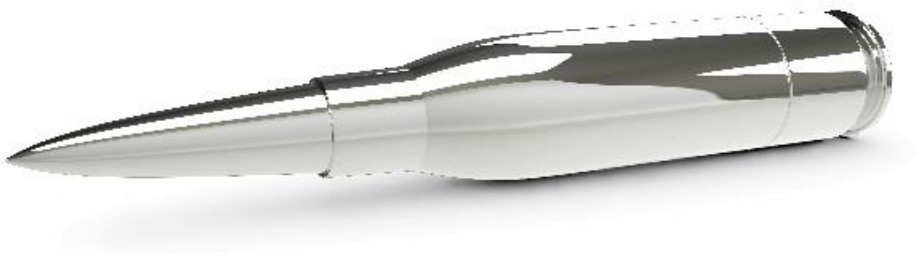
Test Automation Silver Bullet, Myth or Reality?



Original Software

Well obviously, it's a myth. Right? Maybe. But that said, why do many still seek it? It is still perceived to be the Holy Grail of testing and keeps millions of people in gainful employment every day, and to an extent that includes me.

Part of the problem is the lack of understanding in senior positions. They learn of automation successes and believe they can be applied directly to all their software testing problems and beyond. I am one of the first to talk about our clients' successes, and why not, they are successes and they should be proud of them. But they need to be considered in context, as that context is all important.

What is Test Automation Success?

Firstly, what type of organisation was it who had the success? The testing needs of CertainTeed are quite different from those of Cayman National Bank or The Edrington Group. The industry demands mean different things have higher importance in one compared to the other, and a major success for one might be of minor value to another. Just because one company tests the whole of its website content every night automatically without a single script and has a compete audit trail of the content to satisfy a corporate compliance program and changes, does not mean that will solve any of another's needs.

Then there is the technology. Different enterprises have different systems, some built in-house, some purchased, some consumed in the cloud. Many are going through an application modernisation program to transition from one approach to another. Their needs will depend on the nature of the programme. So, each organisation must decide what is relevant to them. Looking to the future it looks bright for the ERPs such as SAP 4/HANA, Oracle Applications, Infor M3, Dynamics to name a few our clients test, but especially for those available in the cloud. Who wants to write when you can purchase or subscribe? Testing a cloud application is a totally different proposition to testing an API or piece of SOA code in an agile development. So, unless you are one of the companies who are creating these ERP and enterprise applications, you will want to think about how you will test what you move to the cloud, and who will test it.

The “Where are we?” question.

Just like in the old joke about asking for directions which earns the response – “If I was going there, I wouldn’t start from here”. If you don’t know where you are, how you do things now, understand what, why and how you test, you are never going to make a success of automation. If your approach to testing is poor now, the jump to test automation will probably just make it worse. You have to gain mastery of what you have and do now and achieve properly organised Application Quality Management. But that doesn't mean you can't tackle test automation at all, it just means you need to choose the software testing tools carefully to be relevant to your need and to set out on that journey. Much is made of having fully documented test cases, but that does not mean you have to write all these up before any step. For example, you can automate the capture of business processes to create reusable test cases and build a library in your Application Quality Management solution. One of our clients Marston’s, took a joined up approach to the whole problem from business process capture to test cases to test automation and a regression pack for SAP. But there were different successes along the way and different testing tools used in the process.

Whose job is it?

Easy. It must be the QA team, the responsibility of the QA manager surely? Mind you, the QA team are not usually responsible for bug creation, so the developers must play a role. And the Shift Left argument proposes that if we did enough testing early on, we would not need a QA team, so maybe we should just make developers responsible for quality. But in the end, the users have to use it, to know how to use it, to have a good user experience (UX) and to be able to do their jobs well. So maybe they are the most important group, they certainly will be in a cloud testing scenario. In this case, they will need testing tools designed to support UAT. That might include automation, or a process of getting to automation by making manual testing and documentation easier in the first instance. This is much more Shift Right than Shift Left. We created TestDrive-UAT with this need in mind and because this area often consumes more resource than any other whilst having until now, any technology aimed at helping solve the problem, a growing problem. If you take a bigger view of the problem, you might see areas of synergy and mutual gain. For example, for a change or a

