

Selenium 2 – Selenium Testing Process

Phase in Selenium Testing Process

- i) Test Planning
 - ii) Generate Basic Tests/Test Cases
 - iii) Enhance Test Cases
 - iv) Run & Debug Test Cases
 - v) Analyze Test Result and Report Defects
-

Before going to learn Selenium Test Process, first know about phases of Software Test Process / Software Test Life Cycle

- i) Requirements Analysis
- ii) Test Planning
- iii) Test Design
- iv) Test Environment Setup
- v) Test Execution
- vi) Test Closure

i) Requirements Analysis:

- Risk Analysis
- Segregate Test Requirements
- Create RTM Document

ii) Test Planning:

- Test Strategy Implementation
- Test Estimations (Scope, resources, time and budget)
- Team Formation
- Test Plan Documentation

Configuration Management planning

Define Test Environment Setup

Etc....

iii) Test Design

Understanding Test Requirements

Derive Test scenarios

Document Test Cases

Collect Test data etc...

iv) Test Environment Setup

Prepare Test Environment Checklist and Test Data

Test Environment Setup

Verify Test Environment

Conduct Smoke Tests

v) Test Execution:

Comprehensive Testing

Reporting Defects

Tracking Defects

Select Test cases for Re & Regression Testing cycle 1

Sanity Testing

Execute Re & Regression Test cases

Reporting defects

Tracking defects

Select Test cases for Re & Regression Testing cycle 2

Sanity Testing

Execute Re & Regression Test cases

Reporting defects

Tracking defects

Final Regression

vi) Test Closure:

- Evaluate the exit Criteria
 - Collect all artifacts
 - Prepare Test Summary Report
 - Send Test deliverables to customer
 - Improvement suggestion for Future Projects
-

i) Test Planning in Selenium Test Process

- > Get Environment (UI Design and Database) details from Development team
- > Analyze the AUT (Application Under Test) in terms of Object identification/Element identification

Selenium IDE: Use Record and playback some navigations and analyze

Selenium WebDriver: Using Element inspectors analyze the AUT,

Mozilla Firefox - Built-in feature "Page inspector"

Google Chrome - Built-in Developer tools (F12)

MS Edge - Built-in Developer tools (F12)

- > Select Test Cases for Automation

Test cases that can be executed on Initial build and every modified build (Smoke, Sanity)

G C Reddy Technologies (www.gcreddy.com)

Test Cases that can be executed on every modified build (Re & Regression Tests)

Test Cases that can be executed using multiple sets of Test data (Data Driven Tests)

> Select Selenium Tools and Others for Automated Testing and Configure (Eclipse IDE, Java, Selenium WebDriver, TestNG, Maven, Jenkins etc...)

> Download Eclipse IDE and Extract -Editor, Syntax Guidance, Context Help and Auto Compilation...

> Download Java (JDK) Software and Install in your computer

> Set Java Environment Variable path in the OS Environment

> Download selenium WebDriver java language binding from www.seleniumhq.org and add Selenium WebDriver jar files to Java Project in Eclipse IDE

> Download & Install TestNG Testing framework in Eclipse IDE

Download browser driver/s (Firefox, Chrome, MS Edge...) and Instantiate in Selenium Test Cases

ii) Generate Basic Tests/Test Cases

In UFT:

1) Object Repository based Test Design (Recording, Keyword Driven Methodology)

Or

2) Descriptive Programming (Static Programming and Dynamic Programming)

In Selenium:

a) Selenium IDE: Using Recording feature or Type Test steps using Element Locators and Selenese commands

b) Selenium WebDriver: Using Element Locators and WebDriver API Commands

iii) Enhance Test Cases

1) Inserting Verification points

In UFT:

Using Check points (UFT Tool feature)

Using VBScript conditional statements

In Selenium IDE: Using "verify" or "assert" commands

In Selenium WebDriver:

Using WebDriver Verification Commands (limited use)

Using Programming Control Flow Statements

Using TestNG Testing Framework "assert" methods

2) Parameterization

- > Replacing constant (fixed) values with parameters is called Parameterization
- > Passing single value or multiple values....

We use Parameterization feature for Data Driven Testing....

In UFT:

- i) Using Data Table (UFT tool feature)
- ii) Using VBScript Automation Object Models (FileSystemObject, Excel Object....)

In Selenium IDE: No Support

In Selenium WebDriver: Using programming (Flat files/excel files)....

3) Synchronization

In UFT: Using Synchronization Point, or wait command or increase Tool default time

In Selenium IDE: Using Synchronization Timeout feature (Tool feature)

In Selenium WebDriver: Using programming wait feature or WebDriver command

4) Error Handling

Handling expected and unexpected Errors

In UFT: Using Recovery Scenarios (tool feature or VBScript features)

In Selenium WebDriver: Using programming features (Ex: exception handling in Java)

5) Adding Comments

To make the code readable and to disable the code from execution

In UFT: We can use VBScript Comment syntax

In Selenium IDE: Using Selenium IDE comments syntax

In Selenium WebDriver: Using Programming Syntax (if we use Java then use Java Comment syntax)

iv) Run & Debug Test Cases

Run > Debug > Run...

Run Test Cases is a mandatory task in Testing, but Debugging Test cases is an optional task in Automated testing

Run Test Scripts / Test Cases

- > Run a Single Test case
- > Batch Testing

In UFT:

Use "Test Batch Runner" tool or using AOM Script or ALM/QC etc...

In Selenium IDE: using Test Suite feature

In Selenium WebDriver: Using Programming and using Testing Framework

Debug Test Cases

Note: Debugging feature is NA for Manual Testing and It is only for Automated Testing

What is Debugging?

Locating and isolating errors through step by step execution

Application Life Cycle

Development Testing Production

Error	Fault	Failure
Mistake	Defect	
	Bug	

When debugging is required?

Scenario 1: Test Case is not showing any error and providing desired result - Not required

Scenario 2: Test Case is showing errors - Optional

Scenario 3: Test Case is not showing any error and Not providing desired result - Required

Whenever Test case is not showing any error and not providing desired result there debugging is required.

v) Analyze Test Result and Report Defects

a) Analyze Test Result:

Selenium WebDriver doesn't have built-in Result Report facility, using Programming control Flow statements or Testing Framework verification methods we can generate Test Results

Status of Test Results in Software Testing

- 1) Pass (if expected == actual)
- 2) Fail (if expected != actual)
- 3) Warning (Whenever Test Case is not executing properly)
- 4) Done (if there is no verification point in the Test case)

b) Reporting Defects:

After analyzing the test Results, if we find any deviation from expected then report defects....

Functional Test Automation

Defect Management

Selenium

Excel

Selenium

Bugzilla or Jira...
