

SOFTWARE TESTING

TRAINING COURSES CONTENTS

Unit I	Software Testing Fundamentals and Best Practices
Course Description	This training course will give basic understanding on software testing, it's types, methods, levels and other related terminologies.
Course Objectives	<p>To learn about software testing fundamentals – concepts, roles and terminology. The course also covers the different types of testing performed at each phase of the software lifecycle and the issues involved in these types of tests, clarifies testing terminology, and much more.</p> <p>Understand testing automation vendors serving the industry and methods in evaluating an organization's needs and the vendor best suited to meet their needs</p>
Duration	30 Hours
Course Contents	<p>Concepts and Terminology</p> <p>Purpose of Software Testing</p> <p>Roles and Responsibilities of Software Test Engineer</p> <p>Software Testing defined</p> <p>Test design methodology</p> <p>Test cases</p> <p>Test scripts</p> <p>Strategies for Software Testing</p> <p>Testing levels</p> <p>Unit testing</p> <p>Integration testing</p> <p>Smoke Testing</p> <p>Sanity Testing</p> <p>System testing</p> <p>Debugging</p> <p>Bug Life cycle In software development process</p> <p>Black box/Glass box/Gray box testing</p> <p>Black Box vs Grey Box vs White box testing</p> <p>Requirements driven Scenario - based development and testing</p> <p>Best Software Testing Practices</p>

	<p>Testing without requirements or with poor requirements Coping with fast pace incremental and highly iterative development process Quality requirements: the unspoken of and the untestable Pre-designed tests Vs ad hoc testing Systematic Negative Testing (no guessing) Getting the most out of your Regression Testing Static Testing Software Testing Automation Introduction General Test Automation strategies Categories of Test Automation Tools Functional Test Automation methodology Functional Test Strategy Managing Test Data Evaluating Functional Test Automation tools Developing Automation Frameworks DataDriven Frameworks Keyword Driven Frameworks Comparision of functional automation techniques Key functional test automation tools Model Based approach to Test Automation.</p>
Unit II	HP Application Lifecycle Management 11.5 - Quality Center 11.5
Course Description	This course provides the tools needed to implement and utilize Quality Center 11.5. Trainees learn how to manage quality information throughout the development cycle, from constructing requirements, designing and executing tests, through monitoring defects.
Course Objectves	<p>After completing this course, trainee should be able to:</p> <p>Describe the new features of ALM 11.5 Describe Application Lifecycle Management Create releases and cycles.</p>

	<p>Construct requirements.</p> <p>Analyze risks associated with requirements.</p> <p>Organize subjects and tests in a test plan tree.</p> <p>Design and create test plans.</p> <p>Generate test scripts from design steps.</p> <p>Create test sets.</p> <p>Execute manual and automated tests.</p> <p>Use Hp Sprinter on manual tests.</p> <p>Record and track test execution results.</p> <p>Log and manage defects.</p>
Duration	30 Hours
Course	Introduction
Contents	<p>Application Life cycle Management Road map</p> <p>ALM – Management</p> <p>Releases and Cycles</p> <p>Project Planning and Tracking</p> <p>Libraries and Baselines</p> <p>Imported Libraries</p> <p>ALM – Requirements</p> <p>Introducing Requirements</p> <p>Requirements Specification</p> <p>Requirements Traceability</p> <p>Traceability Matrix</p> <p>Risk-Based Quality Management</p> <p>Business Process Modles</p> <p>Test Planning</p> <p>Introduction</p> <p>Objectives</p> <p>Building a test plan tree</p> <p>Convert Requirements to Tests</p> <p>Defining Tests</p> <p>Create a Test Plan Tree Manually</p>

Add and Edit a Design Step
Create a Test Configuration
Call a Test
Using parameters in tests
Generate Automated Test Script
Define Test Resources
Link Tests to a Requirement
Designing test steps
Generate a live analysis graph
Generate a Live Analysis Graph
Summary
Review Questions
Exercise: Building a Test

Lab Management

Introduction
Objectives
Overview of Lab Management
Lab Management Concepts
Managing Lab Resources
Test Scheduling and Execution
Continuous Deployment with ALM
Summary
Review Questions
Exercise: Coverage and Execution Analysis

