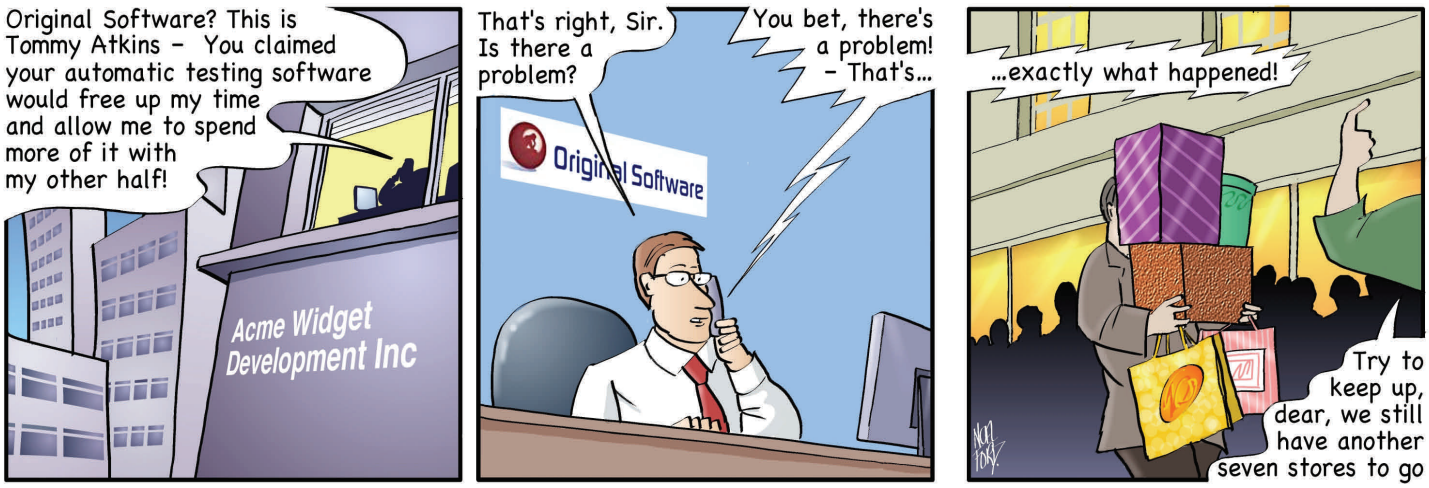
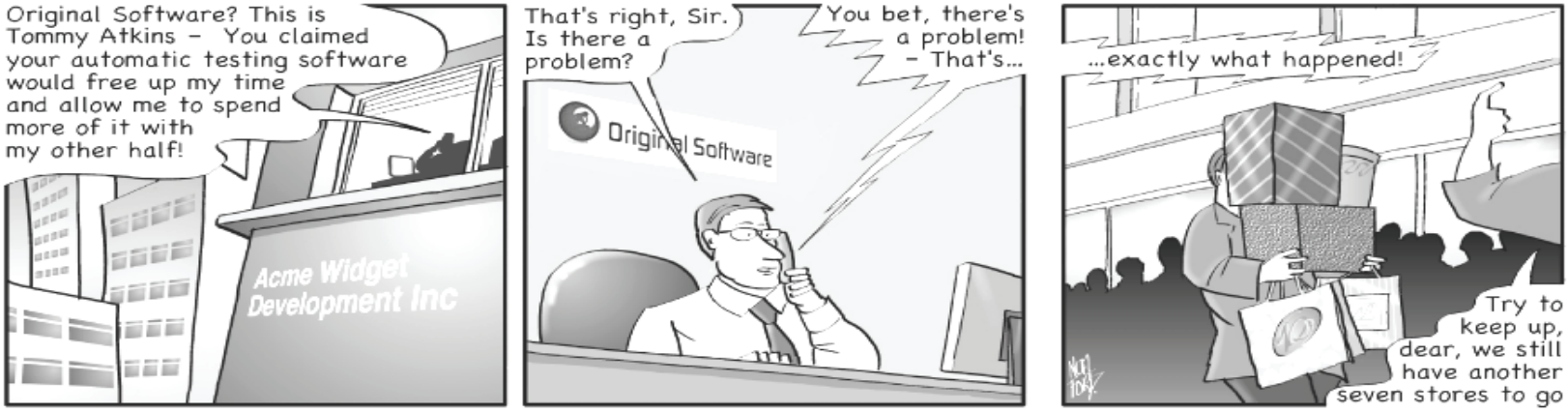


The Business Benefits of Software Test Automation.

Or how to save money, improve efficiencies and increase your competitive standing with one easy investment.

