The Battle for Software Quality

A guide to conquering the challenges and becoming victorious

An Original Insight
Clevel and Senior IT

Original Software

> **“Because testing is still so inefficient, it invariably accounts for a unacceptably large portion of a project’s time and budget”**

Introduction

Most IT executives agree that any company that is able to rapidly deliver software applications of a consistently high quality within minimum budgets will enjoy significant competitive advantage. However, despite this, it is generally accepted that the challenges surrounding software quality remain untouched. Testing is still perceived as a huge bottleneck. Despite vast sums having been spent on big branded automation tools, the large majority of testing is still carried out manually.

Because testing is still so inefficient, it invariably accounts for a unacceptably large portion of a project’s time and budget. But it’s not just a time and money thing; poor software quality can adversely affect brand, customer perception, employee morale and the ability of your company to comply with regulatory legislation. Software quality is serious business. Companies that do not strive to make it a strategic business imperative, lose the fight for the competitive advantage and true return on investment to all areas of the business.

By improving quality processes, less resource is required and employees benefit from far improved efficiency. Weeks of effort and costs from a project deadline can be cut in half. At the same time a project’s chances of success increase and a healthy reputation for delivering applications that are increasingly stable with each release is realistic.

In this whitepaper, we shall look at the issues related to the improvement of software quality and show that not only can these challenges be surmounted, but that a significant, tangible financial benefit can be won in the process. It also looks at some key factors to consider when selecting software automation tools.

The CIO challenge

At the CIO level, a potent mixture of restricted budgets, aggressive timelines, and businesses seeking a competitive advantage through new technologies, increase the pressure and risk associated with managing high profile initiatives.

Executive boards are challenging CIOs to conceive and deliver strategic initiatives that will in turn provide competitive advantage and future business benefits. These will involve projects such as re-engineering existing systems, implementing new technologies, upgrading core vendor packages to current releases, delivering business driven enhancements, and enhancements mandated by legislation.

A cold hard look at these projects will reveal a common and sizeable component: testing. When all of the testing elements are added together (requirements, unit, system, QA, regression, UAT), they account for a minimum of 30% of total project time and can account for as much as 75% (based on our own studies). Which ever way you cut it, testing is a significant activity in application development.

The potential for victory

The amount of effort involved in software testing varies from one project to another. A key management challenge is to balance the risk of errors against the testing effort. Very often a greater degree of testing is desirable but impractical due to time constraints and the sheer effort involved in finding a reducing number of errors in subsequent test cycles.

It is obvious that if the testing process can be made more effective in determining errors and if it can also become less time and resource consuming, projects will be delivered on-time or early, at a reduced cost and implementation will be easier.

A key factor in a successful testing strategy is to maximize testing at the earliest opportunity. This is when the effort in detecting, documenting, fixing and re-testing an error is at its lowest. Very often, testing performed in the development/coding phase is not even considered to be 'real testing', yet it has the greatest opportunity to impact the remainder of the project. In the case of a modification in an existing system, the code change might be relatively small, say two or three hours work, yet the testing that will be involved could easily be measured in days if the change affects a core part of a system. By starting improvements here, it could be possible to eliminate 80% of this testing effort and reduce testing and required re-work because more errors have already been eliminated.

Quantifying the cost

Organisations that fail to take quality initiatives seriously risk significant loss in revenue as well as irreparable damage to the brand and reputation, customer satisfaction, loyalty and market share. But yet we still see too many companies not doing enough to ensure that their software applications are of the highest quality.

Bola Rotibi, Principal Analyst at Macehiter Ward-Dutton concurs. "What I find incredible after all this time, given the weight of evidence and eminent studies on the cost savings and the growing complexity and importance of software in our modern lives, is that the "sloppy" mentality and attitude still holds such sway in software delivery processes." Rotibi continues, "Many organizations don't spend nearly enough effort on improving the quality of the software they produce. More often than not they pay lip service to the concept whilst secretly holding the belief that it is a waste of resources (time, staff and money).*

"Many organizations don't spend nearly enough effort on improving the quality of the software"

*The Dilemma of "Good Enough" in Software Quality
Bola Rotibi

The negative impact on business revenue and profitability as a result of poor software quality are numerous:

> **“ If you continue to release poor quality applications you will build an innate expectation that future releases will be of the same standard ”**

Increased labour costs low morale

Releasing poor quality software increases the risk of receiving more complaints from customers. Not only does this put extra strain on customer service and support functions, it also means development and QA are diverted away from new product development and have to retrospectively fix the errors that the customers are complaining about. Morale can also be adversely affected if there is a large uplift in customer complaints.

