TEST

Centricity 2 Product Development Testing Overview

# Introduction

The purpose of this document is to provide a brief overview of the product development testing methodologies.

With agile development methodologies, quality assurance and testing resources are integrated into the feature and defect development workflow. Quality assurance resources work closely with engineering to test and validate functionality. This close integration has several benefits:

* Creates a very effective and quick feedback cycle. Engineers are able to quickly gain feedback and calibrate while the feature or defect has focus. This is much different from the waterfall method where ALL testing is done months or years after the feature is developed.
* Information exchange, synchronization and collaboration between engineering and quality assurance resources become more fluid and efficient.
* Less quality assurance and engineering tension, since testing is not placed at the end of the project.
* Higher quality software since functionality stacks are frequently validated and hardened. There is less time making major changes to underlying functionality.
* Socialization and understanding build strong relationships thus creating a team environment.

## Agile Testing Quadrants

The various agile testing methodologies can be grouped into quadrants, as shown in Figure 1. The tests within each of these quadrants will focus on two of four areas:

* Supporting the team – working on the stability of the product
* Technology Facing – the functionality behind the end user interface
* Business Facing – working with the end user interface
* Critique the Product – real life review of the product

Agile Development sprints will include a variety of tests, from different quadrants, which helps to ensure that the test coverage is thorough. Emphasis on each testing quadrant may be different during certain phases of the development process. These phases are discussed in the next section.

Figure : Agile Testing Quadrants

Business Facing

Critique the Product

Technology Facing

Supporting the Team

# Product Development Phases

## Pre-ALPHA Testing

Because of agile development methodology, integrated testing is completed for each sprint or iteration. This ensures that functionality being built meets a certain level of quality. This prevents future major changes to previously built functionality. During this phase several testing methodologies are applied. These include:

Unit Testing
Performed by engineers to test components or processes not available for typical end user testing. Often called “White Box” testing, a testing harness is built to automatically test components or processes. We only implement this methodology on areas with enough complexity to deem so.

### Functional Testing

Using testing outlines, test cases and user stories (user requirements), functional tests are run to guarantee that individual parts of the application perform the appropriate tasks. For example, a test case could be “Ability to successfully remove a user”. User stories can be written in the pattern of: “As a (role), I want (function) so that (business value). A sample user story could be written to state: “As an administrator, I want to be able to easily remove users as part of my database maintenance.”

### Automated Regression Testing

With agile development, continuous regression testing must be performed quickly on already tested functionality. An automated testing tool is used to create, store and run legacy test cases, thus allowing testers to focus on current functionality. All test cases are executed automatically after each successful nightly build. An error report is emailed and shared with engineering and quality assurance each morning.

Boundary Testing
Tests focus on evaluating input validation, browser support and error handling and messaging.

Exploratory Testing
Exploratory testing is an investigative tool. The testing is more organic and unstructured. Use of this method may emulate user workflow without using a prewritten script. Testing in this manner allows the tester to learn more about the application and can be used to help design new test cases.

### Scenario Testing

Scenarios and workflows, that mimic end user behavior, are tested. The system is covered from “end to end” and may integrate multiple functions within the scenario. The data as well as the flow should be realistic. However, the usage paths may be complex. Real-life domain knowledge and client input is valuable when creating these types of tests.

In addition to embedded testing and stabilizing in each sprint, several dedicated sprints are scheduled to concentrate on hardening the existing functionality stacks. Within all testing methods, defects are discussed with architecture and engineering and then added to the Software Configuration Management System (SCM) for resolution.

## ALPHA Testing

Although integrated quality assurance and automated testing occurs during each iteration, the maturity of the product dictates a rigorous stabilization effort. Defect resolution and stabilization takes priority for all development resources within the ALPHA phase. Tests from all four quadrants are more concentrated in this phase, with an emphasis on the following test methodologies:

### Usability Testing

Usability tests are created to evaluate a product by having actual users run a series of tests. Usability tests measure the usability, or ease of use, of an application. It usually involves systematic observation under controlled conditions to determine how well someone can use the software. It is considered a valuable and irreplaceable practice.

### Performance & Stress Testing

Though code and database performance is checked within each iteration, a third-party consultant will simulate large scale interactions to uncover potential stress and performance gaps. As gaps are realized, they will be reviewed and resolutions will be scheduled. Several cycles will occur until an acceptable quality level is obtained.

Security Testing and Audit
Throughout each iteration, great care is placed to ensure commonly known security gaps are closed. To make certain all security gaps are closed, a third-party consultant will perform unbiased audit and test executions. Several cycles will occur until acceptable quality is met.

We begin the ALPHA phase by conducting internal testing with select groups in the company, along with trusted clients and focus groups. The testing is controlled and executed in a Schoolwires testing environment. Because some later functionality will be completed during the beginning of the ALPHA testing phase, certain switches will allow safe testing of current functionality while later functionality is being built. As later functionality becomes ready, it is “turned-on” for testing.

