Application Performance Testing Using Microsoft's Silverlight And Visual Studio

Client: A leading law firm, specialising in personal injury claims management

Industry: Legal

Technologies: Microsoft's Silverlight and Visual Studio

Ten10 Services: Performance Test and Engineering

The client's in-house customer call centre and dedicated case handlers were experiencing performance problems with their personal injury claims management application. The application had been developed by the client's in-house team using Microsoft's Silverlight and Visual Studio products. Ten10 was engaged to select an appropriate test tool, create performance test scripts, automate reporting and train the client's staff.

What We Did

- A performance testing Proof of Concept (POC) was carried out which identified two main issues.
- Tool options were reviewed during the POC and a best-fit tool solution recommendation was provided.
- Easy-to-follow scripted web tests were created to overcome Silverlight application challenges.
- Automated reporting was set up for user load, average response times and transaction-based response times with error rates and error details.
- Training was provided to enable ongoing inhouse management of performance testing.

Performance Testing Proof Of Concept

Ten10 was initially invited to produce a Performance Testing POC. The short, fixed-term engagement was used to explore various performance testing options including; Visual Studio load testing using: a) UI Automation; b) Coded UI; or c) Web Test options, as well as evaluating the use of Ranorex (another UI-based product).

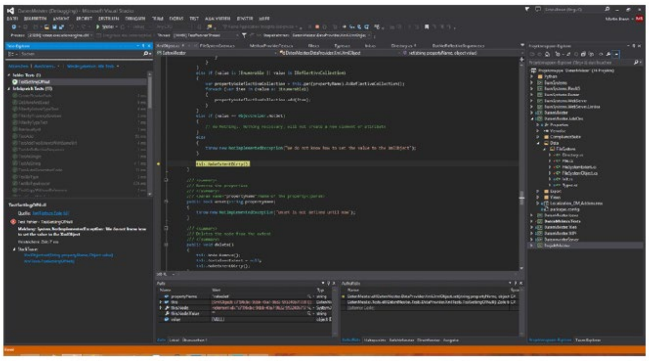
During the development of the POC, Ten10 uncovered two issues within the application; one minor functional issue and one performance issue which had not been previously identified.

At the end of the POC, Ten10 recommended the use of Visual Studio Web Tests, complemented with Ten10's own reporting extensions as the best-fit performance testing solution to meet the client's requirements.

Overcoming Silverlight Application Challenges

One of the limitations of the Visual Studio tool was the inability to decode, modify and encode messages recorded between the Silverlight client and server, as they were encoded using Microsoft's proprietary .NET Binary Format for SOAP (NBFS).

Ten10 was able to overcome this challenge by developing a series of plugins within Visual Studio to convert the binary format into plain-text XML data, allowing the recorded messages to be parameterised from databanks in order to load test the system and to decode the responses back from the server.



Using this approach enabled Ten10 to deliver easy-to-follow scripted web tests, with the power and flexibility of coded modules (where necessary) to create scripts which could be maintained by the client's automation testers, despite having no previous experience of using Visual Studio's load testing tool.

Why Web Tests?

Web Tests are easy to record, easy to understand and easy to maintain. A benefit of using the Visual Studio tool was the ability to add a collection of web tests to a load test plan, whilst remaining within a familiar Visual Studio environment.

The load test plan included details such as the overall user mix for each script and the ramp up/down times. Furthermore, the scripts were data-driven, enabling more complexity to be added to the test plan. Finally, for complete flexibility, dedicated code modules could be used interchangeably and as required.

Meaningful Reporting

Another limitation of the Visual Studio tool was the out-of-the-box performance graphs and reports. There was only a basic level of information provided in the reports and they were not particularly user-friendly to interpret.

Ten10 generated a series of modules and scripts which integrated with the Visual Studio load test database, enabling a performance test report to be generated automatically showing the user load, average response times, transaction-based response times along with error rates and error details. The reports included a summary as well as detailed graphs and allowed for the narrative to be added by the individual tester.

Training And Ongoing Management

To conclude the project, Ten10 conducted a number of training sessions with the client's in-house test team, as well as producing 'how-to' documentation on the delivered scripts, enabling the internal team to maintain them as the business evolved.

Following the project closedown and handover, the client's internal test team took control of the maintenance and execution of the performance test solution.

An in-depth understanding of the client's requirements, together with consideration of the existing resources and capabilities, enabled Ten10 to identify the right solution to answer both the immediate performance testing needs as well as ensuring the client's longer term support requirements could be met through their available resources.