UNIT –IV
SOFTWARE TESTING TOOLS

(1) Introduction to Testing-Automated Testing:

- Automated Testing means using an automation tool executing the test case suite. Manual Testing is performed by a human sitting in front of a computer carefully executing the test steps.
- The automation software can also enter test data into the System. Now compare expected and actual results and generate detailed test reports. Test Automation demands considerable investments of money and resources.
- Successive development cycles will require execution of same test suite repeatedly. Using a test automation tool, it's possible to record this test suite and re-play it as required. Once the test suite is automated, no human test is required.
- This improve ROI (Return On Investment) of Test Automation. The goal of Automation is to reduce the number of test cases to be run manually and not to eliminate Manual Testing.

(2)Concepts of Test Automation:

(i)Test Automation

- Automated Software Testing or Test Automation is the process of automating the manual test cases. This also involves comparing the run time data with the test data provided, and producing useful Test Results.

(ii)Test Automation tool – What we can expect

- To test a software, a manual testing engineer needs to do the following basic actions:
  1. Click a button, link, image
  2. Select radio button
  3. Check / Uncheck a checkbox
  4. Type into an Edit box or controls which resemble edit box
- But also he needs to get the run time data from the application and compare with the test data provided (i.e. checkpoints). For this actually, we will be extracting the values from an edit box, button, link, image, table, drop down list, etc.
- All of these basic features are provided by Test Automation tools such as QTP(Quick Test Professional). There are many test automation tools which allow users to record user actions and replay them any number of iterations and compare the actual results with the test data and generate results.

(iii)Why Test Automation?

- Since the very large quantity of software development tools available in the market the productivity of the developer is dramatically increased over the past years.
- To manage the quality of the product developed it is essential to thoroughly test all the features of the application.
- If the application is a large software product having lot of features, then it is very much difficult to test by manually executing the test cases.
- There will be a lot of pressure on testing team to test more and more functionalities in less time. *Hence, the Automation Testing has become an integral part of software development*.

(iv)Benefits of Test Automation:

(a)Execution is Less Time Consuming:

- Automated tools execute the test cases (test script) faster than manual test case execution.
- Hence, in less time, more end-to-end test cases can be covered during test execution.

(b)Un-attended mode execution:

www.Jntufastupdates.com

- There is no need of human involvement for the execution of entire Test Automation Suite for regression testing. But, to achieve the unattended mode execution, the automation framework has to be properly designed.

(c)Repeatable:

- The test suite can be executed multiple times on the application under test.
- For example, if there is a need to test the application on different browsers and environments, we need to just change the configurations of the test automation suite and execute.
- In case of manual execution, we would need one more resource to execute the same set of test cases on different environment / browser.

(d)Reusability:

- The test suite can be built in such a way that the functions or methods written are highly reusable across the framework. Also, the entire test suite built with a proper framework can also be utilized for different versions of the application under test.

(e)Consistency of Test Execution:

- There is a chance of manual tester making errors during execution of test cases. But, the test suite being automated we can expect no or zero errors during execution.
- For example, if there is a need to enter a value in an edit box such as 7693178.87651, a manual tester might make mistakes (as this is a big number with five decimal values) but the automation tool will not make any mistakes. It will enter the same value even if the test is run for many times.

(f)Better Coverage:

- As the time required executing automated test suite will be less compared to manual test case execution, more number of test scenarios can be covered during the execution.
- Hence, we can expect better coverage.

(g)Cost Effective:

- Once the test suite is completely ready for regression testing, the resources required will be less compared to manual test execution. This reduces the cost of testing.

(v)Key factors to be considered in Test Automation:

(a)Test automation is expensive

- It involves testing tools as well as skilled professionals.

***(b)Size of the Regression Test Suite*:**

- Generally, a recommendation is made for test automation if there is a long regression cycle.

(c)Tool compatibility:

- Test Automation tool has to be compatible with the application under test.
- A comprehensive tool evaluation has to be conducted and a best suited tool should be recommended for automation testing.

(d)No. of Regression Cycles:

- If there is a need to execute the automated test suite against many builds / releases then the test automation becomes cost effective and beneficial.