Test Execution

Introduction
Objectives
Create a Test Set Folder
Assign Test Set Folders to Cycles
Create a Test Set
Define Test Execution Flow

Test Conditions and Dependencies
Run Tests Manually
Set Test Notifications and Failure Rules
Running Test Automatically
Summary
Review Questions
Exercise: Building and Executing a Test Set
HP Sprinter
Introduction
Objectives
Overview of HP Sprinter
Sprinter Interface
Run a Basic Test
Use Subtitles
Make Annotations
Submit Defects
Power Mode
Authoring Tests
Scanner - Capabilities
Exploratory Testing
Story Board & Reporting
Summary
Review Questions
Exercise: Building Manual Testing with HP Sprinter
Defect Tracking
Introduction
Objectives
Using the Defects Module
Defect-Requirements Relationship
Defect-Test Relationship
Update a Defect

	<p>Summary</p> <p>Review Questions</p> <p>Exercise: Logging Defects</p>
Unit III	HP – Unified Functional Testing 11.5
Course Description	<p>This course provides a comprehensive understanding of how to use the Unified Functional Testing (UFT) 11.5 application as an automated functional testing tool. Trainee use the point and click interface to record and play back tests, add synchronization points and verification steps, and create multiple action tests. The hands-on lab exercises use version 11.5 of the Unified Functional Testing application.</p>
Course Objectives	<p>At the end of the course, trainee should be able to:</p> <ul style="list-style-type: none"> Create basic scripts from a manual test case Enhance basic tests with synchronization and verification Parameterize tests to run with multiple sets of data. Create and reuse modular actions Use the Object Repository Use debugging tools Write custom checkpoints to create more precise verification points within a test Use the Object Repository Manager Resolve object recognition problems Use Local System Monitoring Identify when Expert View is useful. Retrieve and use the properties of an object. Create programmatic descriptions. Create tests that include VBScript. Retrieve data from application objects. Use the Data Table object to store run-time data and drive actions. Create scripts that access data from external sources. Create new subroutines and functions.

	<p>Use the Function Library editor.</p> <p>Identify when to handle exceptions programmatically</p>
Duration	45 Hours
Course	Introduction
Contents	<p>Describe Unified Functional Testing (UFT) GUI for Testing</p> <p>Describe the GUI testing workflow</p> <p>Identify additional GUI testing capabilities and tools</p> <p>Recognize a sample application under test (AUT)</p> <p>Identify UFT for GUI Testing resources</p> <p>Preparing to Record</p> <p>Identify functional testing principles</p> <p>Document the user steps of a business transaction</p> <p>Identify the application under test and its environment</p> <p>Create a new test</p> <p>Navigate the Unified Functional Testing (UFT) for GUI Testing user interface</p> <p>Prepare the test environment to effectively use UFT for GUI Testing</p> <p>Creating a Basic Test</p> <p>Create a basic test from a manual test case</p> <p>Run a test and check for errors</p> <p>Save a test</p> <p>View test results</p> <p>Working With Objects</p> <p>Identify objects</p> <p>Define a UFT for GUI Testing object</p> <p>Identify objects in UFT for GUI Testing</p> <p>Use the object repository to manage objects in UFT for GUI Testing</p> <p>Adding Synchronization</p> <p>Define synchronization in UFT for GUI Testing</p> <p>Identify the uses of synchronization in UFT for GUI Testing</p> <p>Add a synchronization step for a specified object</p> <p>Verifying with Standard Checkpoints</p>

Define standard checkpoints
Add standard checkpoints to a test
Use a regular expression to add flexibility to a standard checkpoint

Using Parameters

Identify and use different parameter types
Insert an input parameter
Insert an output parameter
Parameterize a checkpoint
Evaluate test results for iterative tests

Automatically parameterize steps

Automatic Parameterization

Access options to automatically parameterize steps
Identify automatic global parameterization
Identify global data table parameters
Use test parameters
Evaluate test results of automatic parameterization
Create built-in and user-defined environment variables

