Review Article **ISSN: 2319-507X**
Nirmal Kumar A, IJPRET, 2013; Volume 2 (5): 81-91 **IJPRET**

INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

A SYSTEMATIC REVIEW OF AUTOMATED SOFTWARE TESTING TOOLS

A. NIRMAL KUMAR[1], DR. B. G. GEETHA[2]

1. Assistant Professor, Department of Computer Science and Engineering, Christian College of Engineering and Technology, Dindigul, Tamilnadu - 624619, India.
2. Professor & Head, Department of Computer Science and Engineering , K S Rangasamy College of Technology, Thiruchengode, Tamilnadu - 637215, India.

Accepted Date: 25/12/2013 ; Published Date: 01/01/2014

Abstract: The software can be tested in two ways namely manual testing and automated testing. The process of testing the software by manual inputs by human is called manual testing. The process of testing the software with the help of software tools is called automated testing. This paper presents about how the software are tested with the various testing tools. Even though, the software development life cycle includes various phases, only the software testing can ensure the software quality. The some of the automated software testing tools are WinRunner, LoadRunner, Quick Test Professional, OpenSTA, Test Director, SilkTest, Rational Robot, Silk Performer, Astra Load Test and Astra Quick Test In this paper, the various automated testing tools and their working procedures are explained briefly.

Keywords: Software Testing, Testing Tool, Test Cases.

PAPER-QR CODE

Corresponding Author: Mr. NIRMAL KUMAR A.

Access Online On:

www.ijpret.com

How to Cite This Article:

Nirmal Kumar A, IJPRET, 2013; Volume 2 (5): 81-91

1. INTRODUCTION

Manual testing is a time consuming process. The automated testing speeds up the testing process. Manual testing requires a heavy investment in human resources. In automated testing, we can create test scripts that check all aspects of application. The benefits of automated testing are fast, reliable, comprehensive and reusable. The tools which are used only for testing is call as an automated test tools. The test tools are classified into functional test tools, performance test tools, test management tools. The functional test tools are used for checking only the functionality of the software (WinRunner). The performance test tools are used for checking the performance of the software or system. Something like load testing and stress testing (LoadRunner). The test management tools are used to manage the entire testing activity like creating test plan, test case design, defect tracking (Test Director).

LOAD RUNNER

Load Runner reduces an environment in which thousands of users work load with a client/ server communication system continuously. So, Load Runner replaces the human user with a virtual user (Vuser). The actions that a Vuser performs are described in a Vuser script. The Vuser script generator also known as VuGen enables us to develop Vuser script for a variety of application types and communication. VuGen creates the script by recording the activity between the client and the server. LoadRunner replaces human users with virtual users or Vusers. By increasing the number of Vusers, we increase the load on the system. For example, you can observe how a server behaves when one hundred Vusers simultaneously withdraw cash from a bank's ATMs. we divide our client/ server performance-testing requirements into scenarios. A scenario defines the events that occur during each testing session. For example, a scenario defines and controls the number of users to emulate, the actions that they perform, and the machines on which they run their emulations. The actions that a Vuser performs during the scenario are described in a Vuser script.

A controller reads a single scenario to co-ordinate several host machines which specify the use of different run-time settings, running different Vuser script and storing results in different locations. A load runner scheduler on a controller machine executes scenario automatically on a pre set date and time. After running a scenario, you can use load runner's graphs and reports to analyze the performance of your client/server system. VuGen creates a Vuser script by recording the actions that you perform on a client application You can monitor scenario execution using the LoadRunner online transaction and server resource monitors. LoadRunner provides the following online monitors: Server Resource, Vuser Status, Transaction, Web. Vuser

Groups are a collection of Vusers with a common property, for Example, a common script. We can place your Vusers into different Groups to classify them according to their tasks. During scenario execution, you execute individual or multiple Groups. We define transaction to measure the performance of the server. Each transaction measures the time it takes for the server to respond for a specified Vuser request. These requests can be simple tasks when waiting for a response for a single query, or complex tasks when submitting several queries and generating a report.

To emulate heavy user load on our client server system, we synchronize Vusers to perform a task as exactly the same moments. We ensure that multiple Vusers act simultaneously by creating a rendezvous point. When a Vuser arrives at the rendezvous point the controller holds it until all Vuser participating in the rendezvous arrive. When the rendezvous conditions are met, the Vusers are released by the controller. we designate the meeting place by inserting a rendezvous point into our script. When a Vuser executes a script and encounters the rendezvous point, script execution is paused and the Vuser waits for permission from the controller to continue.