(3)Introduction to list of tools like Win runner, Load Runner & JMeter:

(i)Win Runner:

- Win Runner is the most used Automated Software Testing Tool.
- Main Features of Win Runner are
  - •Developed by Mercury Interactive
  - •Functionality testing tool
  - •Supports o/s and web technologies
  - •To Support .net, xml, SAP, Multimedia etc we can use QTP.
  - •Winrunner run on Windows only.
  - •Xrunner run only UNIX and linux.
  - •Tool developed in C on VC++ environment.

www.Jntufastupdates.com

•To automate our manual test win runner used TSL (Test Script language like c)

The main Testing Process in Win Runner is

1) Learning

❖ Recognization of objects and windows in our application by winrunner is called learning.

2) Recording

❖ Winrunner records over manual business operation in TSL

3) Edit Script

❖ Depends on corresponding manual test, test engineer inserts check points in to that record script.

4) Run Script:

❖ During test script execution, winrunner compare tester given expected values and application actual values and returns results.

5) Analyze Results

❖ Tester analyzes the tool given results to concentrate on defect tracking if required.

(ii)Load Runner:

❖ LoadRunner is a software testing tool from Micro Focus.
❖ It is used to test applications, measuring system behaviour and performance under load.
❖ LoadRunner can simulate thousands of users concurrently using application software, recording and later analyzing the performance of key components of the application.
❖ The components of LoadRunner are The Virtual User Generator, Controller, and the Agent process, LoadRunner Analysis and Monitoring, LoadRunner Books Online.
❖ The Component of LoadRunner would you use to record a Script is the Virtual User Generator (VuGen) component .
❖ LoadRunner copy hundreds or thousands of human users with Virtual Users (Vusers) to apply measurable & repeatable production workloads and stresses an Application end-to-end.
❖ Vusers copy the actions of human users by performing typical Business Processes.
❖ For each user action, the tool submits an input request to server and receives the response.
❖ Increase number of Vusers, to increase the load on the Server.

(iii)Jmeter:

❖ The Apache JMeter is pure Java open source software, which was first developed by Stefano Mazzocchi of the Apache Software Foundation, designed to load test functional behavior and measure performance. You can use JMeter to analyze and measure the performance of web application or a variety of services.
❖ Apache JMeter is a testing tool used for analyzing and measuring the performance of different software services and products. It is a pure Java open source software used for testing Web Application or FTP application. It is used to execute performance testing, load testing and functional testing of web applications.
❖ First JMeter can be integrated into selenium using the plugins, So Selenium you can use as an automation tool for your testing needs. The Apache JMeter is pure Java open source software, designed to load test functional behavior and measure performance.
❖ JMeter is an open source desktop Java application that is designed to load test and measure performance. It can be used to simulate loads of various scenarios and output performance data in several ways, including CSV and XML files, and graphs.
❖ Apache JMeter is one of the most popular open source systems for testing applications, and also a 100% Java scripted desktop application.

(4)About Win Runner:

❖ WinRunner is a test automation tool, designed to help customers save testing time and effort by automating the manual testing process. Automated testing with WinRunner addresses the problems by manual testing, speeding up the testing process

www.Jntufastupdates.com

- WinRunner uses the GUI Map file to recognize objects on the application. When WinRunner runs a test, it uses the GUI map to locate objects. It reads an object's description in the GUI map and then looks for an object with the same properties in the application being tested

(5)Using Win runner:

- After installing the WinRunner on your computer, invoke the WinRunner application:
  Start -> Programs ->WinRunner ->WinRunner
- The opening screen of the WinRunner application is displayed, prompting you to select one of the three options:
  New Test: To create a new test script
  Open Test: To open an existing test script
  Quick Preview: To view the quick preview of WinRunner



(6)Recording Test Cases

- To test any application, first you can run the application and understand its operation. Then, you can invoke WinRunner, again run the application and record the GUI operations.
- During the recording mode, WinRunner will capture all your actions, which button you pressed, where you clicked the mouse etc. You need to work with the application as usual and perform all the actions to be tested.
- Once the recording is completed, WinRunner generates a script in TSL (Test Script Language). You can run this test script generated by WinRunner to view the results. The test results will show whether the test has passed or failed.
- There are two modes of recording:

(i)Context Sensitive mode:

- This mode of recording is used when the location of the GUI controls (i.e. X and Y coordinates) or the mouse positions are not necessary.

www.Jntufastupdates.com

(ii)Analog mode:

