CASE STUDY ON REGRESSION TESTING

CLIENT

Our client is a US based highly scalable in memory self service, visual data mining, discovery and analysis systems, which are used by business and public sector clients worldwide.

APPLICATIONS AND REQUIREMENTS

1. POC to search tools to cover maximum scenario of application.

2. Automation bookmarks for validation of Data in DataGrid, Graph, Tooltips etc and also validate bookmarks data by exporting run time with saved data set.

3. Automation of defects logged into the Lotus defect tracking tool.

4. Setup report to access using URL with in network.

5. Setup automation test suite with Jenkins for nightly build execution.

6. Configuration of email for notification of execution.

7. Test suite should execute after RDC machine close.

TOOLS AND TECHNOLOGY

1. Testing AnyWhere

2. Ranorex

3. SilkTest

4. TestComplete

LANGUAGES

1. C#

2. VBScript

3. Java script

OTHER TOOLS

1. Jenkins

2. IIS Server

PROBLEM STATED/CHALLENGES FACED

AT THE START, CLIENT SUGGESTED TO AUTOMATE APPLICATION USING TESTING ANWHERE, WHERE WE FACED THE FOLLOWING CHALLENGES:

I. Unable to capture (Object) of Data Grid cell, so data could not be captured from Grid cell

II. Unable to capture object of TreeView's row .

III. We were unable to build our own logics to perform action on AUT, only VB scripts supported and there is limitation to use.

IV. In some cases when we used keyboard shortcuts like to save file (Alt + F or S) or use arrow key to navigate up and down, it created issues in case focus change.

V. Unable to scroll the mouse on graph cell to capture tooltips value.

VI. Tools verified image checkpoint using screen coordinate, so if image position was changed, it failed in result. If check points were applied in sub scripts (function) and called them main scripts, then checkpoints status does not show in execution report.

AFTER LIMITATIONS OF THE TESTINGANYWHERE TOOL, WE WENT THROUGH OTHER AUTOMATION TOOLS AND DID A SMALL POC TO COVER SOME POINTS AS MENTIONED BELOW:

I. Ability to test MFC applications on Windows 7 and Windows 8 operating systems

II. The application had right click functions and menus that required testing. We need to be able to right click in a specified area of the window and find that location based on an image or a word vs. coordinates.

III. Search data fields on the exception dialog.

IV. Easily maintained and ability to create own logics like create custom function, reusable function, Suite, Test case, put setup and teardown in test case.

V. Image verification.

VI. Can export analyses and compare created file to baseline file.

VII. Able to run a script on any machine.

VIII. Must support dual (or more) monitors.

IX. Recognize images in a "toolbar" and tell if they are active or not.

X. Work in a virtual environment and from a remote desktop session.

XI. Test Graphs produced by our software to be sure that they are drawn as expected.

XII. Script readability is very important. Anyone should be able to pick up a script and be able to tell what it's testing. Of course there is still a need for comments in the scripts.

XIII. Is it compatible with application testing in a Citrix environment

XIV. Automate bookmark DataGrid from data grid like verify data, perform click action and perform ruberbanding.

XV. Handle the complex scenario of application.

XVI. Capture tooltips of graph cell.

XVII. Generate reports based on many different criteria (just about anything that's tracked), setup report in server for each execution so that anyone can access.

XVIII. Mail configuration for each build execution and also result of each execution could be saved.

XIX. Centralized repository, integration of Test Suite with Jenkins.

XX. Modify object repository without changing in code.

PROCEEDING AND SOLUTIONS

We researched automation tool and verified all points with automation tools: TestComplete, SilkTest, Ranorex, TestingAnyWhere and finally selected Ranorex automation tool.

RANOREX SOLUTION

1. To handle Grid cell data we captured Grid Data Object and parameterized object using row and column of cell, by passing row and column we navigated and verified all of data grid.

2. We followed a similar approach like the above, in graph cell to verify tool tips. Also used image verification to verified graph

3. For customized reporting, Ranorex has some advanced functions to customize the report.

4. To create own logics we used C# programming language and created a library.

5. Report to maintain history, we integrated with Jenkins CI tool and also configured mail to send email for each build.

6. We setup IIS server and moved generated report into IIS server project folder to access globally. Also each execution report using with suffix of build number was maintained.

7. Created TestSuite and Test cases

8. Parameterized all static values from Suite level and passed into test scripts.

9. We used tscon command to run test suite while RDC machine closed.

10. We could create scripts in two ways keyword view and code view and divide into section, give proper comments which make more readable test scripts.

11. Ranorex has ability to run on Citrix environment and support dual monitor.

ADVANTAGES OF THE TOOLS USED

1. By creating central repository, if any changes occur in the object locator it will reflect in all test cases.

2. Parameterized in object, it reduces the redundancy of objects; we can use one object for multiple elements by just passing values.

3. Setup Report with IIS server - it maintains history and any one can access from anywhere to see the reports.

4. To Setup Jenkins, Ranorex automatically executes test and report will be sent to the registered email.

5. Reduces the duplicate of functions in a created library

6. Ranorex saves time to validate bookmarks and defects.

BENEFITS TO THE CLIENT

1. Saved time and cost involved in Regression Testing.

2. User friendly reporting via email and IIS server.

3. Can focus their attention on the right areas by analyzing execution reports, sent via Jenkins email.

4. All Bookmark data correct validation without any mistake without calculation manually.