SILK TEST

Silk Test is the functional and regression testing tool. It works on Object Oriented Concept. It has Record and Playback Option. It is the Browser and Platform Independent. It is used Link Testing. It has two components 1) Silk Test Host Software 2) 4Test Agent Software. The processes in Silk Test are Creating a Test Plan, Recording a Test Frame, Creating Testcases, Running Testcases and Interpreting Results. A Testplan consists of two distinct parts: an ***outline*** that describes the test requirements, and ***statements*** that connect the outline to the 4Test scripts and testcases that implement the test requirements. All Test Plan files are stored as sample.pln. The Test Frame is the one which contains the descriptions for the GUI Objects of the application under test. All the GUI information are stored in the file called as Frame.inc. The test frame is the backbone that supports your testcases and scripts. It is an include file (.inc) that contains the 4Test declarations for the main window and all its menus, as well as a generic declaration that is valid for each of the standard message boxes in the application. The GUI information refers to Window Declarations. All the Window Declarations consists of Class, Identifier and Tags.

The Testcases are generated by the normal recording process (i.e) nothing but the user actions on the application. During the recording process press Ctrl+Alt to verify the state of the particular object. The verify dialog will be displayed with the captured object properties. Click

OK to verify all the properties of the object. Invoke the particular testscript and run them using the Run Test Button. After the successful run, the result file will be displayed to show the test status. The features are Platform Independent, Browser Independent, Technology Independent and Test Window based Applications.

RATIONAL ROBOT

Rational Robot is a component of the Rational suite of products. The other components are Rational Administrator, Rational TestManager, Rational LogViewer and Rational SiteCheck. Rational Robot is an automation tool that can be used for following:

- Perform full functional testing.
- Perform full performance testing.
- Create and edit scripts using the SQA Basic & VU Scripting environment.
- Test applications developed with IDE

Robot is a powerful tool for planning, development, execution, and analysis of functional tests. Record and play back scripts that navigate through your application and test the state of objects through verification points. Robot and Load Test together determine whether a multi-client system is performing within user-defined standards under varying loads. The term performance testing includes Load tests, Stress tests, Configuration tests. Testing effort involves planning and recording *scripts*. Script properties are defined while planning the script with TestManager. Create and edit scripts using the SQA Basic and Virtual User scripting environments. The Robot editor provides color-coded commands with keyword Help for powerful integrated programming during script development. Test applications developed with Integrated Development Environments such as Visual Basic, Oracle Forms, Power Builder, HTML, and Java. Robot is integrated with Rational Purify, Rational Visual Quantify, and Rational Visual Pure Coverage, which let you instrument certain types of applications-under-test. Guidelines before you begin recording:

- Establish predictable start and end states for the scripts
- Setup the Test Environment
- Create modular scripts
- Plan scripts in TestManager

- Enable applications for testing
- Select the IDE Extensions to Load
- Set GUI Record Options
- Select an object order preference
- Setting Robot Window Options
- Changing the Hot Keys in Robot
- Allows low-level recording options
- Allows comments in scripts
- Inserts a Delay Value in a GUI Script

A verification point is a point in a script that is created to confirm the state of an object across builds. During recording the verification point captures object information and stores it as the baseline. During the playback the verification point recaptures the object information and compares it with the baseline. The Types of Verification Points are Alphanumeric

- Clip board
- File Comparison
- File Existence
- Region Image
- Web Site Compare
- Web Site Scan
- Window Existence

A data pool is a test data set. It supplies data values to the variables in a script during script play back. Data pools automatically pump a different set of test data to a script each time a script sends data to the server during the playback of a test. Rational Robot is a powerful automation tool for testing Record scripts for testing, Reuse scripts in part and as a whole and Maintain repositories of test cases and test logs.

