Creation And Implementation Of A Test Automation Framework For An HTML5 Application

Client: An international financial services provider

Industry: Capital Markets

Technologies: Angular JS, Protractor, TeamCity, Git

Ten10 Services: Test Automation

The client is a global bank operating in over 40 countries, with over 55,000 employees, with £12 billion in revenue. An international financial services provider operating on the basis of cooperative principles and offering retail banking, investment banking, private banking, leasing and real estate services.

The Project

The Client was redeveloping a number of applications in HTML5 and was keen to implement a more test-driven approach to its UI development strategy.

The client selected an in-house application used for managing, calculating and publishing curves for the Central Treasury group as the first project to be developed in this way.

As part of the new QA process for the redevelopment of these applications, the client was looking to automate tests so that they could be integrated into their continuous integration (CI) processes and provide immediate feedback on the stability of new code builds.

What We Did

- Generated a test automation framework and defined an automation process which formed the strategy of all HTML5 UI development going forward.
- Screencasts were analysed to extract test scripts.
- The extracted test scripts were automated using Protractor and the framework.
- The solution was integrated into TeamCity.
- A handover was provided to the client to enable it to take on the work after Ten10's involvement.

Challenges

- No test automation capability existed.
- The client's system was under active development, meaning a large number of changes whilst scripting for test automation.
- There was a lack of a stable environment.
- Test data issues were resolved on-demand in collaboration with the development team.

Ten10's Approach

Ten10 developed a Proof of Concept (POC) for the test automation framework based on the application with a view to applying structure to scripting and best practices for the delivery of test automation.

Key Elements Of The Delivered Automation Framework Included

- Use of Protractor, which couples with AngularJS applications.
- Definition of a structured and easily readable approach to writing test cases.
- Adoption of the PageObject pattern for test automation to improve maintainability.
- Loading of hierarchical test data from a JSON file to allow for test data reuse.
- Integration with TeamCity and Git to provide quick feedback on results.
- Execution of parameterised builds against different environments.
- Creation of reports and screenshots for each run to facilitate the identification of issues.

The POC was followed by test analysis of the application. Specifically, the client recorded and provided a number of screencasts which described functional tests that were performed manually against the application. These were analysed by Ten10 to identify the complexity of automation and to estimate the effort required in the development of the corresponding scripts.

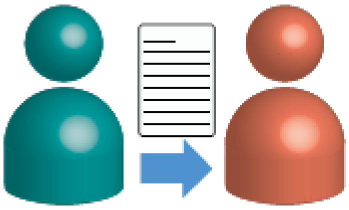
Ten10 proceeded to the delivery of a frontend, UI-level test automation regression pack, which would integrate with TeamCity and run against the latest release of the application, reporting results directly onto the CI environment.

Test Automation Framework Best Practice

The framework designed and implemented by Ten10 used Protractor, which offered test automation best practices such as separation of concerns, hierarchical test data loading from JSON files, page-object model and multi-browser support.

The framework was also used to create an effective regression test pack with training for the client provided by Ten10, enabling them to confidently take on responsibility in-house for their regression testing.

During the engagement, Ten10 worked closely with the development team to ensure stability and test data issues could be raised as early as possible, avoiding any blockages or delays to the regression pack automation progress.



The Results

Ten10 successfully met the client's goal to establish a departmental UI-level test automation approach. Following Ten10's engagement, test automation has successfully become part of each sprint. The continuously extending coverage of automation scripts has significantly reduced the manual testing effort requirements, increased the speed of testing, and reduced the risk of a critical defect passing into a release.

Benefits

Due to the demonstrated benefits, the client has now adopted this automation approach and framework across all its development streams, thus underlining the success of the engagement.