Building Multiple, Reusable Actions

Identify actions in GUI testing
Identify action types
Identify action and test iterations
Identify calls to existing actions and copies of actions
Share values using the global data table
Call actions with parameters
Store action return values
Create multiple actions from a single action
Create a new action
Call a reusable action from another test
Use local and global data sheets
Resolve missing actions

Adding Steps without Recording

<p>List the types of steps that can be added to a test without using the record feature</p> <p>Use conditional statements in a test</p> <p>Use the Step Generator</p> <p>Use the reporter object to report events in the test results</p> <p>Creating Tests on a Web Application</p> <p>Record and run a test on a web application</p> <p>Insert a text checkpoint in a test for a web application</p> <p>Writing Custom Checkpoints</p> <p>Create a custom checkpoint</p> <p>Compare captured parameter values with expected values</p> <p>Use debug tools to investigate the causes of failure</p> <p>Verify that a reported error message reflects the state of the test</p> <p>Using Database Checkpoints</p> <p>Identify the purpose of a database checkpoint</p> <p> Create a Structured Query Language (SQL) statement using Microsoft Query</p> <p>Create a database checkpoint</p> <p>Parameterize a database query</p> <p>Significant Checkpoints for GUI Testing</p> <p>Table checkpoints</p> <p>Text and text area checkpoints</p> <p>Text recognition for Windows-based objects</p> <p>File content checkpoints</p> <p>XML checkpoint</p> <p>Utilizing a Shared Object Repository</p> <p>Identify the types of object repositories</p> <p>Manage shared object repositories using the Object Repository Manager</p> <p>Use visual relation identifiers</p> <p>Object Identification Techniques</p>
--

Configure object identification
Describe mandatory and assistive properties
Use ordinal identifiers
Use smart identifiers
Describe when to use Smart Identification
Use the Smart Identification process
Describe how UFT for GUI Testing uses Smart Identification – Use Case Scenario
Test object mapping for unidentified or custom classes
Using Recovery Scenarios
Identify exceptions in a test
Create a recovery scenario
Associate a recovery scenario with a test
Set an optional step in a test
Local System Monitoring
Use the Local System Monitoring feature
View performance graphs
Export the system monitor tab result
Introduction to Expert View
Identify the advantages of Expert View
Translate steps between Keyword view and Expert View
Explain how VBScript and objects are relevant to UFT for GUI Testing
List common test objects and methods used in UFT for GUI Testing
Using Expert View
Use the step generator to create new steps in the test
Enter steps manually in the Expert View
Use the VBScript With statement
Trace and debug tests
Use statement completion (IntelliSense)
Working With Object Properties
Retrieve the properties of an object during a test run

Retrieve and set identification property values
Access native properties and operations
Use constants and variables in scripts
Use the reporter utility object to report events to the test results

Working with Dynamic Objects

Set dynamic values of objects in the object repository
Build a programmatic description for an object
Create and use a description object
Identify additional uses of programmatic descriptions

Using VBScript

Describe commonly used VBScript operators
Describe commonly used VBScript statements
Describe commonly used VBScript functions
Use VBScript operators, statements, and functions in a script

Working with Dynamic Data

Retrieve data from application objects
Describe the VBScript looping statements
Use the Data Table object to store run-time data and drive actions
Build a script for a data-driven test case

Retrieving External Data

Build scripts that access data from external sources
Import data from and export data to a Microsoft Excel worksheet
Use the Connection and RecordSet objects to query a database
Import and export data to text files

Creating New Procedures

Identify the advantages of creating a procedure in a test
Create new subroutines and functions
Register a procedure with an object class
Build a function library and associate it with a test

Handling GUI Testing Exceptions

Describe exceptions in a test

	<p>Handle positive and negative test data</p> <p>Use recovery scenarios for exceptions in a script</p>
--	--