Adverse impact on brand and reputation

Your brand and it’s reputation is your most valuable asset. Positivity on blogs, forums, and other networking arenas can be a great source of free publicity and marketing. Potential customers will be more likely to trust a recommendation from a peer or friend more than any promotional or advertising campaign. Conversely, negative comments can spread like wildfire! Once a reputation is tarnished in this way, it is a time consuming and very expensive task to win back trust.

Exposure to regulatory non-compliance

Depending on how you plan to use the software, poor quality can leave your company susceptible to severe penalties if corporate regulatory controls are not adhered to. Companies that handle personal or potentially sensitive data are required to exercise extreme caution with regards to how this data is stored and handled. Research suggests that over 40% of IT departments are using live data for testing and training purposes. If such data is personal or of a sensitive nature then this could represent a potentially serious exposure - especially if the data is outsourced to a third party for testing or training services. If a company compromises private customer information or fails to record the correct regulatory information, an organisation can face serious legal penalties, including fines and imprisonment.

Loss of revenue

If you continue to release poor quality applications you will build an innate expectation that future releases will be of the same standard. This can build up a level of distrust in your customer base. If a release is extremely late because of issues, then customers/clients will assume there are problems and may be reluctant to buy or use the application until the product has proven itself. The worst case scenario is they don't buy at all.

Lost opportunities

Successful automation is fundamentally about two things: Delivering software to the business quicker and in the best quality possible. In essence, the business gets exactly what it needs, sooner. What is the loss to the business of a two week delay in a key IT implementation? Or to put it another way, what advantage can the business gain over its competition by delivering the application two weeks earlier?

“Effectively deployed software automation solutions can have a significantly positive affect on functional and regression testing timelines”

Why automate your testing?

So, we are in agreement. Improving software quality is good. Good for morale, good for management, good for customers, and ultimately good for the bottom line. Business leaders should recognize this and strive to instill a passion for high quality software across the entire organization. They should provide the correct environment, technology, training, guidance and processes to enable the kind of performance the organization, and indeed its customers, are demanding.

In short, to deliver improved software quality there needs to be commitment, a plan, and a process.

Ironically, this is where software comes into its own. More specifically, software designed to test the quality of other software, what you and I would call software test automation.

So, why should one consider test automation? What are the benefits? Here are ten to get you started.

- Relieve the testing bottleneck and achieve faster application time-to-market
- Reduce the money spent on testing
- Increase test coverage & reduce risk
- Configure and repeat your tests
- Re-assign your resources
- Deliver highly accurate tests and find defects earlier
- Ensure corporate compliance
- Ensure the scalability of your applications
- Test data creation, management, and security
- Ensure that every test you do is consistent and the most thorough it can be

Effectively deployed software automation solutions can have a significantly positive affect on functional and regression testing timelines. Let’s look at these benefits in a bit more detail.

Relieve the testing bottleneck

It is often suggested that testing takes up to 40% of the entire application development timeframe. It stands to reason then that even a small percentage of improvement in efficiency could have a large positive impact in getting an application launched. Once ready, automated tests take much less time to complete than manual tests, and can often run unattended. Just press the ‘go’ button, and analyze the results when it is complete. In this situation tests can be scheduled to run 24 hours a day, over a weekend or during holidays: More testing completed with less resource.

Reduce the money spent on testing

Time is money, so it stands to reason that the less time and effort spent on testing lowers the actual cost. Look at this a different way: every time a specific task, that could be automated, is carried out manually, time is wasted. As a direct result, businesses lose money. There are tests that can be automated that can fast track the whole process of developing the application. In essence, time and money are saved.

“By implementing automated tests, a larger amount of tests can be executed on any given application, achieving a higher level of coverage than could be possible via manual testing”

Increase test coverage on each testing cycle

By implementing automated tests, a larger amount of tests can be executed on any given application, achieving a higher level of coverage than could be possible via manual testing. With a larger amount of testing being carried out, more features can be tested (giving greater breadth of test coverage) as well as more stringent testing of features (giving a better depth of testing). All of this results in a higher quality application and happier customers!