Original Software

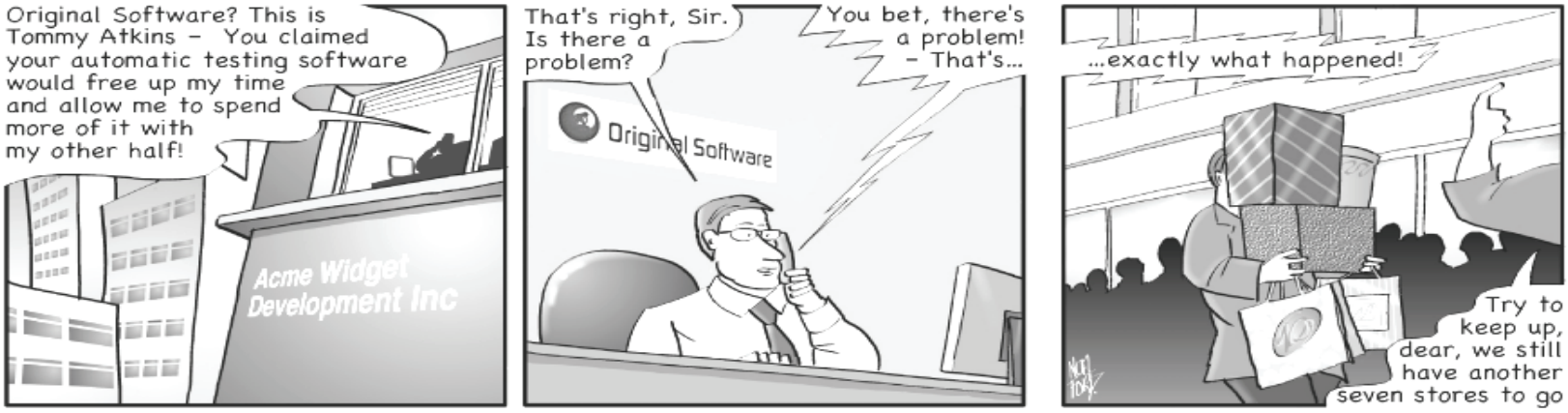


Synopsis

Poor software quality is costing industry billions of dollars in lost revenue and higher costs. It also has a negative effect on corporate reputation and branding and can even lead to court cases and legislation. In short, software quality is vital to the welfare of your organization. Yet, still software quality takes a back seat with minimal investment in technology, process, training or education to ensure that everything possible is being done to ensure applications are of the highest quality possible.

Software test automation carries the potential to revolutionize your testing process. By automating many testing tasks, organizations could benefit in numerous ways: financially, an improved reputation, highly motivated staff and happy customers.

This whitepaper, aimed primarily at IS and QA Management, attempts to outline the main benefits of test automation as well as looking at the costs of poor quality software. Finally, assuming you are bought into the concept of automating your quality process, there is a list of questions you may want to ask any prospective automation vendor to ensure you get the most out of your investment.



Defining Quality.

Before we talk about software test automation, lets take a step back and thing about quality. What exactly is 'quality'? More importantly, what exactly do we mean by software quality?

The Oxford English dictionary offers up the following definition: *'degree of excellence, relative nature or kind or character'.* That is all very well, but it doesn't really help software developers understand the nature of *software* quality. We need a better understanding of what software quality is, so when quality is discussed during the software development process, all team members are in agreement and are talking about the exact same thing. A solid definition of software quality that all team members understand and accept promotes thoroughness and attention to detail.

James Juran (1988) is succinct in his definition: *'fitness for use'* gives us a good starting point, and Juran continues to suggest that a software product is not of high quality until it adds value to all stakeholders.

The key point here is adding value to *all stakeholders*. It is not just the customer who benefits from Improved software quality. Your organization will be a winner too. The business benefits of ensuring the highest level of quality throughout all phases of your application development lifecycle are both deep and wide. Innovation is encouraged and facilitated through higher quality applications and costs are lowered by reducing risk and eliminating rework. Embarking on a continuous program of quality control will always cost less than ignoring the problems and retrospectively fixing them .

Low quality output is tolerated in software development more often than any other discipline or profession. Often software releases are shipped to customers or sent live on the web with numerous known faults, but because 'it is still better than the last version' or its 'better than nothing' it goes live without management asking too many tough questions.

Do you think the same would happen at Boeing, making Jumbo Jets? Picture the scene:

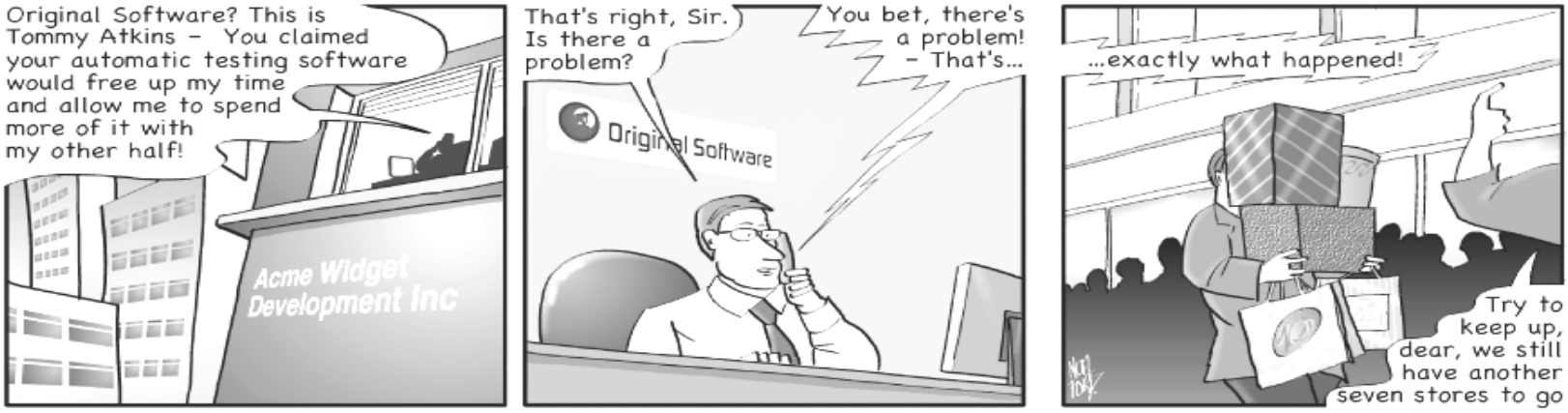
Engineer 1 'That plane has one of its wings flapping in the breeze, why is that?
Engineer 2 'Yeah, didn't have time to check it, but the other wing is ok I think, and a plane with one wing is surely better than a plane with no wings, so I didn't think it was a problem'

Not sure that excuse would 'fly'. So why is there not a similar zero tolerance to quality in software development, than there is in (for example) aircraft manufacturers?

Well, no one will die if someone releases poor quality software right? Wrong. In the summer of 2008 a medical firm were forced to withdraw a product after a software flaw in a drug pump led to patients being given potentially lethal overdoses. *The importance of software quality is unquestionable.*

The thing is, quality, like beauty, is in the eye of the beholder. From a software point of view it is very dependant on what the software is written to do, the tests used to check, and the observer doing the checking. A piece of software that is designed to monitor the mating habits of long eared bats will have slightly less stringent quality principles attached to it than say, the software that keeps fighter jets in the sky. This may be a bit pedantic, but you hopefully get the point; quality is a self determined thing and organizations must define it for themselves.

It's about time organizations realized that the issue of software quality needs attention. Now. On the next page we look at some of the ways poor software quality can cost businesses.