Unit IV	HP Loadrunner 11.0
Course Description	<p>This course introduces students to LoadRunner</p> <p>The course covers topics for the VuGen, the Controller, and Analysis tools. It also includes a hands-on lab. LoadRunner is an automated load testing tool that allows to test the applications before, during, and after deployment. This course is designed to give a firm foundation in basic load testing tasks.</p>
Course Objectives	<p>At the end of the course, trainee should be able to:</p> <ul style="list-style-type: none"> Identify the components of LoadRunner Record scripts in the web environment using VuGen Measure steps and business processes using transactions Parameterize scripts to vary user input data Customize scripts by adding VuGen and basic C functions Correlate scripts to process server-generated data Correlate the script using new correlation studio that hosts all correlation functionality in one place: record-based correlations, replay-based correlations, correlation rules and correlation from snapshot Organize and manipulate multiple VuGen scripts packaged under a Solution Explorer, such as Action, Parameters, Run-time settings, Script Replay, and External Files. Describe the features of the new custom VuGen layout Apply the workflow recommended for creating a basic scenario Assign scripts, run-time settings, performance monitors, load generators, and users to a scenario based on load testing goals Load test application by executing a scenario Use the Analysis tool to view the results of a scenario
Duration	30 Hours
Course Contents	Introducing VuGen Define VuGen

Identify the main components of the VuGen interface
Experience the new look and feel of VuGen UI

Recording for the Web

Describe the main VuGen UI
Use the Protocol Advisor
Create VuGen scripts by recording user steps
Identify the differences between the HTML and URL recording levels
Save scripts
Create Business Process Reports

Replaying Scripts

Identify and configure the appropriate web run- time setting for replay
Replay the script in VuGen to verify script functionality
Recognize the debugging tools available in VuGen

Inserting Transactions

Explain the function of a transaction in a script
Insert a transaction in a script during and after recording

Parameterizing a Script

Explain what parameters are and how they work
Solve playback problems with parameterization
Parameterize a script for load testing

Verifying Scripts

Recognize why and when to use verification
Identify visual cues to check for during load testing
Add Text Checkpoints during and after recording

Creating Actions

Create Actions for a web script
Configure Actions to achieve load testing goals

Using the Editor

Work with the editor
Send customized output messages to the Replay Log

	<p>Identify basic C code including statements, variables, and functions</p> <p>Apply basic debugging techniques in VuGen</p> <p>Advanced Scripting Techniques</p> <p>Recognize general LoadRunner functions</p> <p>Recognize protocol specific functions</p> <p>Auto Correlation After Recording</p> <p>Work with Correlation Studio</p> <p>Correlate dynamic values found by using the Auto Correlation tool</p> <p>Manual Correlation</p> <p>Manually insert the web_reg_save_param_ex correlation function</p> <p>Parameterize the dynamic value in a script</p> <p>Correlate a script manually by using Wdiff</p> <p>Auto Correlation During Recording</p> <p>Create correlation rules to auto correlate during recording</p> <p>Import and export correlation rules</p> <p>LoadRunner Hands-On Lab</p> <p>Identify LoadRunner’s load testing components and process</p> <p>Identify the scenario execution process</p> <p>Create a baseline script with multiple action sections</p> <p>Create transactions in a script</p> <p>Create checkpoints in a script</p> <p>Define lab and script requirements</p> <p>Define steps</p> <p>Calculate the load test</p> <p>Execute scenarios</p>
--	---

Unit V	Selenium
Course Description	This training course on selenium provide a thorough coverage on selenium as a web application testing framework. Selenium is a portable software testing framework for web applications. Selenium provides a record/playback tool for authoring tests without learning

	<p>a test scripting language (Selenium IDE). It also provides a test domain-specific language (Selenese) to write tests in a number of popular programming languages, including Java, C#, Groovy, Perl, PHP, Python and Ruby. The tests can then be run against most modern web browsers. Selenium deploys on Windows, Linux, and Macintosh platforms.</p>
Course Objectives	<p>Understand the following topics in great detail and also shall be able to perform the test automation using selenium web applications</p> <ul style="list-style-type: none"> Test Automation for Web Applications To Automate or Not to Automate? Introducing Selenium Brief History of The Selenium Project Selenium’s Tool Suite Understand Architecture of selenium Choosing Selenium Tool Learn about Supported Browsers and Platforms Understand the features Flexibility and Extensibility Learn about Webdriver and Selenium server . Introduction to selenium RC Learn about Selenese
Duration	30 Hours
Course Contents	<p>Selenium Overview</p> <ul style="list-style-type: none"> What is Selenium? Why Automated Testing? History Architecture Selenium in Action Calling WebDriver Directly Summary <p>Selenium IDE</p> <ul style="list-style-type: none"> Introduction Installing

