

Overcoming Usability Challenges With Test Automation Scripts

Ten10

Client: A global investment bank

Industry: Capital Markets

Technologies: FitNesse, Java, Python, Elasticsearch and AngularJS

Ten10 Services: Quality and Test Strategy Consultancy, Test Automation

A leading global investment banking, securities and investment management company, this client provides a wide range of financial services to a substantial and diversified customer base. Ten10 was engaged to improve the automated testing already in place and relied on by the client to deliver the reporting required to meet strict financial regulations.

What We Did

Following the crash of the financial markets in 2008, much stricter regulations were introduced in order to govern transactions that financial institutions were carrying out. As such, financial institutions are required to report on the transactions that they perform and to provide information about these transactions to several regulators.

To meet these reporting requirements, the client had implemented a solution using FitNesse (a stand-alone wiki and acceptance testing framework) that would drive Sikuli automation scripts to book financial transactions. The solution would then check the automatically generated FPML report and send it off to a regulator. Errors in reporting could potentially result in penalties for the client.



As part of the test execution, the solution built into an existing test execution farm, allowing computational resources to be dynamically assigned to allow tests to be executed.

The framework and automation scripts worked from a technological point of view but offered usability challenges. The challenges mostly related to the definition and implementation of test scripts, whereby the knowledge of what needed to be reported and how those transactions should be booked laid solely within business knowledge.



When trying to onboard the business operations resources onto the system to write test cases, the client faced several pitfalls. Firstly, the FitNesse wiki format did not prove intuitive for the users – the syntax of writing tables and providing values consumed a lot of time, effort and required upskilling.

Secondly, the solution did not provide any indexing capabilities of tests that had already been written. The client was, therefore, finding that they were often duplicating test scripts as resources were not aware what scenarios had already been covered. The solution also offered several other complexities related to test design, execution and results gathering.

Goals

Ten10 was tasked with defining, designing and implementing a solution to this problem. Rather than fundamentally changing the way business resources were writing test scripts (which would mean re-writing existing test scripts), Ten10 established that a mechanism was required to leverage the existing scripts whilst providing a more intuitive way of creating, maintaining and executing scripts going forwards.

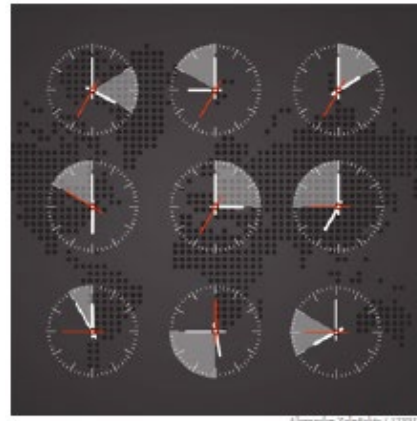
Ten10 also identified that a tagging mechanism was required for test scripts. This would allow the users to search existing test scripts and establish whether the business process that they were tasked with testing had already been covered by an existing test case. Further, the solution would provide management with the capability to see, for example, whether a particular asset class had more test scripts than another.

Of paramount importance was the usability of the system. Ten10 needed to design an intuitive and smart user interface (UI) in order to allow non-technical resources to create the required transactions with their specified parameters and then to check that the output report was being populated with the correct values.

Challenges

The client is a very large company with team members dispersed globally. Ten10, therefore, needed to work with people in places as widespread as Japan, India and the USA as well as the UK; co-ordination across many timezones was required.

With so many stakeholders involved from varying locations, a key challenge was identifying the core requirements of the solution whilst designing software that would satisfy several, often differing, requirements.



Pragmatic Approach

In order to understand the issues experienced, Ten10 carried out an initial consultation period with client resources in New York. Following the initial consultation, Ten10 delivered a solution to the problem in an iterative, agile fashion that was received very positively.

Utilising experience across a number of different industries and its strong SDET capability, Ten10 designed, developed, tested and deployed a solution for business users that strongly enhanced the client's testing capability.

Rather than fundamentally redesign the way the client built their test scripts, Ten10 established a solution that could leverage the existing scripts and utilise the platform that the client had already built. Utilising technologies including Java, Python, Elasticsearch and AngularJS, Ten10 built an intuitive web-based solution on top of the client's existing SDLC that satisfied stakeholder requirements and solved operability issues.

The solution included features such as test tagging (including scanning existing scripts to derive tagging metadata), auto-complete suggestions, test-script duplication protection and a convenient test execution platform that significantly enhanced the client's efficiency and accuracy when writing test scripts, as well as providing a non-tech friendly user interface.

For Ten10, the key to success was taking a step back from existing solutions and analysing the wider problem. Based on a solid understanding of the issues that the client faced, Ten10 was able to design a solution that did not require a gross amount of rework and proved a great success.