The Cost of Poor Software Quality.

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

Today's business environment means risks sometimes have to be taken to obtain competitive advantage. However unmanaged risks in IS developments can have disastrous consequences. Application failures lead to a loss of external and internal confidence, cause damage to the brand and ultimately can have a negative effect on the revenue stream.

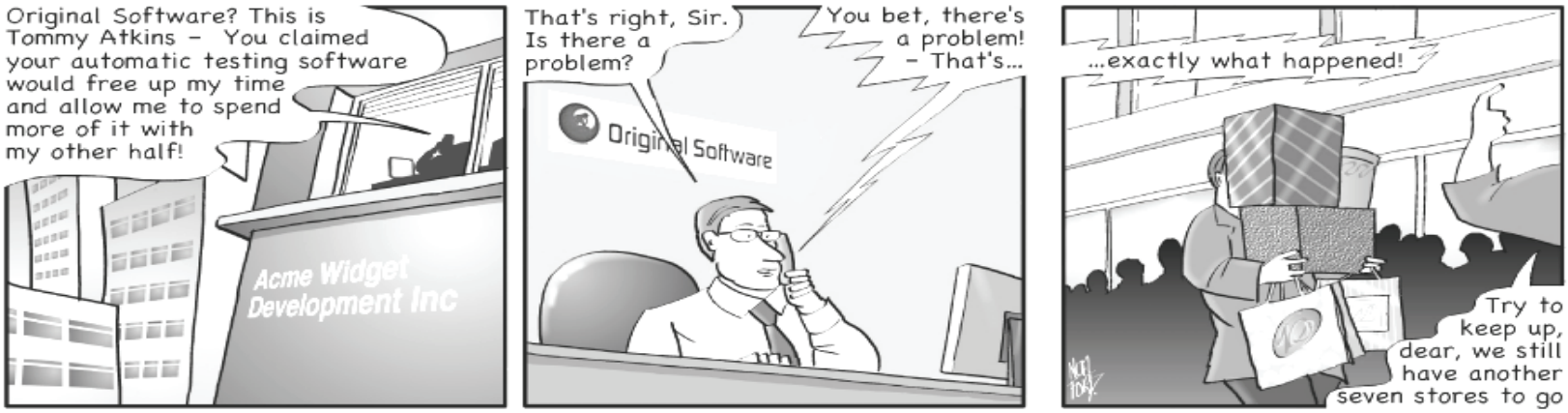
The negative impact on a businesses revenue and profitability from poor software quality are numerous:

Increased labor 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. Studies consistently show that the cost associated with fixing errors in software increase significantly as the development process continues. It is more efficient and less costly to prevent errors getting into the software in the first place, rather than trying to find them after the product has gone live. Morale can also be affected if there is a large uplift in customer complaints.

Adverse impact on brand and reputation
Your brand and its reputation is your most valuable asset. Positivity on blogs, forums, and other networking arenas, can be a great source of free publicity and marketing. Many potential customers will trust a recommendation for a peer or friend far more than any promotional or advertising campaign. Conversely negative comments can spread like wildfire in today's environments, and once a reputation is tarnished in this way, it is a time consuming and very expensive job to win back trust. Just ask the owners of Terminal 5 at Heathrow.

Dr. Vince Nowinski, principal methodologist at Satmetrix explains the word of mouth phenomenon; "When customer experiences are positive — and loyalty is high — we expect customers to spend more on average and to generate new business via positive word-of-mouth. Conversely, when customer experience is poor and loyalty is low — we expect lower purchasing value (perhaps even defection), as well as the potential loss of new business through negative word-of-mouth."**

*The Dilemma of "Good Enough" in Software Quality—Bola Rotibi
**Satmetrix Study Defines Economic Impact of Word of Mouth (May 2008) http://www.customerthink.com/news/



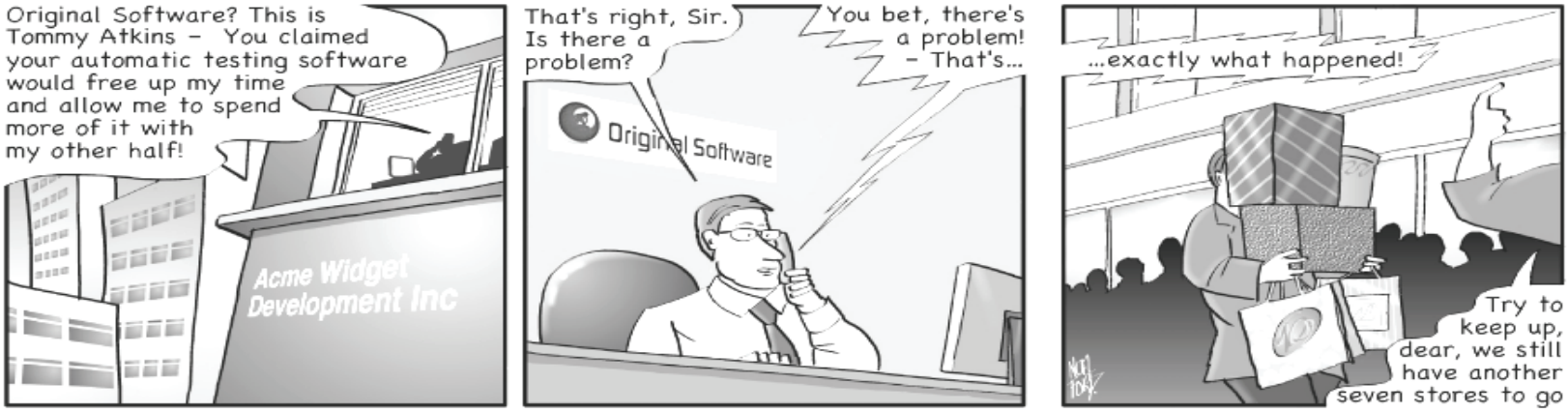
Exposure to regulatory non-compliance
Depending on how you plan to use the software, poor quality can leave your company susceptible to severe penalties if corporate regulatory controls are not adhered to. Any company that handles personal or potentially sensitive data is required to exercise extreme caution in the ways that such 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 of a personal or sensitive nature then this represents a potentially serious exposure - especially if the data is outsourced to a third party for testing or training services. Colin Armitage, CEO of Original Software says: "Regulatory controls are continuing to have an ever greater impact upon the ways that companies do business. If a company compromizes private customer Information or fails to record the correct regulatory information, an organization can face serious legal penalties, including fines and imprisonment. "

If that isn't reason enough to ensure correct software quality, I don't know what is.