Automated tests are repeatable

This is another advantage of software test automation. Here, developers do have an opportunity to see how certain programs react when the same processes are being done repeatedly. These tests are also configurable. As a result, developers can develop complicated tests that could reveal data hidden from the application itself. Additionally automated tests can be reusable. This means that they can be utilized in different approaches unique to certain applications.

Re-assign resources

By automating some tests, it is possible to free up significant amounts of the testers time, enabling them to move on to the next project sooner or spend more time testing at a greater depth. However, it must be stated that test automation will never fully replace the need for manual testing. No matter how sophisticated automation becomes it will never be as good at detecting quality issues as a human, particularly on less obvious errors, or by using initiative to uncover errors. However, by freeing up manual testers from having to execute easy, repetitive testing tasks, automation enables them to focus on using their creativity, knowledge, and instincts to discover more important, hard to find errors, giving them greater job satisfaction.

Ensure corporate compliance

Internal and external audit pressures, such as Sarbanes Oxley, require that the depth and effectiveness of the testing can be rapidly and intuitively appreciated. This is another area where automation can formalise and streamline the process for efficiency gains. By recording the entire testing process and producing reports formatted to audit standards automation can provide the management data and audit trails needed to satisfy compliance requirements.

Ensure the scalability of applications

For some applications such as websites it is necessary to test the amount of usage or traffic they can handle before they ‘break’. Replicating the behavior of such applications under extreme stress or load can help avoid machine overloads, unnecessary infrastructure investments and the implementation of enhancements that fail the scalability tests. It is very unlikely that any corporation will be able to simulate 100,000 users on its application at the same time, so software is needed to put an application through its paces. An effective load test will provide assurances that when your code goes live, there will be no surprises caused by a full production load.

Test data management, security, and creation

Data is king. The effectiveness of any testing will be largely dependent on the quality of the test data used. Manually creating quality test data takes time and as a result testing is often performed on copies of live databases. This is not an ideal scenario as it leads to elongated test times, exposure of confidential data and increased data storage requirements. In addition, once a subset of live data is created, data de-identification needs to take place to comply with confidentiality and data protection requirements. Some automation solutions can help with creating, manipulating and protecting your test database, allowing data to be to re-used time and again. The time and cost savings in this area are potentially huge.

The Dangers of Choosing the Wrong Solution

Having said all this, automation may not be the answer to all problems. If the wrong solutions is purchased it could actually create more work than had been done initially!

“ If the wrong solutions is purchased it could actually create more work than had been done initially! ”

First generation testing tools, most of which are from household names, along with tools developed in house are useful to an extent in automating some elements of the testing process. However there are widely recognised flaws in their overall approach and capabilities that limit their usefulness. Let’s quickly look at these issues.

These tools can be complicated

There is no doubt that the mapping of the route of a test must take through an application in order to thoroughly test all areas is a complex process. The traditional approach taken to address this challenge has been to require the testers to learn a scripting language. The flaws in this approach are rather obvious:

- The people involved in testing (especially with UAT) often come from a business rather than technical background. For them, the need to learn a new scripting language is a significant barrier to the successful adoption of test automation.

- The number of people who can use such tools becomes self-limiting. It will tend to attract those with technical strengths rather than business knowledge. These ‘experts’ will be rare and expensive and their departure from a company could result in a temporary halt in automation until someone else can either be hired or taught.

These limitations are costly. Significant investment will have been made in the technology, even more in trying to train staff to proficiency, and now all that is left is a product that has become shelf -ware and a return to manual testing.

These tools also have a tendency to only test what is on the screen. The visual layer. But what use is it to test what can be seen on a PC, when the real processing has occurred on a remote server? If one considers the entry of an internal order, does the fact that the next screen says ‘Thank you for your order, please print this screen as your confirmation’ truly indicate that the details have been successfully stored and the associated processing initiated and tracked through to completion? Effective automated testing must not be limited to driving and validating the visual layer.

