



Case Study

Automation Framework Maintenance and Enhancement using Selenium RC

Your company information

Branch	Sydney, Australia
Website	www.adactin.com
Contact name	Sapna Bhatia
Title	Business Manager
Phone number	+61 - 420983561
E-mail address	info@adactin.com ; sapna.bhatia@adactin.com
Solution group	Testing Services Group
Solution offering	Fixing and Writing New Automation Scripts in Selenium RC
Project name or title	Updating Selenium Automation Framework

Case Study for Maintenance and Enhancement of Automation Framework using Selenium RC

Client profile

Client is a community service organization that has been transforming the lives of Australians in need for more than 150 years. Client provides: accommodation and support programs for the homeless; creative services for homeless youth; counseling and support programs for families and accommodation and support for people with disabilities. 3,500 employees, volunteers, Board and supporters advocate for disadvantaged Australians and help them to get back on their feet. Working with government, corporate Australia, churches and the wider community, Client is determined to overcome disadvantage across the nation.

Business situation

The client had web-based, secure, client record management system that helps to create and maintain an electronic version of records. Application used advanced Ajax controls.

Client used Selenium RC as automation tool for testing and the script were written in java using Eclipse as an IDE. They were using existing Excel driven framework. Since the application was being incrementally enhanced with new features and caused changes to the workflow, existing scripts starting failing. Also the framework gave memory errors when executing scripts in batch mode.

New functionalities added required design change in framework to get existing scripts to work. Client wanted to get these scripts fixed and develop new scripts according for new functionality

The Application had single and multiple pop-ups which were not handle by Selenium so the client also wanted to handle these pop-ups during execution of the script which required adding new functionality in framework.

Technical situation

Below was the technical situation and need of the client

- Install Eclipse and setup of Selenium RC on 64 bit machine to get rid of memory leak issue
- Analysis of existing script to find the time required to fix these scripts
- Fixing of existing scripts in a given timeframe
- Install Autoit ,for handling the Popup issues
- Developing new scripts

Solution

Adactin proposed and implemented gradual update process which was divided into 5 Phases

Phase 1-Analysis of memory issues in framework

- Analysis of real cause memory issues. It was found that script were failing because automation suite was being executed on 32 bit eclipse which had less java heap size. After moving the application to 64 bit eclipse and OS, memory leaks were resolved
- Setup of 64 bit Automation framework System
 - Install Eclipse and setup Selenium RC
 - Adding External Libraries and jar files
 - Install java
 - Install Autolt

Phase 2-Analysis of Existing Failing scripts

- Analysis and get to know the Application functionalities
- Setting up the testing Environment
- Setup of automation framework environment
- Running all the scripts
- Analyzing the Issues that causing the script failure
- Making Report of those issues according to their complexity and Time required for fixing
- Analyzing issues in the java code which is not executing some methods correctly apart from workflows

Phase 3- Fixing of Failed Scripts

- Setting up the testing Environment
- Setup of automation framework environment
- There were following issues that were solved
 - Entering the Data in TestData.xls for new functionality
 - Introducing new keywords in the framework and in the scripts
 - Debugging and fixing the java code to solve the code issues and making the methods and variables work
 - Entering the missing Xpath and correcting the Incorrect ones
- Implementing Popup Solutions
 - Understanding the workflow of the application where we getting the popup
 - Making Popup specific Autolt exe
 - Writing the script to call these exe

Phase 4-Writing New Scripts

- Setting up the testing Environment
- Setup of automation framework environment Writing the Test Steps and making the new Keywords as per workflow in Framework.xls
- Entering the required Test Data in Test Data.xls
- Entering the Xpath for new fields
- Making the particular java file in eclipse writing the code for the functionality
- Running all scripts

Phase 5-Knowledge Transfer

- Making Reports of all the Updates and fixes
- Knowledge Transfer of all the updates and fixes that were made
- Providing Training to the client internal automation analyst

Benefits

Find below benefits of technical solution proposed to the client

- Testing on 64 bit machine is faster and scripts were being executed overnight in batch mode
- Automation Regression Testing for all functionality save 40% time of testing
- Pop ups were now handled during the automation testing through Autoit
- Reduced execution time of scripts using Framework which led to further increase in productivity
- As knowledge transfer process, documents were designed on how to debug and enhance the framework assisting in further maintenance of automation framework

Products and services we used

- OS- Windows2007
- Database –SQL Server 2000
- Java
- Automation Tool – Selenium RC
- Autolt

Assistance provided by client resources

- Assistance provided by client's IT team in setting up 64 bit machine
- Client's Testing team helped in setting up Testing Environment
- Client's Testing team provided Test Data and Scripts

For More Information

- For more information about Adactin products and services, call us at +61-420883561 or +61- (02) 86771032
- Or email us at info@adactin.com
- To access information using the World Wide Web, go to: <http://www.adactin.com>

© 2012[AdactIn]. All rights reserved. This case study is for informational purposes only. AdactIn MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.