How Adopting An Agile Approach Reduced Regression Testing Time By 70%

Ten10

Client: A £1.2 billion online B2B retailer

Industry: Retail

Technologies: Java, Selenium, Cucumber, Gherkin

Ten10 Services: Quality and Test Strategy Consultancy, Test Automation

The client is a global B2B retailer of electrical and electronic components, tools, testing equipment and related items across the web with customers in 32 different countries.

The Project

Historically the client had a robust manual test process in place which was heavily dependent upon contractors. However, the frontend system had to change rapidly in response to new products, customer requirements and new features. The client wished to adopt new ways of working to allow them to move to an Agile approach and enable software to be released faster and more frequently.

What We Did

Ten10 worked with the client to successfully help them transition from a manual, functional regression testing process to an agile delivery methodology, introducing continuous testing and providing automated regression testing that could be executed with confidence.

As a result the full regression testing time was reduced from one week to 1.5 days.

Automated tests ran three times per day continuously – providing constant reporting and feedback on all applications, including on a variety of browsers across all the regions automatically. The greatly improved integrated and Agile approach to testing was complemented by the introduction of user-friendly visual reporting, ensuring key stakeholders had access to information in a timely manner.

Challenges

Despite an appetite for change, the full value of some of the new techniques were not being realised by the client. An automation regression suite had been created and was running on one of the eight applications. However, there was little confidence in the quality of the information and results produced by the test pack. Specifically, the entire pack would produce a single binary pass or fail upon execution. Confidence in the results was so low that the same tests were then being run manually to confirm the results.

Ten10

The client was facing challenges around the management of new features and fixes into live. Each time a change was made the entire software estate had to be deployed as a single unit, regardless of whether a change had been made to a particular application or not. This increased the volume of testing as all eight products had to be regression tested following a single code release, irrespective of how trivial the changes were.

A further consideration was the need to deliver the product suite across 32 different countries. This created a significant regression backlog, either with functionality not being tested or testing taking a considerable period of time.
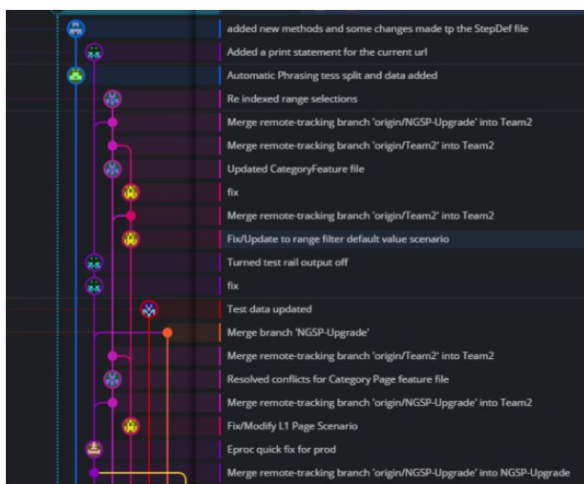
There had previously been a number of projects undertaken to modularise the code base, allowing the monolithic code to be broken down into a number of smaller, discrete components that could be tested and released independently.

Key restrictions included; the complexity involved in testing all the modularised applications to a level of suitable functional coverage, testing across all the different markets and across all the supported browsers and the overall length of time it would take to complete. Due to these complexities and associated risks, the historical projects to separate code in modules had previously been unsuccessful.

An additional challenge was the client's three week sprint cycle; development was undertaken for nearly two weeks with the final week devoted to a full regression test before release. This significantly reduced the effectiveness of the development team by one third and the testing team by two thirds.

The Project Goals

The first goal of the project was to successfully modularise the code to enable faster releases. After this a primary goal of the engagement with Ten10 was to successfully adopt more of an Agile development methodology; incorporating testing throughout the sprint and allowing the client to spend the full three weeks developing and testing new features rather than just two weeks out of a three week sprint.

The client wanted to be able to see fast feedback when changes were made to fix bugs, so another objective was to improve the management, reporting and visibility of their defects, as well as to give early feedback of the application's current state.

There was also an overall requirement to significantly reduce costs as well as to improve efficiencies through new testing techniques and approaches.

Modularisation To Support Agile Testing

Ten10 provided strategic advice on where to focus effort and resource to ensure the initial six week project to modularise the code was successful, whilst also deploying a team of four testers to work alongside the in-house team of developers. During this period all development was focussed on the modularisation of the code.

Using pre-existing and approved documentation for the test scripts, it was agreed that the initial goal should be expanded to include automation of test scripts for each module.

For three of the eight applications, the original specification was followed, however, for the other five applications, it was identified that additional requirements were needed, so these were scoped and created.

The modularisation project was delivered successfully on time, with the first application ready to go into full regression testing three weeks into the project. 64 critical defects were found from the automated tests prior to go-live. Following go-live only three early life support defects were discovered, all outside of scope. Furthermore, the discovery of those critical defects was 'left-shifted' and all were picked up in the integrated window.

Following successful deployment into live, the team then implemented a solution around the automated test pack. The full regression pack was now run three times per day, with HTML generated reports displayed on big screens in the agile delivery team office space. This visual reporting gave the client the fast feedback they required. If there was a failure, it was possible to drill down into the details there and then (including screenshots, test code and stack trace) for rapid insight into the issue.

Internationalisation Of Regression Testing

The regression testing automation packs also had to be reprovisioned to ensure they would meet the global requirements of the 32 different countries, with each market running on a number of warranted versions. In order for the packs to work, they were refactored, so they were not reliant on any labels or names in English.

Additionally, Ten10 undertook to remove deprecated CSS on the webpages. In order to ensure this didn't have any negative impact, a tool was created to compare screenshots pre and post-change to check for any rendering errors and was run across a matrix of browsers and languages using Browserstack. If differences were found above a configurable error threshold, a heatmap was generated which showed the changes at a glance, and which was then sent to a tester for manual checking.

Cost-Effective Software-Developers-In-Test

To meet the cost-efficiency requirements, Ten10 was able to heavily leverage its graduate Academy to provide the dedicated, software-developer-in-test resources required by the client at a competitive on-site cost.

Running since 2013, Ten10's graduate Academy takes the brightest and best young talent and trains them in Agile development methodologies, testing, technology, performance, automation and mobile testing. Ten10's bespoke training, delivered by experienced practitioners who are also certified ILM trainers, ensures our graduate testers are equipped with a balance of theory and effective, pragmatic test techniques. Through this model we were able to ensure the client had the skilled resource required for the project whilst still meeting their cost-efficiency targets.