Customers could delay purchases, and you will lose revenue because of poor quality
If you continue to release poor quality applications you will build an innate expectation that future releases will be of the same standard. This will build up a level of distrust in your customer base. If a release is very late because of issues, then people will assume a problem and maybe reluctant to buy or use the application, at least until the new 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 better shape. 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 put it another way, what advantage can the business gain over its competition by delivering the application two weeks earlier?



Why Automate your Software Testing?

The Business Benefits of Software Test Automation

So we are all 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 must recognize this and instill a passion for high quality software across the entire organization. They need to 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 you consider test automation? What are the benefits? Here are nine to get you going.

- 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

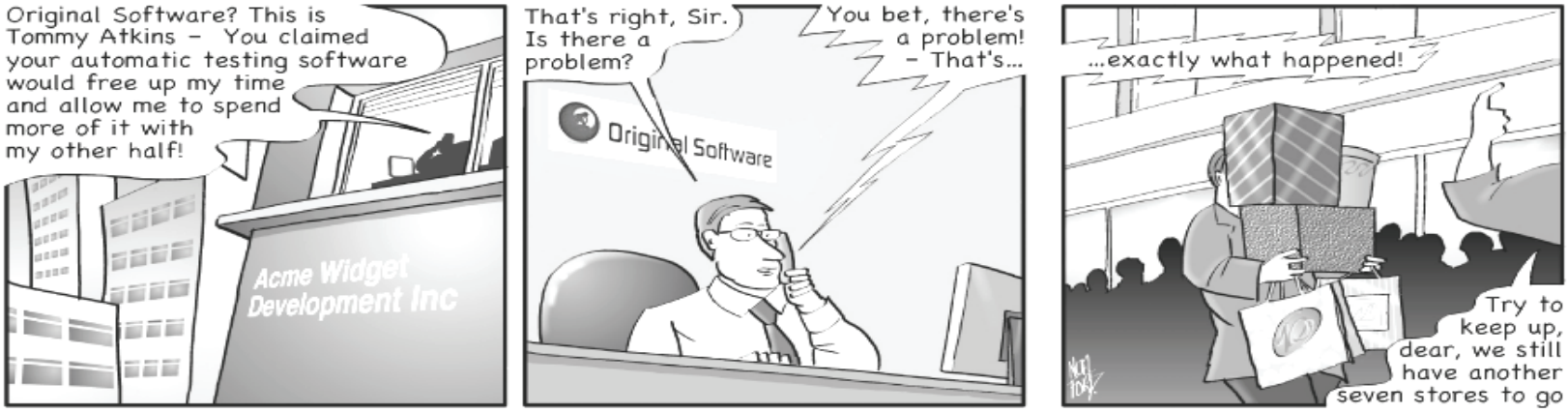
Effectively deployed software automation solutions can have a significantly positive affect on functional and regression testing timescales. Lets look at these benefits in a bit more detail:

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 % improvement in efficiency in this area could have a large positive impact on the time needed to get an application launched. Once set up, automated tests take much less time to complete than manual tests, and can often be run unattended. You just need someone to press the 'go' button, and analyze the results when it is complete. In this situation tests can be run 24 hours a day, over the weekend or during holidays: More testing completed in less time.

Reduce the money spent on testing

Time is money, so it stands to reason that the less time and effort that is spent on testing over time the lower the actual cost. Look at this a different way, every time a specific task, which is best done via automation, is done manually, time is wasted. As a direct result, your business will lose money.



Increase test coverage on each testing cycle
By implementing automated tests, a larger amount of tests can be executed against a given application, achieving a higher level of coverage that could be possible via manual testing, unless you had a massive team of manual testers and no time constraints. 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 happy customers!

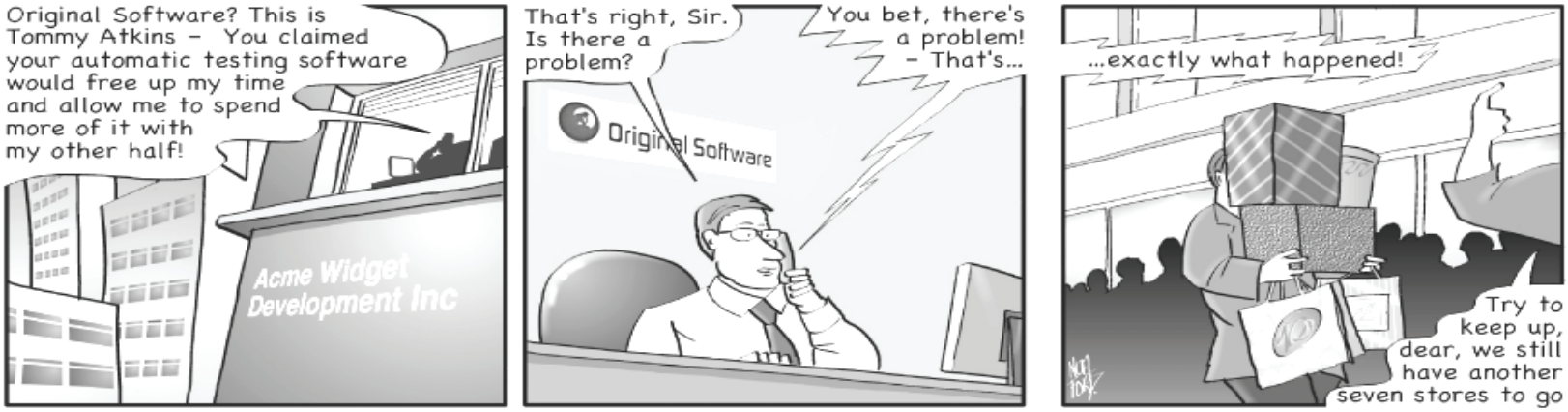
Automated tests are configurable and repeatable.
This is another advantage of software test automation. Here, developers do have an opportunity to see how a certain program reacts when the same processes are being done repeatedly. These tests are also configurable. As a result, developers can develop some complicated tests that can possibly reveal data hidden from the application itself. Another thing about automated tests is that they are reusable. This means that they can be utilized in different approached that a certain application may have.