“Gone are the many months of script building and preparation before a single item can been tested. With next generation solutions, testing can start in days, not months”

So, a big brand test tool has been purchased, expensive scripters have been hired and placed on the latest and greatest training course that will enable them to utilise the tool to the maximum. They spend weeks building a library of scripts and test cases and everything is going well. Then a new version of the application being tested is launched. Overnight the bulk of the investment has become worthless. Existing scripts and test cases simply make no sense to the new version of the application, the test team has a choice: either amend hundreds of scripts to get them working on the new version or continue the testing manually. Either way it is not a clever use of time and resources.

Next Generation Solutions

Next generation solutions by Original Software, negate the issues this paper has addressed and empowers any business user to become a software quality custodian. The Original Software solution embraces the full spectrum of Application Quality Management, (AQM), across a wide range of applications and environments.

No scripting language

The technology is built into the solution, not the script, so a more diverse audience within the business can use them. Test automation is no longer confined to programmers and scripters. Graphical scripts are built based on the user’s interaction with the system under test. In essence the scripts are built via pointing and clicking a mouse on screen.

Speedy implementation

Gone are the many months of script building and preparation before a single item can been tested. With next generation solutions, testing can start in days, not months. This means test assets can be utilised earlier resulting in improved return on investment.

Scripts that update when applications change

The intelligent technology behind these solutions mean that they automatically recognise when an application interface has changed and the scripts are updated instantly to reflect the changes. This means that the solutions are completely re-usable, no matter how often the application changes. As a result, the automation zone is no longer confined to those areas of the application that are stable and risk free.

Broader testing scope - database testing

Data underpins the entire testing process. Poor data will result in poor testing. With a growing emphasis being placed on data quality, it is vital that the integrity of the test data is protected at all stages of the testing project. The next generation of solutions can ensure test data is in A1 condition all the way through the testing process.

Conclusion

We have seen that inefficient testing and reworking, together with the multiple costs associated with downtime due to poor software quality, is costing many companies hundreds of thousands of dollars per year.

Yet, we have also seen that effective automation of those testing processes can not only drastically reduce the inefficiencies and downtime, but also improve the quality and the speed of application development programs. We have also seen that all of the shortcomings of traditional automation tools can be overcome by purchasing next generation test automation solutions with more functionality and a design ethos cantered around the user.

> **“We have also seen that all of the shortcomings of traditional automation tools can be overcome by purchasing next generation solutions with more functionality and a design ethos centred around the user”**

Next generation solutions do not depend on old scripting languages. A good solution should be able to help meet the demands of the business and ensure:

- The testing bottleneck is no longer a business burden.
- Test coverage is increased and corporate risk reduced.
- Defects are found earlier in the application development.
- Corporate compliance is met.

All this is realistic and can be achieved through new technology that eliminates the need to know a scripting language.

Since the first automation tools appeared on the horizon over 20 years ago, software test automation has now finally been hauled into the 21st century and software testing does not need to be the burden it once was.

So what is the next step?

Original Software's innovative approach to solving real application quality issues has resulted in a solution suite that provides a dynamic approach to quality management and automation, empowering all stakeholders in the quality process, as well as uniquely addressing all layers of an application stack. Original’s market leading solutions are taking on and beating the previous incumbents in the market, proving that they can deliver rapid value at a speed that makes a real difference and helping over 500 customers to provide quality applications to the business, faster and at a lower cost.

If you want to make software quality a strategic business imperative and would like to ask us a few questions about our unique approach, simply get in touch today from our website at: www.origsoft.com/contact

About Original Software

With a world class record of innovation, Original Software offers a solution focused completely on the goal of effective quality management. By embracing the full spectrum of Application Quality Management across a wide range of applications and environments, the company partners with customers and helps to make quality a business imperative. The solution encompasses a quality management platform, manual testing, full test automation and test data management, all delivered with the control of business risk, cost, time and resources in mind.

More than 400 organizations operating in over 30 countries use Original Software solutions. Current users range from major multi-nationals to small software development shops, encompassing a wide range of industries, sectors and sizes. We are proud of our partnerships with the likes of Coca-Cola, HSBC, Unilever, FedEx, Pfizer, DHL and many others.

Original Software

European Headquarters
Basingstoke, UK
solutions.uk@origsoft.com
www.origsoft.com

North American Headquarters
Chicago, USA
solutions.na@origsoft.com
www.origsoft.com