

Crawl

When a baby learns to crawl it is a defining moment in its young life. No more constrained by his inability to self-propel, the world suddenly opens up and becomes an exciting and stimulating place of discovery and achievement. Similarly in the world of testing, small improvements in certain key areas can make massive differences, and there are two candidate areas that will immediately yield significant benefits for limited effort: planning and manual testing.


We all know that poor planning prevents proper performance but unless definitions are in place on exactly what needs testing, and success criteria is documented and agreed upon how can any test be performed to a satisfactory level? Indeed the chances of successful implementation are low. It is therefore wise to invest in a planning and test management solution that can act as the communication backbone for both the quality and development teams. Scrappy bits of paper and inventive spreadsheets may appear a cheaper alternative but they are a partial, short-term sticky plaster for a strategic need.

The second candidate is manual testing. It represents the bulk of current effort and even with "full automation" in place manual testing will continue to play a significant part of your QA process for ever. (Industry research suggests that a maximum of 30% of all testing is automated). But does manual testing have to be so...well, manual? Thankfully not any more.

A new solution called *TestDrive-Assist* is taking the QA world by storm. TestDrive-Assist offers full support for manual testing, requires minimal training, and delivers immediate benefits in terms of test logs, error annotation, database tracking, and error reproduction material. Manual testing is no longer the bottle neck it used to be, timesavings can be remarkable and tangible benefits are immediate. Indeed, just as the life of the baby is transformed when he is able to crawl on his own, so the life of the QA department is transformed when they free themselves from the shackles of manual testing. The Crawl phase is important not only for the benefits it rapidly delivers but also because it starts to free some test cycles to consider where the next biggest payback can be achieved.

Walk

For many IT departments getting control of the data that is used for testing is a prime candidate. Many will be using entire copies of their live database – not an ideal scenario with the inherent need for disk capacity and the detrimental effect this will have on test times. Also, the regulatory environment is strengthening every year and it is now no longer possible to simply ignore issues such as data confidentiality. Of course a key benefit gained by properly controlling and managing your test data is that this is an essential prerequisite to successful full test automation. Effective test data extraction from live databases will allow the running of meaningful tests over live data without compromising live information – allowing for clear understanding and verification of test results.



Unfortunately, while most current products pay lip-service to the database with some support for verification queries, the overwhelming focus is on the visual layer. It is important to keep in mind that just because you reach a screen telling you an order has been received, it doesn't necessarily mean that everything is OK – you may have corrupted the whole database in the process and be blissfully unaware that anything is wrong... until it is too late.

The requirement then is for a solution that can help create a representative, coherent subset of the live database and then crucially protect it so that it is always available as a predictable starting point for each test run. With our TestBench solution this requirement is met.

Run

Finally, with solid foundations now in place we reach full test automation, and this is the proper sequence. Full automation was never a valid starting point and even as the final step it would have limited chances of success without the building blocks created in the Crawl and Walk phases.

But we now know that the previous incarnations of test automation that require specialized skills and create an unsupportable maintenance burden have not worked. Business imperatives demand that we empower the subject matter experts, fortunately technology has moved on and the TestDrive suite of solutions offers the full power of historic code-based solutions without the need for technical skills. This approach critically also addresses the fundamental requirement of script maintenance as it enables and delivers full “self-healing” of the test assets. When scripts are run over an updated application they will update themselves to reflect any changes – meaning the end to time consuming re-scripting.

In Conclusion

The big benefit to adopting the CWR strategy is the immediacy of benefit. By starting at the very beginning – with manual testing or planning - significant benefits can be enjoyed almost immediately, with minimum cost and set up. At every stage of this strategy there are unique capabilities that sets it apart – whether it is the unique capability to test and manage data or code free testing – the benefits are easy to see, and, more importantly, easy to realize.

Colin Armitage is the CEO of Original Software.

For more information visit www.origsoft.com