Re-assign some of your resources
By automating your testing you will be able to free up significant amount of your testers time, this enables them to either move on to the next project sooner or spend more time testing at a greater depth, which will ensure a greater robustness of your application. 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.

Deliver highly accurate tests and find defects earlier
Test automation gives your team an easier way of replicating and documenting software defects. This can help increase the speed of development processes whilst ensuring correct functionality across all areas. A defect that is identified early in the development cycle can be significantly cheaper (up to 100x) to rectify than those that are not found until much later in the production of the software.

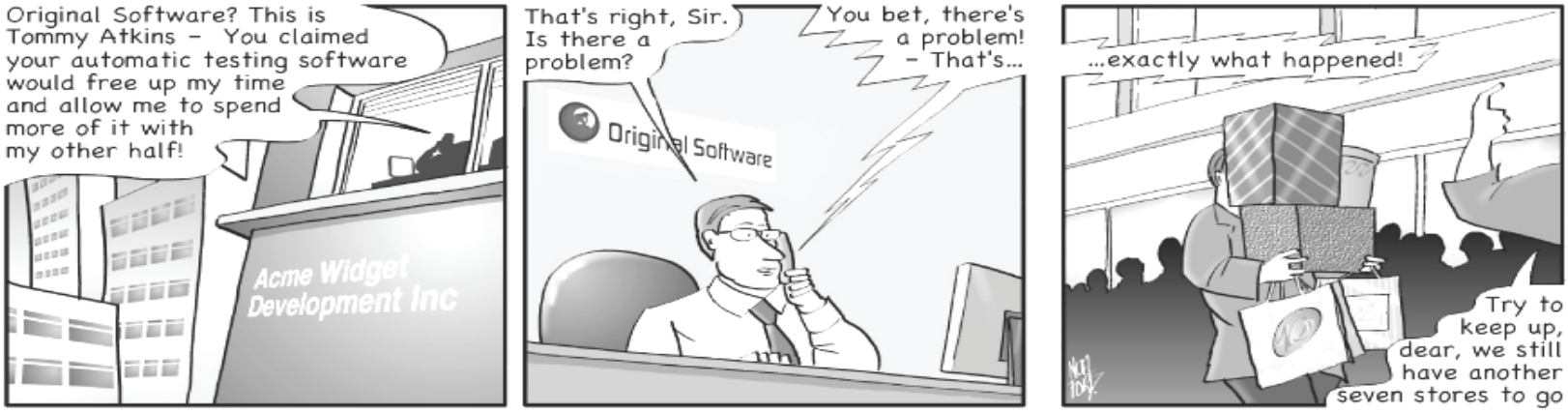
Ensure corporate compliance
Internal and external audit pressures such as Sarbanes Oxley all require that the depth and effectiveness of the testing can be rapidly and intuitively appreciated. This is another area where automation can formalize and streamline the process for efficiency gains. By recording 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 your applications
For some applications such as websites it is necessary to test how much 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. Obviously it is very unlikely any corporation is going to 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 is put live, there will be no surprises caused by a full production load.



Test data management, security, and creation.
Data is king. The effectiveness of your testing will be largely dependant on the quality of the test data you use. 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 you to re-use your data time and again. The time and cost savings in this area are potentially huge, as well as ensuring you stay out of prison by not mis-managing data.

Ensure consistent and thorough testing.
With manual testing, every tester has a different way of doing things, a different style, methodology and approach. This could mean that one tester is better served running one specific test and not another. With automation, you have the opportunity to let the 'expert' in a particular test do the first run, build the automation, save the test and make it available for others to re-use. By doing this you are ensuring each test is run consistently, and by having the expert run and set up the test in the first place, you are ensuring that your tests are the best possible quality. Time and time again, regardless of who runs them subsequently.



Why Isn't Automation More Widely Used?

The benefits of automation are numerous, we know that, but how come it is not more widely used? Many of the reasons are unique to the individual organization but here are our thoughts on the most common reasons automation is not implemented.

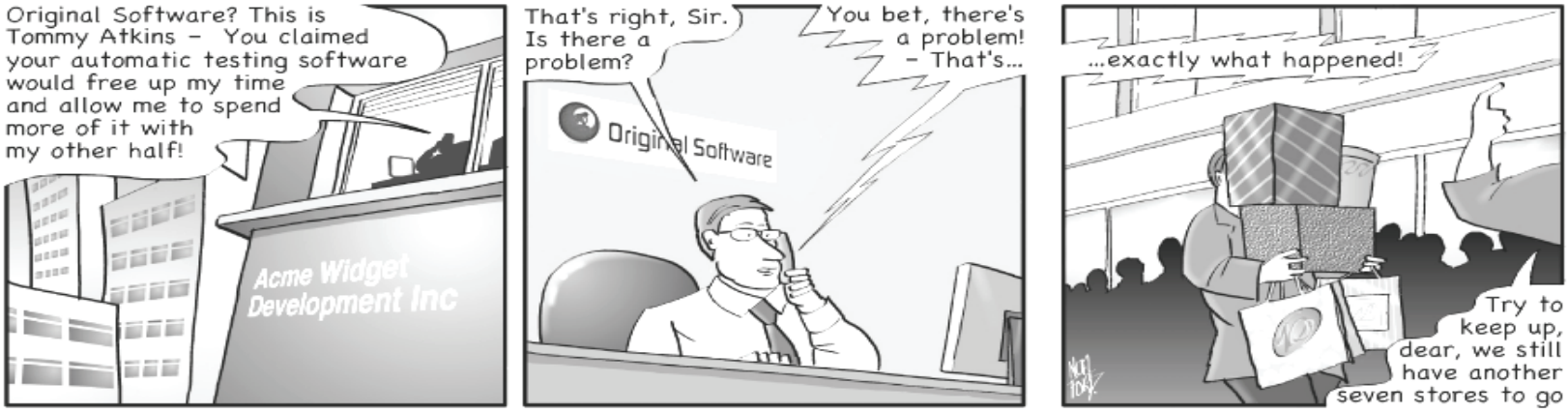
Reason 1: Cost.
There is no escaping it, software test automation can be a large investment, but the benefits can be significant in terms of resource, productivity, quality and accuracy. These are the areas where the business case needs to concentrate. In many organizations the cost of one customer-facing defect can justify the spend. Thinking of it in this way and the cost isn't as high as first imagined. Also, some vendors will allow you to phase your investment, allowing you to implement automation at a speed that suits your organizational structure, your requirements, and your budget. Gone are the days when you have to spend now and wait months and months for any payback. For more information see Question 8 of 'Eight Questions to Ask Your Automation Vendor' on page 12 of this whitepaper.