Selenium IDE Basics
Recording a Test
Running the Test
Selenese
Looking at Commands
Creating Commands
Verifications
Using Assert and Verify
Creating Test Suites
Exporting
Other Features
Summary
Webdriver Basics
Introduction
What is WebDriver?
Setting up WebDriver
Other Browsers
Finding Elements
A Simple Script
Page Elements
A More Complex Script
Clicking the Image
Assert and Verify
Summary
Selenium Remote Control (RC)
Installation and configuration
Running HTML test suites against various browsers
Creating batch files
Cross browser testing
Test scripting with Selenium RC
Selenium Server

	<p>Introduction</p> <p>what is Selenium Server?</p> <p>What is Selenium Grid?</p> <p>Starting Selenium Server</p> <p>Running a Test</p> <p>Grid Configuration</p> <p>Setting up Grid</p> <p>A 2nd Node</p> <p>3rd Node and Configuration</p> <p>Running a Test in Grid</p> <p>More Grid Considerations</p> <p>Grid Strategies</p> <p>Summary</p>
--	--

Unit VI	
Silk Test Product Suite	
Course Description	<p>This training course on Silk Test Product Suite includes the following components:</p> <p>Silk Test Workbench – Silk Test Workbench is the native quality testing environment that offers .NET scripting for power users and easy to use visual tests to make testing more accessible to a broader audience.</p> <p>Silk4NET – The Silk4NET Visual Studio plug-in enables to create Visual Basic or C# test scripts directly in Visual Studio.</p> <p>Silk4J – The Silk4J Eclipse plug-in enables to create Java-based test scripts directly in Eclipse environment.</p> <p>.</p>
Course Objectives	<p>Silk Test is an easy to use tool for automated functional testing, which includes a set of interfaces (clients) that enable to perform automated functional testing in a variety of development environments and programming languages.</p> <p>This course focuses on the Workbench interface, and is designed to</p>

	<p>teach how to use the visual testing component of Silk Test.</p> <p>Upon completion of this module, trainee will:</p> <ul style="list-style-type: none"> • Be able to create a new visual test. • Know how to edit the visual test to add validation and logic. • Be able to create data-driven visual tests. • Know how to playback automated tests, and understand the results. <p>Silk4J – use of Java for Test automation</p> <p>Silk4Net – use of .NET (C# and VB.Net) for test automation</p>
<p>Duration</p>	<p>45 Hours</p>
<p>Course Contents</p>	<p>Silk Test Workbench</p> <ul style="list-style-type: none"> • About this Course • Introduction to Silk Test Workbench • Tour of the Silk Test Workbench User Interface • Projects and Users • Assets • Recording a Visual Test • Object Recognition • Decision and Repetition Logic • Verification Logic • Using Data in Visual Tests • Debugging Visual Tests and Error Handling Logic • Playback and Analyzing Results • Introduction to .NET scripts • Database Maintenance • Putting It All Together

Silk4J, the Silk Test plug-in for Eclipse, and is designed to teach how to use Java to create automated tests.

The following will be covered in this Module:

- Introduction to Silk4J
- Silk4J Projects
- Configuring applications
- Recording a test
- Object recognition
- Adding verifications
- Playing back tests and analyzing results
- Example: Cross-browser testing
- Putting it all together

This course focuses on Silk4NET, the Silk Test plug-in for Visual Studio, and is designed to teach how to use C# and VB.NET to create automated tests.

The following will be covered in this module:

- Introduction to Silk4NET
- Configuring applications
- Silk4NET Projects
- Recording a test
- Object Recognition
- Adding logic to a test
- Verifications
- Playing back tests and analyzing results
- Enhancing tests
- Example: Cross-browser testing
- Setting options
- Putting It All Together

--	--