- This mode of recording is used when the mouse positions, the location of the controls in the application, also play an important role in testing the application. This mode of recording has to be used to validate bitmaps, testing the signature etc.
- The procedure for recording a test case is as follows:

  Step 1: Open a new document: File -> New (or) Select "New Test" from the WinRunner's Welcome screen.
  Step 2: Open (run) the application to be tested.
  Step 3: Start recording a test case.
  Create ->Record - Context Sensitive (or) click on the toolbar's "Record" button once, to record in Context Sensitive mode.
  Step 4: Select the application to be tested by clicking on the application's title bar.
  Step 5: Perform all the actions to be recorded.
  Step 6: Once all required actions are recorded, stop the recording.
  Create -> Stop (or) Click on the toolbar's "Stop" button to stop the recording WinRunner generates the script for the recoded actions.

Mapping the GUI,

- When WinRunner runs a test, it uses the GUI map to locate objects. It reads an object's description in the GUI map and then looks for an object with the same properties in the application being tested. Each of these objects in the GUI Map file will be having a logical name and a physical description.
- There are two views in the GUI Map Editor, which enable you to display the contents of either:
  (i)the entire GUI map
  (ii)an individual GUI map file

(i)the entire GUI map

- When viewing the contents of specific GUI map files, you can expand the GUI Map Editor to view two GUI map files simultaneously. This enables you to easily copy or move descriptions between files.
- To view the contents of individual GUI map files, choose ***View GUI Files***.

(ii)an individual GUI map file

- In the GUI Map Editor, objects are displayed in a tree under the icon of the window in which they appear. When you double-click a window name or icon in the tree, you can view all the objects it contains. To concurrently view all the objects in the tree, choose ***View Expand Objects Tree***. To viewwindows only, choose ***ViewCollapse Objects Tree***.
- When you view the entire GUI map, you can select the ***Show Physical Description*** check box to display the physical description of any object you select in the ***Windows/Objects*** list.
- When you view the contents of a single GUI map file, the GUI Map Editor automatically displays the physical description.
- Suppose the WordPad window is in your GUI map file. If you select ***Show Physical Description*** and click the WordPad window name or icon in the window list, the following physical description is displayed in the middle pane of the GUI Map Editor:

(7)Working with Test:

- WinRunner facilitates easy test creation by recording how you work on your application. As you point and click GUI (Graphical User Interface) objects in your application, WinRunner generates a test script in the C-like Test Script Language (TSL).
- You can further enhance your test scripts with manual programming.
- User can create tests using recording or programming or by both. While recording, each operation performed by the user generates a statement in the Test Script Language (TSL).

www.Jntufastupdates.com

- These statements are displayed as a test script in a test window. User can then enhance the recorded test script, either by typing in additional TSL functions and programming elements or by using WinRunner's visual programming tool, the Function Generator, or using the Function Viewer.

(8)Checkpoints:

- Checkpoints allow user to compare the current behavior of the application being tested to its behavior in an earlier version. If any mismatches are found, WinRunner captures them as actual results.
- User can add four types of checkpoints to test scripts they are
  - (i)GUI Checkpoints
  - (ii)Bitmap Checkpoints
  - (iii)Text checkpoints
  - (iv)Database checkpoints

(i)GUI checkpoints

- It verifies information about GUI objects. For example, user can check that a button is enabled or see which item is selected in a list. There are three types of GUI Check points they are
  - (a)For Single Property
  - (b)For Object/Window
  - (c)For Multiple Objects

(a)GUI checkpoint for single property:-

- User can check a single property of a GUI object. For example, user can check whether a button is enabled or disabled or whether an item in a list is selected.

(b)GUI checkpoint for object/window:-

- User can create a GUI checkpoint to check a single object in the application being tested.
- User can either check the object with its default properties or user can specify multiple properties to check.

(c)GUI checkpoint for multiple objects:-

- User can create a GUI checkpoint to check multiple objects in the application being tested. User can either check the object with its default properties or user can specify multiple properties of multiple objects to check.

(ii)Bitmap Checkpoint :

- It checks an object, a window, or an area of a screen in the application as a bitmap. While creating a test, user can indicate what user want to check. WinRunner captures the specified bitmap, stores it in the expected results folder (exp) of the test, and inserts a checkpoint in the test script. While running the test, WinRunner compares the bitmap currently displayed in the application being tested with the expected bitmap stored earlier.
- In the event of a mismatch, WinRunner captures the current actual bitmap and generates a difference bitmap. By comparing the three bitmaps (expected, actual, and difference), user can identify the nature of the discrepancy there are two types of bitmap check points they are

(a)Bitmap Checkpoint for Object/Window: -

- User can capture a bitmap of any window or object in the application by pointing to it.