Reason 2: F.U.D.
Fear. Uncertainty. Doubt. With big investments there is always an element of fear, uncertainty and doubt. No one wants to be the scapegoat if a large investment doesn't perform as well as had been hoped. And let's be honest, over the past 20 years the software testing industry has not covered itself in glory has it? With many of the leading software testing companies promising the earth, but simply not delivering, its no wonder there is a little uncertainty when it comes to investing in software test automation. It is therefore vital that you interrogate your vendor shortlist before you choose a supplier, ensure they can meet your objectives. The next section in this whitepaper should help you by providing you with some questions to ask your chosen vendors.

Reason 3: I am too busy to think about automation.
This is all too common. QA and test teams are working flat out to keep up with their manual testing deadlines that they have not time to think about automation. But, you don't have to stop everything to implement automation. With a phased implementation it can be achieved whilst working around your current activities. The benefits of automation are obvious, it just needs a clear head to think about how improving your teams' productivity, accuracy and speed could better serve your organizations' corporate objectives.

Reason 4: Resistance to change.
It is a fact of human nature, people in general do not like change. If you have a team of manual testers, happily going about their business they may not be too happy at the prospect of a change to their role. Some of them may think this is the first step on they way to them losing their jobs—as a manager it is up to you to communicate properly the advantages of automation and dispel any fears of them losing their jobs.

Reason 5. The tipping point.
Software test automation is often sought only when there has been a business loss due to poor quality. Then there is a reaction of 'how can we ensure this doesn't happen again?' People don't really think of test automation until they have been burnt, as such there is very little prevention taking place. But as we have already discussed, poor software quality is costing industry billions of dollars, so maybe it is better to start improving quality now, rather than waiting for the tipping point, waiting for something to break.



Eight Questions to Ask Your Automation Vendor

There are a myriad of software vendors out there, all claiming to be the perfect solution, the panacea, to your problems. But be careful. All may not be as cozy as it would first seem. To help you in your choice of vendor, here are some questions you may want to ask your prospective supplier.

Q1. How long will it take to get up and running and how long will it take to see a return on my investment?

Good Answer: Depending on what solution you implement you should be up and running anywhere between 1 hour and a couple of weeks of installation. As for a return on your investment you should see this within a year. Indeed 6-9 months is typical, and you should start to see benefits from the first week of using the solution.

Likeliness of hearing that good answer: Very unlikely, with script based tools there is a long period of set up time where the library of scripts need to be built. This can take a team of scripter's several months, meanwhile the testing still needs to be carried out manually. Once the library of scripts are ready, you can begin to automate. But because of the limitations of scripts, and the amount of ongoing maintenance these tools continue to demand, ROI is not attainable for years. Indeed, HP Mercury admit in one of their application notes 'Studies have show that the efforts to properly record, enhance and maintain automated tests typically require up to six test cycles to provide a positive return on investment (ROI).*

With Original Software, you will be up and running very quickly, will enjoy fast return on investment , typically only 3 or 4 test cycles are needed for positive ROI. How can Original Software produce such easy-to-use and effective solutions? Because our tools do not rely on any of the manual script production. Scripts are built automatically based on the users interaction with a point and click interface.

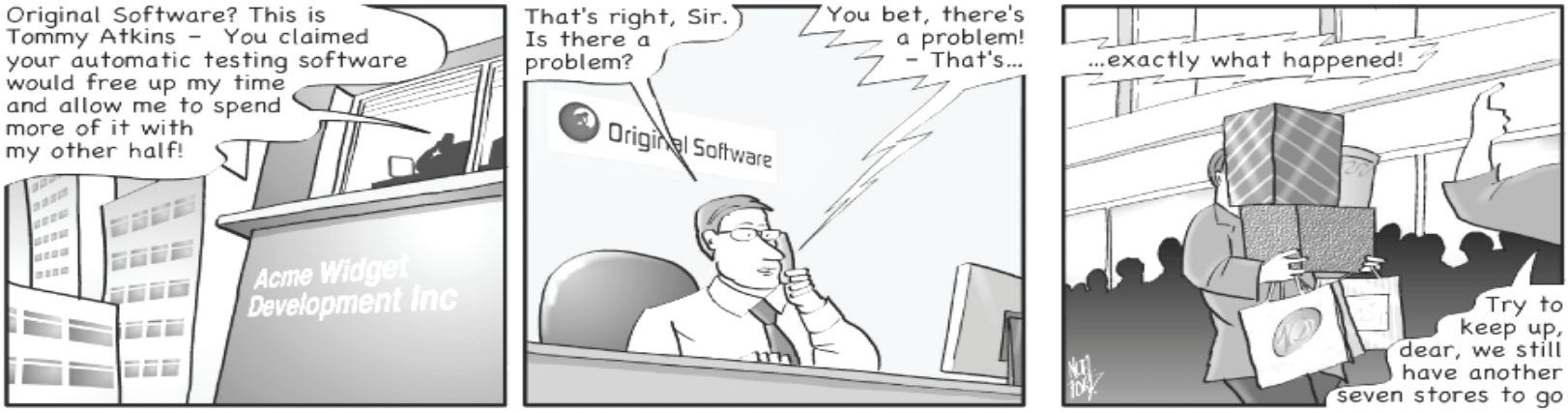
Q2. Do I have to be a programmer to use the tool or can anyone use it?