ASTRA LOADTEST

It is used for simple web load testing. During the test, Astra LoadTest tracks the response times of Web transactions to identify and pinpoint performance. Astra LoadTest makes testing as easy as using a browser. Real-time monitors and analysis, minimize test cycles. Astra LoadTest's Web server and Web application monitors help you analyze the behavior and performance of your Web site by pinpointing problems such as inadequate bandwidth and poor Web server scalability. Using this information, you can quickly identify ways to optimize the application. Astra LoadTest Key Features Include:

- **ActiveScreen :** This technology provides visual interaction with the screens of your Web application so you can quickly and easily edit virtual user tests after recording.
- **Graphs :** Astra LoadTest provides entire data sets in spreadsheet form, allowing you to create custom analyses.
- **Real-time Monitors and Analysis :** Using real-time monitors, Astra LoadTest provides end-to-end diagnostics that reveal when hardware is really needed and that highlight software optimization opportunities.
- **Reports**
  You can obtain reports containing summary data that presents high-level information, such as transaction by user, transaction performance summary, failed transactions, performance under load and more.
- **ScenarioWizard**
  You can be guided through the process of creating a test scenario, including selecting the workstations to host the tests, the tests to users and groups.
- **Scheduler :**

With Astra LoadTest's Scheduler, you can automate tests at specified times, ramp up Vusers and run tests for a desired period of time.

- **Security :**

Astra LoadTest supports SSL, digital server certificates and many other security features.

The components are:

- Virtual User recorder: Creates test by capturing web traffic generated when navigating a web application.
- Controller : allows hundreds or thousands of virtual users to be directed from a single machine.
- Analysis : contains graphs , spread sheets & reports
- Running Mercury Tours

ASTRA QUICKTEST

It is automated web testing. It is an icon-based tool that allows testers to validate dynamically changing Web applications. It quickly creates interactive maintainable tests by mirroring end-user behavior. Astra Quick Test simplifies and shortens the testing cycle for even the most complex Web environments. Astra QuickTest simplifies and automates the entire Web testing process by making testing as easy as using a browser. Astra QuickTest validates links, objects, images and text on Web pages continue to function properly. Simple tests can be transformed easily into multiple test cases using different data, thereby shortening the application testing cycle. A single test can be used to test your application as it runs in multiple versions of Internet Explorer or Netscape Navigator.

The key features are:

- **Active Screen**

This technology provides visual interaction with the screens of your Web application, so you can quickly and easily edit tests after recording.

- **ActiveX Support**

Astra QuickTest provides record and playback support for every ActiveX control in a Web browser.

- **Checkpoints**

By using checkpoints, you can verify specific information during a test run.

Checkpoints automatically capture and verify properties such as the number of links, HTML content, page load time, number of images, image source, broken links and more—which is very useful for testing static Web pages.

- **Cross-Browser Tests**

Astra QuickTest enables you to record in either Netscape or Internet Explorer and then replay the same test in any browser to verify functionality across multiple browser types.

- **GUI Spy**

Using this tool, you can extract information from browser objects to enable easier test development and debugging.

Astra QuickTest provides support for Java applets in Web pages. It allows firing events and driving methods for a complete replay solution for Java objects. Astra QuickTest Professional provides support for Flash, including record and playback for Frame and Clip animation technology. Both flavors of Astra QuickTest support Real, including record and playback functionality for video and audio.

TEST DIRECTOR

Test Director simplifies and organizes test management by giving a systematic control over the testing process. It helps you create a framework and foundation for your testing workflow. The Test Director administrator must balance the need to protect project data with the need to enable each team member to perform his or her assigned testing tasks. When you create a Test Director project, you need to store and manage the data generated and collected by Test Director. Test Director has four components:

- Requirements Manager Specify testing requirements. This includes defining what you are testing, defining requirement topics and items, and analyzing the requirements.
- Test Plan Manager Develop a test plan. This includes defining goals and strategy, dividing your plan into categories, developing tests, automating tests where beneficial, and analyzing the plan.
- Test Lab Manager Run tests on your application and analyze the results.
- Defects Manager Report defects, determine repair priorities, repair open defects, and analyze the data.

The Quality Assurance manager uses the testing scope to determine the overall testing requirements for the application under test. Requirement topics are recorded in the Requirements Manager by creating a requirements tree. The requirements tree is a graphical representation of your requirements specification, displaying the hierarchical relationship between different requirements. Test plan is an essential requirement for successful software testing. A good test plan enables you to assess the quality of your application at any point in the testing process. It is essential that the tests in your test plan meet your original testing requirements. You create requirements coverage by linking each test in the test plan tree to one or more requirements in the requirements tree. Once you have defined test sets that cover your testing goals, you schedule test execution and assign tasks based on your priorities and strategy. When you run manual tests, you execute the test steps you defined in test planning. You pass or fail each step, depending on whether the application's actual results match the expected output. Following a test run, you analyze test results. Your goal is to identify failed steps and to determine whether a defect has been detected in your application, or if the expected results of your test need to be updated.