new system to be rolled out successfully users need to understand how to perform tasks. This is why TestDrive-UAT and other parts of the solution such as TestDrive-Assist create documentation and 'how to' videos as a by-product of the testing process. Is it part of software testing? Perhaps traditionally not, but it is part of getting a successful deployment. Ah, so it is everyone's job in some ways.

Test Data. Testing is all about data.

Good testing and a good testing strategy need to include test data. It's an area many have applied successful automation to and to Original Software it means typically data extraction, manipulation, protection and validation. In the past, I have seen the only successful use of legacy test automation tools to be in creation of new records for test data. I'm not saying that's a bad thing, but I have heard vendors and users talk about the number of automated scripts they run successfully every week, only to find these scripts don't do any testing, they just create test data. The whole concept of reusable protected test data thankfully provides a different strategy.

Living with Test Automation

If there is one thing that kills test automation it is script maintenance. It is essential to minimise this for longer term success. It is massively important in Agile methodologies as there is simply not time to have a long-winded approach to maintenance. A technical approach perhaps involving code may be acceptable if testing SOA components if the pieces are small and their functions and output do not change much. But when components are combined or if testing the User Interface (UI) then resilience is massively important. A coded approach to automation here will probably fail, it just creates too big a millstone on the rest of the project unless it is very well and expensively resourced. We have seen this so many times and it is the main reason why test automation has historically had a bad reputation. If you are developing code changes you have enough of a problem as it is without initiating a parallel test automation coding project with similar challenges. TestDrive reflects the pinnacle of code free UI automation. There is no code to change when the application changes. The way the solution is architected with data, checking, logic, linking and interfaces abstracted from scripts (process and steps) reduces the impact of change from the start. But also, the

patented features such as automatic UI annotation means that changing object technical parameters, names or locations are handled automatically. This is especially important where you don't have any control of them in your vendor supplied applications. But the busy-sense, self-healing, drag and drop interface, data drive and automatic picture taking capabilities all add to the minimised maintenance and maximised usability goals.

Reuse is a key goal which drove TestDrive's architecture, but also the approach to responsive web testing on mobile and other devices. This has meant that a single script can test any mobile device that the browser can emulate such as a Samsung phone or tablet, a Nokia phone, an iPad or iPhone, HTC etc. A single solution supporting eCommerce channels. Look for reuse.

Lessons in seeking the silver bullet

So perhaps there is a silver bullet, but you are going to have to find the right one for you and look after it. In order to make test automation successful there will be different tools deployed in different places by different people, one tool will not do it all. Each will have their own successes and criteria for success ideally in an integrated strategy to Software Quality Assurance. This is great, because you don't have to find a single silver bullet. There are many and they can be selected to fit the needs of each case and the people involved in the different phases, all the way from unit testing to UAT.

- There is not only one, there may be many.
- One person's or company's success may not map to your needs.
- Get good at testing first so you don't try and automate a bad approach.
- Choose tools that match the need and the skills.
- Do not build an on-going maintenance burden.
- Don't do it all at once.
- Do something, a small step if you cannot take a big one.
- Look at the big picture and all the areas that are involved.
- Look for actions that bring multiple benefits.

Maybe “Silver Bullet” is the wrong term, but the right, realistic and immensely valuable solution exists, that’s just a bit of a mouthful.

About Original Software

With a world class record of innovation, Original Software offers a solution focused completely on the goal of effective application delivery through 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 make quality and efficiency a business imperative. Solutions include 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 Allianz, Bimbo Bakeries, Costco, CertainTeed, Delta Dental of WI, Euronet. IAT Insurance, O’Reilly Autoparts, Cayman National Bank, Topcon, and DSC Logistics