Good Answer: No, anyone can use the tool, including line of business members that may be involved in UAT.

Likeliness of hearing that good answer: For all script based tools, you will not hear this answer. Because of the complexity of the scripts, specialist programmers are needed to use the tools. Not only is this an expensive labor pool, it is very limiting in the number of people that can get involved in automation, and as a consequence will severely limit your automation coverage and effectiveness.

With our script free technology, anyone can use an Original Software automation solution. As a result, UAT and other areas of the testing process can benefit from automation.

*Controlling risk:quality management for SAP implementations and upgrades. HP Software, June 2008



Q3. Will I be able to re-use my tests without re-coding, even if the application under test has changed?

Good Answer: Yes

Likeliness of hearing that good answer: Very unlikely. With all traditional scripting-based test automation tools, every time your application changes, you will have to re-code all the relevant scripts. So much for automation!

What you need is Original Software's self healing script technology that automatically detects changes to the application, presents you with the changes to the scripts, and gives you the opportunity to review and update the old scripts with one click.

Easy.

Q4. Realistically, how much of my testing will I be able to automate?

Good Answer: There will always be some areas of testing that needs to remained manual, but you should aim to automate at least half of all your testing.

Likeliness of hearing that good answer: Very unlikely. Most of the more well known automation tools will automate 20-25% of your testing. And that 20-25% will be the more stable areas of your application that do not change very often (and therefore do not pose a great risk to your organization), because the time and effort it takes to manually change the scripts every time the application changes can be un-manageable.

HP admit this when talking about SAP testing, but the situation is the same with any application that changes frequently.

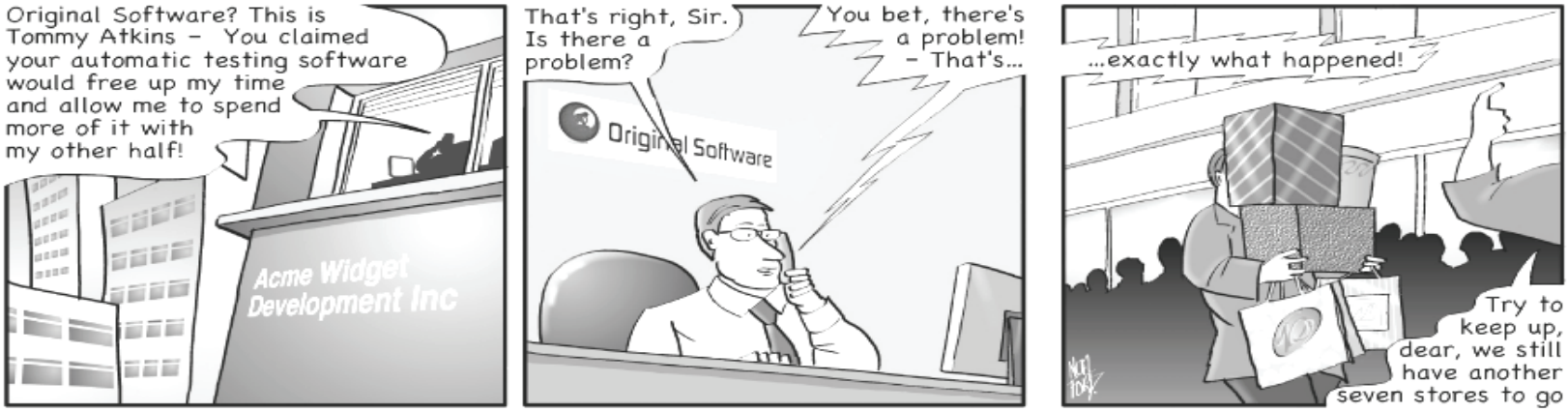
"HP QuickTest Professional… offer(s) relatively easy record-replay mechanisms... companies often find that significant effort still is involved in creating and maintaining their automated SAP test scripts....with the magnitude of changes common in SAP environments, automation engineers often find it difficult—if not impossible—to keep up with test maintenance. As a result, defects are found too late in the SAP implementation lifecycle, increasing the projects risks and cost.

*Studies have show that the effort to properly record, enhance and maintain automated tests typically requires up to six test cycles to provide a positive return on investment (ROI). Given the fact that a typical SAP implementation only goes through three test cycles, it is difficuly to justify investing in traditional record-replay test automation.**

Is that an admission of failure?

With Original Software's unique self healing technology you will be able to automate much more of your application because the tedium of manually altering scripts is eliminated from the process. Not only will you automate earlier in the test cycle, you will also automate more of the risky areas of the application, those areas that are subject to change.

*Controlling risk:quality management for SAP implementations and upgrades. HP Software, June 2008



Q5. How do you manage and maintain security of the test data?

Good Answer: We have a solution that provides the capabilities to efficiently create, manipulate and most importantly protect your test database, delivering the ability to return to the test database at pre-defined checkpoints, synchronizing your scenarios and data to achieve reuse.

Likeliness of hearing that good answer: Unless you are talking to Original Software, the likelihood is very very remote.

Q6. Is there anything to help me with the manual testing that cannot be automated?

Good Answer: Yes, we have a tool available that has been specifically designed to help with the manual testing that automation cannot work on. It is easy to use and set up, works in the background to record all of your testing and you can be working more efficiently within a couple of hours.

Likeliness of hearing that good answer: There are a number of manual testing solutions on the market, but many are just glorified notepad applications. Ask if these tools can convert these manual tests to automated scripts easily, also you may want to ask if there is any spell check or link check capability, also markup capability to add notes, screen and input capture and the storage of all results.

Q7. How early in the development cycle can I start to automate my tests?

Good Answer: As soon as you have a screen, even if it is a mock up, you can start automating your testing.

