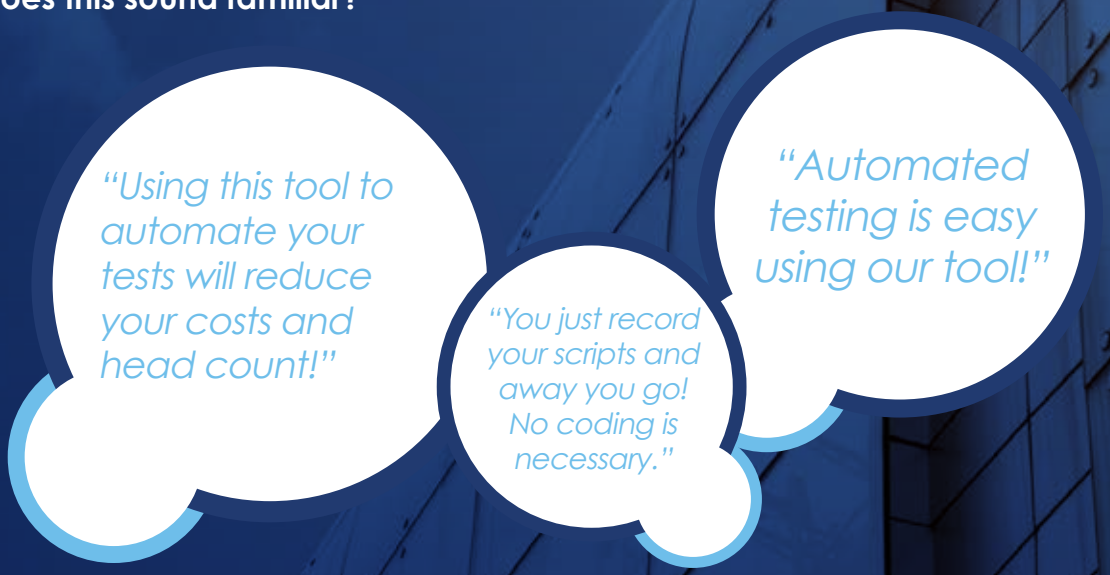
nFocus Testing

6 Reasons why automated testing fails to save money and 6 Things you must do to ensure success

Does this sound familiar?



TEST AUTOMATION

6 Reasons why automated testing fails to save money

How many times have we sat across the table from tool vendors telling us this!? The truth is, automated testing is not easy, it takes the right strategy and planning to give you a fighting chance of getting it right. In our experience of working with clients who have tried to automate their testing, there are a number of common factors that caused them to fail to meet their intended objectives or give them any cost savings:

1. 'Record and Playback' Automated Tests easily and frequently break and are time consuming to maintain. Changes to the application under test quickly render the tests out of date and useless or otherwise an expensive overhead.
2. Automated tests fail to handle errors correctly. It is difficult to identify the root cause of failure and often the execution run stops, requiring re-execution of all test cases once the problems are investigated. This results in a lot of unnecessary, wasted time.
3. It takes too long to automate functional tests 'In sprint' and the automated test coverage lags behind the system under test. Automated tests retrace the same steps over and over, and don't find new bugs. All this results in application code that is not fully tested being released.
4. The majority of functional test automation tools only test the GUI which is an inefficient way to test most of the system functionality.
5. A lot of time, effort and money is spent creating custom automation frameworks that are often inflexible and brittle. Most only work for a handful of applications and do not scale when new systems are created.
6. Automated testing is left to developers. Typically, developers do not have the same mind set or skills as testers and they do not test the right things. Who tests the automated tests?

Ultimately the single cause of failure was:

The people tasked with implementing test automation did not have the correct or varied experience to select the right strategy, tools or approach for their organisation. I have heard many customers say 'We thought we'd be OK because we got in a couple of experienced automation testers but it still cost way more than we thought'.

However, if the combination of tools, experience and planning are right, then automated testing can deliver rapid returns on investment and increased test coverage. With the correct solution in place it is possible that your projects will experience a reduction in testing time as well as the opportunity for more focussed testing on new functionality. The nFocus approach avoids all the above pitfalls and ensures the following six 'Must Do' things happen so that automated testing is robust and provides a cost saving to your organisation.