(b)Bitmap Checkpointfor Screen Area:-

- User defines any rectangular area of the screen and captures it as a bitmap for comparison.

(iii)Text checkpoints

- It reads and check text in GUI objects and in areas of the screen. While creating a test user has to point to an object or a window containing text. WinRunner reads the text and writes a TSL statement to the test script.
- Later user can add simple programming elements to test scripts to verify the contents of the text. User should use a text checkpoint on a GUI object only when a GUI checkpoint cannot be used

www.Jntufastupdates.com

to check the text property. There are two types of Text checkpoints they are

From Object/Window

From Screen Area

(iv)Database checkpoints

❖ It checks the contents and the number of rows and columns of a result set, which is based on a query user, create on database. There are three types of database check points they are

(a)Default Check:-

❖ Used to check the entire contents of a result set, Default Check are useful when the expected results can be established before the test run.

(b)Custom Check:-

❖ Used to check the partial contents, the number of rows, and the number of columns of a result set.

(c)Runtime Record Check:-

❖ User can create runtime database record checkpoints in order to compare the values displayed in the application during the test run with the corresponding values in the database.

(9)Test Script Language & Enhancing Test

❖ TSL stands for "Test Scripting Language". The test scripts are written in Test Scripting Language in winrunner. TSL is an enhanced, C-like programming language designed for testing.

❖ The advantages of TSL are:

It is easy to use.

It is similar to other **programming languages**. ...

It is a high level **language**.

❖ When you record a test, a test script is generated in Mercury Interactive's Test Script Language (TSL). Each line WinRunner generates in the test script is a statement.

❖ A statement is any expression that is followed by a semicolon. Each TSL statement in the test script represents keyboard and/or mouse input to the application being tested. A single statement may be longer than one line in the test script.

❖ TSL is a C-like programming language designed for creating test scripts. It combines functions developed specifically for testing with general purpose programming language features such as variables, control-flow statements, arrays, and user-defined functions. You enhance a recorded test script simply by typing programming elements into the test window, If you program a test script by typing directly into the test window, remember to include a semicolon at the end of each statement.

❖ TSL is easy to use because you do not have to compile. You simply record or type in the test script and immediately execute the test.

❖ TSL includes four types of functions:

- ✓ Context Sensitive functions perform specific tasks on GUI objects, such as clicking a button or selecting an item from a list. Function names, such as button_press andlist_select_item, reflect the function's purpose.
- ✓ Analog functions depict mouse clicks, keyboard input, and the exact coordinates traveled by the mouse.
- ✓ Standard functions perform general purpose programming tasks, such as sending messages to a report or performing calculations.
- ✓ Customization functions allow you to adapt WinRunner to your testing environment.

❖ WinRunner includes a visual programming tool which helps you to quickly and easily add TSL functions to your tests. For more information, see "Generating Functions."

❖ This chapter introduces some basic programming concepts and shows you how to use a few simple programming techniques in order to create more powerful tests. For more information, refer to the following documentation:

www.Jntufastupdates.com

(10)Running and Debugging Tests

(i)Create Tests

- Next, you create test scripts by recording, programming, or a combination of both. While recording tests, insert checkpoints where you want to check the response of the application being tested. You can insert checkpoints that check GUI objects, bitmaps, and databases.
- During this process, WinRunner captures data and saves it as expected results—the expected response of the application being tested.

(ii)Debug Tests

- You run tests in Debug mode to make sure they run smoothly. You can set breakpoints, monitor variables, and control how tests are run to identify and isolate defects.
- Test results are saved in the debug folder, which you can discard once you've finished debugging the test.
- When WinRunner runs a test, it checks each script line for basic syntax errors, like incorrect syntax or missing elements in *If*, *While*, *Switch*, and *For* statements.
- You can use the Syntax Check options (*Tools Syntax Check) to check for these types of syntax errors before running your test.*

(iii)Run Tests

- You run tests in Verify mode to test your application. Each time WinRunner encounters a checkpoint in the test script, it compares the current data of the application being tested to the expected data captured earlier.
- If any mismatches are found, WinRunner captures them as actual results.