Validate test results regularly by viewing run data and by generating TestDirector reports and graphs. Locating and repairing software defects is an essential phase in software development. Defects can be detected and reported by software developers, testers, and end users in all stages of the testing process. Using TestDirector, you can report design flaws in your application, and track data derived from defect reports. When you detect a defect in the application under test, you send a defect report to the TestDirector project. These defect reports are imported into the database by the quality assurance or project manager. TestDirector enables you to create four types of reports are Requirements Coverage Report, Planning Report, Execution Report and Defects Report.

WINRUNNER

WinRunner identifies each GUI object in the application being tested by its physical description: a list of physical properties and their assigned values. In the test script, WinRunner does not use the full physical description for an object. Instead, it assigns a short name to each object: the logical name. Spying on GUI Objects (Gui spy)Help us to understand how WinRunner identifies GUI objects.

By recording, we can quickly create automated test scripts, clicking objects with the mouse , entering keyboard input. It generates statements in TSL, Mercury Interactive's Test Script Language. Context Sensitive mode records the operations you perform in terms of the GUI

objects, WinRunner identifies each object u click and type. A GUI Checkpoint examines the behavior of an object's properties. If the application contains the bitmap areas (drawings, graphs) we can check these areas using the bitmap checkpoint. You can read text from any bitmap image or GUI object by adding text checkpoints to a text script. A text checkpoint reads the text from the application. You can use text checkpoints in your test scripts to read and check text in GUI objects and in areas of the screen. While creating a test you point to an object or a window containing text. You can read the entire text contents of any GUI object or window in your application, or the text in a specified area of the screen.

Create database checkpoints, you define a query on your database and your database checkpoint checks the values contained in the result set. The result set is set of values retrieved from the results of the query. Create standard database checkpoints to compare the current values of the properties of the result set during the test run to the expected values captured during recording. Two types of standard database checkpoints: Default and Custom. You can create dialog boxes that pop up during interactive test execution. Prompting the user to perform an action— such as typing in text or selecting an item from a list.

CONCLUSION

The various automated software testing tools and their working procedures are explained. Based on the experience and knowledge lever of the testers, the appropriate testing tool will be selected. However there are lot of testing tools available in the market, the selection of testing tool will be based on the nature of the project. There will be separate testing tools for the different purposes as that has been discussed.

REFERENCES

1. ManjitKaur, "Comparative Study of Automated Testing Tools", International Journal of Computer Applications (0975 – 8887), Volume 24– No.1, June 2011.

2. Ilene Burnstein, "Practical Software Testing", Springer International Edition, 2010.

3. Mesbah, A.; van Deursen, A.; Roest, D., "Invariant- Based Automatic Testing of Modern Web Applications", IEEE Transactions on Software Engineering, Volume: 38 , Issue: 1 , Page(s): 35 – 53, 2012.

4. Anisha Kumar, "Approaches of the WinRunner and LoadRunner Software Testing Tools: A Case Study", The IUP Journal of Computer Sciences, October, 2009.

5. Roper, R.M.F. ; Smith, P., "A software tool for testing JSP designed programs", Software Engineering Journal , Volume: 2 , Issue: 2, Page(s): 46 – 52, 1987.

6. Gang Luo ; Probert, R.L. ; Ural, H., "Approach to constructing software unit testing tools", Software Engineering Journal , Volume: 10 , Issue: 6, Page(s): 245 – 252, 1995.

7. Richa Rattan, "Performance Evaluation & Comparison of Software Testing Tool", International Journal of Information and Computation Technology, Volume 3, Number 7(2013), pp. 711-716.

8. Muhammad Shahid1, "A Study on Test Coverage in Software Testing", International Conference on Telecommunication Technology and Applications, vol.5 (2011).

9. Harpreet Kaur, "Comparative Study of Automated Testing Tools: Selenium, Quick Test Professional and Testcomplete", Int. Journal of Engineering Research and Applications, Vol. 3, Issue 5, Sep-Oct 2013, pp.1739-1743.

10. Cullyer, W.J. ; Storey, N., "Tools and techniques for the testing of safety-critical software", Computing & Control Engineering Journal, Volume: 5 , Issue: 5, Page(s): 239 – 244, 1994.