6 Things you must do to ensure success

1. Automate at the GUI, Services and Database Layers

Why would you only focus on automating the GUI tests when most applications are like icebergs with the majority of the complex functionality lying below the surface? Doing so has to be considered extremely high risk as well as delivering low value to the project and little confidence to the organisation.

What is required is a more holistic view of the application architecture. The approach we use has a common test framework for the GUI, Web Service and Database layers. This means that we can automate much more functionality in a common and robust manner. Consideration to automated testing should be given at the beginning of the application design and tests identified for each layer of the application architecture.

2. Be Tool Agnostic and Future Proof Your Test Architecture

In our experience, teams change (or at least want to change) their strategic test automation tool every 3 – 5 years. As the technology of the application under test evolves, so must the automation tool technology but unfortunately test tool vendors don't always keep pace with the development tool vendors. In order to future proof your solution, it should be execution tool agnostic, meaning that you can build your test assets (automated test scripts) and choose whichever automated test tool is most appropriate to execute your test cases. Our solution lets you do this without any additional overhead. Inherent in this is the capability to use multiple automation test tools. So if part of your solution requires Selenium but another requires CodedUI, then the two can be used with a common framework and consistent approach.

3. Automate Early

Historically automated testing begins at the end of the first release when the GUI has reached the point where it is no longer going to undergo much more change by development. At this point the application will have gone through many cycles of development and testing especially if we are talking about a multi-layer application. With this in mind, why would you wait until the end of the Application lifecycle to begin on the journey of writing automated tests? By referencing common object names used by the development and the test team, we write functional automated tests while the development team is writing the application code and their automated unit tests. This means that the functional automated tests can be written early in the sprint or development lifecycle and can be executed as soon as the application is released.

By building a common Object Map that is used by developers and testers alike, automated tests can be written by the testers in parallel with application code being written by the developers. This is NOT TDD or BDD: this is creating end to end, automated business process tests. Our approach makes it possible to execute new automated tests on new functionality within the same sprint.

4. Get Testers Doing the Automated Testing

If automated testing is more efficient than manual testing, shouldn't we be trying to get more testers automating? Our approach gives thorough-bred testers the ability to create and maintain robust, complex, automated functional tests while maintain coding standards and best practices. We do NOT advocate the use of Record and Playback tools which are easy to use but do not create robust or very good tests and are dependent on a finished product before automation can start. We write our automated tests using a unique code generating platform that enables test scripts to be executed by a range of automation tools such as QTP, Selenium, Rational Functional tester and Microsoft's Coded UI.

This provides the ability to reduce the number of highly technical resource whilst at the same time opening up the opportunity for skilled testers to contribute to the creation of automated tests.

nFocus Testing

5. Create Build Verification Tests and Regression Tests

As well as functional testing, focus should be given to creating automated test packs to cover Build Verification Test and Regression Tests so to get as much benefit from automation as possible. These tests can also be incorporated into the continuous integration process which in turn will enforce a more robust build process by providing instant feedback to the development team when code is built and deployed.

6. Manage Test Data Well

Having the correct test data is critical to the success of all testing but it is particularly important for automated testing as it is more difficult to notice and identify poor test data that could lead to false positive results. To ensure that automated testing is successful, defining, creating and managing test data is just as important as defining, creating and managing your tests.

About nFocus

nFocus is the leading UK based software testing provider. Our clients include Government and blue chip organisations such as News International, Marks & Spencer and Barclays and we have a long-standing reputation for delivering successful automated testing solutions.

About nFocus' Automated Testing as a Service Offering

Who says you can't increase quality while reducing costs and timescales? nFocus have a unique and innovative approach to test automation that works for clients; we will design, create, maintain and execute your automated tests on demand. We have a results based pricing model that means you pay a cost per test and only pay when working automated tests are delivered.

Our automated testing service uses a rigorously constructed method to ensure a positive return on your investment and successful outcome for your automation approach, saving time and money.

Testimonial: "It was taking a week for our 7 offshore testers to manually execute a suite of 567 tests, now it's done in 4 hours overnight. We have moved from a monthly build cycle to 2 or 3 builds a week!" - **Alan Wallet – Test Manager, ING Direct**

Want to learn more?

If you want to automate your regression tests more cost effectively or have tried and failed to automate in the past, then call **0870 242 6235** or **email info@nfocus.co.uk**

VISIT nFOCUS ONLINE
WWW.NFOCUS.CO.UK