(11)Analyzing Results:

- After you run a test, you can view the results in the Test Results window. The appearance of this window depends on the Report View option you select in the **Run**category of the General Options dialog box.
- There are two types of Test Results Views:

(i)WinRunner report view

- Displays the test results in a Windows-style viewer.
- If you run a test that includes a call to a QuickTest test, the WinRunner report view displays only basic information about the results of the QuickTest test.
- When running tests that call QuickTest tests, it is recommended to use the Unified report view.

(ii)Unified report view

- Displays the results in an HTML-style viewer.
- The unified TestFusion report viewer is identical to the style used for QuickTest Professional test results.
- If you run a test that includes a call to a QuickTest test (version 6.5 or later), the unified report view enables you to view detailed results of each step in the called QuickTest test.
- Regardless of the selected report view, the test results window always contains a description of the major events that occurred during the test run, such as GUI, bitmap, or database checkpoints, file comparisons, and error messages. It also includes tables and pictures to help you analyze the results.

(12)Batch Tests,

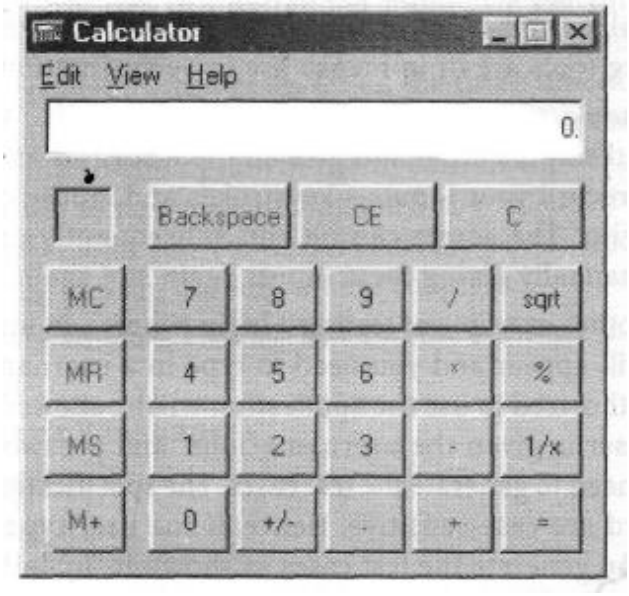
- A batch test is a test script that calls other tests.
- You program a batch test by typing call statements directly into the test window and selecting the Run in batch mode option in the Run category of the General Options dialog box before you execute the test.

www.Jntufastupdates.com

- A batch test is a test script that calls other tests. You program a batch test by typing call statements directly into the test window and selecting the **Run in batch mode** option in the **Run** category of the General Options dialog box before you execute the test.
- A batch test may include programming elements such as loops and decision making statements. Loops enable a batch test to run called tests a specified number of times.
- Decision-making statements such as if/else and switch condition test execution on the results of a test called previously by the same batch script.
- See "Enhancing Your Test Scripts with Programming," for more information.
- You execute a batch test in the same way that you execute a regular test.
- Choose a mode (Verify, Update, or Debug) from the list on the toolbar and choose **Test > Run from Top**. See "Understanding Test Runs," for more information.
- When you run a batch test, WinRunner opens and executes each called test. All messages are suppressed so that the tests are run without interruption.
- If you run the batch test in **Verify** mode, the current test results are compared to the expected test results saved earlier.
- If you are running the batch test in order to update expected results, new expected results are created in the expected results folder for each test.
- See "Storing Batch Test Results" below for more information. When the batch test run is completed, you can view the test results in the Test Results window.
- Note that if your tests contain TSL **texit** statements, WinRunner interprets these statements differently for a batch test run than for a regular test run. During a regular test run, **texit** terminates test execution.
- During a batch test run, **texit** halts execution of the current test only and control is returned to the batch test.

(13)Rapid Test Script Wizard

- The Rapid test script wizard is the fastest way of performing the test process. It systematically opens up all the windows in the application, stores the learnt information in the GUI Map file and generates the test cases based on the information learnt from the application.
- It is possible to apply these tests only on those applications, which open windows upon performing some task (like clicking a button, selecting the menu item etc). Let us apply this wizard on the Calculator application whose GUI is shown in Figure.



www.Jntufastupdates.com