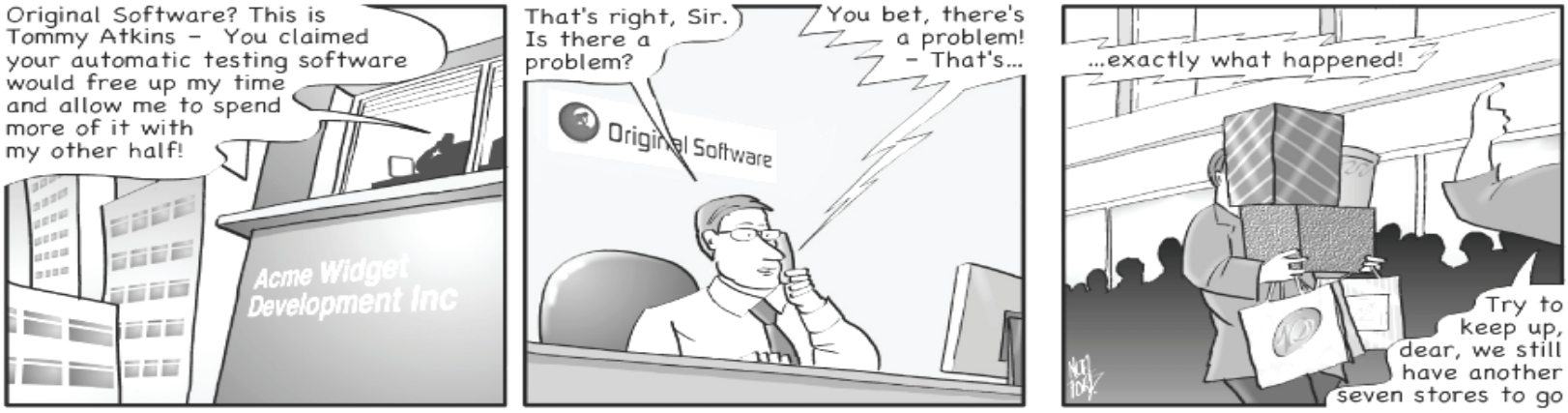
Likeliness of hearing that good answer: Unlikely. The problem here is, the earlier your start to automate the more changes you will need to make to your automated test scripts. So with those script heavy tools this may not be worth while attempting. With Original Software and its self healing capability, this problem is negated as script updates are carried out with a click of the mouse. Meaning you can start your automation earlier in the development cycle. Catching defects early saves time, money and effort. It is well known that the later in the development cycle you leave it, the harder and more expensive it becomes to find and fix errors.

Q8. I cannot afford hundreds of thousands of dollars for automation.

Good Answer: You don't need to. You can phase your investment at a pace that suits your organizational requirements and budget. There is no need to buy everything all at once.

Likeliness of hearing that good answer: Unlikely. The majority of tool vendors will want you to take everything all at once. This will probably overwhelm you and leave you with a lot of expensive shelfware. At Original Software we have a different approach to implementation. We call it Crawl, Walk, Run. You only buy what you need to make a difference at that precise moment in time, and move onto the next phase of automation only when you are ready. This cuts down the budget outlay means you can carry on with your current testing while you implement automation, and allows you progress steadily at a pace that suits you.*

*For more information see 'Three Steps to Successful Software Test Automation' - www.origsoft.com/Reading_Room/whitepapers.htm



In Conclusion

We have seen that poor software quality can carry with it a hefty price-tag. You can loose out to your competitors, reputations can be damaged, customers can become alienated, annoyed and even put in danger. All of this can and will hit organizations where it hurts the most: the bottom line.

Despite this, organizations still do not give software quality the time, resource, and budget to enable improvements in this area. The only way to ensure software quality is to test. Manual testing is time consuming, dull, full of repetitive tasks and extremely inefficient. By automating key areas of testing organizations can benefit from huge reductions is downtime and inefficiencies, and also improve the quality and the time-to-market of their application development program.

Software test automation is an essential investment for any organization with a continual application development program in place, and they are spoilt for choice for vendors, all of which claim to be able to automate their testing and save them large amounts of time effort and money. However, buyers beware. All software testing tools were not made equal. There are some questions to ask your prospective vendor. Do they allow you to test, manage and secure your test data? Do you have to be a specialist programmer to use the tool? How long will it take to get automating? Can tests be easily reused? These questions and more are important to ask and will allow you to make an informed decision as to the best tool for your needs.

About Original Software

Original Software offers automated software testing and quality assurance solutions that deliver tangible benefits across a wide range of IT and application environments. As a recognized innovator, Original Software's goal is to reduce business risk and improve application time to market for IT departments through the development of class leading automated solutions.

Over the last 10 years, more than 400 organizations operating in 25 countries have come to depend on Original Software for their software testing solutions Current users range from small software development shops to major multinationals, including: Cargill Global Financial Solutions, Barnes & Noble Bookstores, Royal Bank of Scotland, JPMorgan Chase, Debenhams, Pfizer Pharmaceutical (Ireland), DHL, Coca-Cola, Skandia and hundreds of others.

Original Software operates central offices near Chicago, and London. Their solutions can be obtained through these offices or through a network of qualified and knowledgeable business partners throughout Europe, the Middle East, Australasia and the Americas.

Original Software's range of powerful solutions includes:

Qualify - A complete Application Quality Management solution that unites requirements, cases, test execution, defect management and reporting within one platform.

TestDrive Assist- A new concept in testing that delivers active support for manual testing and compiles history on recent testing, making it easy to recreate and isolate software defects.

TestDrive A full-featured automated testing solution that allows technicians to define and execute sophisticated tests, without being hindered by complex programming languages. State of the art self updating technology automatically adapts tests to new software releases and upgrades.

TestBench- A suite of solutions that facilitate the management and manipulation of database and visual layer components. As Original Software's flagship solution, TestBench addresses test verification, disk space, and data confidentiality issues are all addressed. In addition, control of test data ensures that every test starts with a consistent data state, essential if the data is to be predictable at the end of testing.

www.origsoft.com
www.manualtesting.com

Original Software