### ALPHA Tester Selection

|  |  |
| --- | --- |
| District Testers | * District diversity is critical when selecting testers.
* We will select Districts based on geography, size and level of adoption.
* Each District will be required to sign an NDA to participate.
 |
| Internal Testers | * We will invite key individuals from Client Support, Training, Creative Services, Client Success and Implementation.
* Other key groups from Marketing will be considered.
* Each tester will be required to sign an NDA to participate.
 |
| Focus Groups | * We will invite individuals from established focus groups to participate.
* Each focus group member will be required to sign an NDA to participate.
 |

##

Each group will be provided with test cases and access to a dedicated ALPHA testing site. The test cases may include functional tests, user stories and scenarios, based on the functionality available in that particular stabilization sprint. The testing site will be updated as each hardened ALPHA release is available. This will likely be after completion of each sprint. Each release will be named with a numeric prefix. For example, Centricity 2 ALPHA1 would denote the first ALPHA release.

### Communication and Feedback

A devoted website will be built for ALPHA testers to submit results, defects and to provide feedback. In addition, key announcements, blog updates and testing materials will be provided through the website.

# BETA Testing

After the ALPHA testing is complete and the migration tools have been built, we begin BETA testing to ensure stability before mass distribution. The testing is still controlled, but testing is performed on actual district data and sites. This phase gives us the opportunity to test the functionality on actual client data along with testing the quality of the migration tools. All previous ALPHA testing methodologies will continue during this phase.

Because some later functionality will need to be completed during the beginning of the BETA testing, certain switches will allow safe testing of current functionality while later functionality is being built. As later functionality becomes ready, it is “turned-on” for testing.

## BETA Tester Selection

In addition to the existing ALPHA testing team, the tester pool will be broadened. Similar selection methods will be used.

Each group with be provided with test cases and a dedicated BETA testing site. The site will be updated as each hardened BETA release is available. This will likely occur after completion of each stabilized sprint. Each BETA release will be named with a numeric prefix. For example, Centricity 2 BETA1 would denote the first BETA release.

## Communication and Feedback

A devoted website will be built for BETA testers to submit results, defects and provide feedback. In addition, key announcements, blog updates and testing materials will be provided through the website.

## BETA Release Steps

Initial BETA releases will occur on internal BETA sites or **copies** of district data and not live websites. Once reasonable quality is met, Release Candidates will be built to testing live websites. Each Release Candidate will be named with a numeric prefix. For example, Centricity 2, RC1 would denote the first Release Candidate.

|  |  |
| --- | --- |
| **Internal BETA Site** | * Initial BETA releases will occur on an Internal BETA Site.
 |
| **Copy of District Site** | * All districts will test a copy of their actual website.
 |
| **Actual District Site** | * Select district will test their actual website.
* A copy of the data will be captured and maintained in case of critical failure.
* Initial Release Candidates will be released.
 |

# Defect & Feature Development Workflow

All product development resources utilize a Software Configuration Management System (SCM) to schedule and manage releases, sprints, defects, features and tasks. The software is web-based and can be available from any location with Internet access. The software has the Agile / SCRUM methodology woven into its architecture making development more effective. For each feature and defect, we have an automated and well-tested workflow to ensure quality and timeliness of each feature and request.

## Feature Workflow

## Defect Workflow

This document is not to be copied or distributed without the express written consent of Iteamic. No part of this document may

be used for purposes other than those intended by Iteamic

**Restricted**

**Submitted by: V & V Team**

#209, 28th Cross, 7th Block, Jayanagar

Bangalore, 560 082

Tel: 91 (80) 2663 4841

www.iteamic.com

This document is created using IPAL V1.0 Selenium-Reference Material V 1.0

Document Version No.: 1.0

Approved By:

Date of Release:

**Selenium- Reference Material**

**Introduction to Selenium**

7/3/2008 Page 2 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

**Table of Contents**

**A. DOCUMENT CHANGE HISTORY ............................................................................... 4**

**1. INTRODUCTION................................................................................................. 5**

**2. SELENIUM IDE................................................................................................... 6**

**3. SELENIUM CORE ................................................................................................ 6**

**4. SELENIUM REMOTE CONTROL................................................................................ 6**

**5. SELENIUM GRID................................................................................................. 6**

**6. WHICH SELENIUM TOOL TO USE............................................................................. 6**

**7. SELENIUM TOOL CONSTRAINTS.............................................................................. 6**

**8. SELENIUM CONFIGURATION WITH INTELLIJ ............................................................... 6**

**9. FURTHER READINGS AND SCOPE ............................................................................ 6**

**Introduction to Selenium**

7/3/2008 Page 3 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

**A. Document Change History**

**Note**: This section is to be maintained by the Project team

**Ver. Author Change Description Reviewed By Approved By Approval Date**

1.0

Tarun

Kumar

Bhadauria

Document Created Ashwith Rai 03-July-2008

**Introduction to Selenium**

7/3/2008 Page 4 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

**1. Introduction**

**Selenium** is a test tool for web applications. Selenium tests run directly in a browser, just like real

users do. Selenium can be used for unit-testing, regression testing, smoke-testing, integration and

acceptance testing of web applications in a variety of browsers and platforms as following –

**Supported Platforms:**

Windows:

o Internet Explorer 6.0 and 7.0

o Firefox 0.8 to 2.0

o Mozilla Suite 1.6+, 1.7+

o Seamonkey 1.0

o Opera 8 & 9

Mac OS X:

o Safari 2.0.4+

o Firefox 0.8 to 2.0

o Camino 1.0a1

o Mozilla Suite 1.6+, 1.7+

o Seamonkey 1.0

Linux:

o Firefox 0.8 to 2.0

o Mozilla Suite 1.6+, 1.7+

o Konqueror

o Opera 8 & 9

There are two modes of operation for Selenium *-* ***Core*** and ***Remote Control*** *(RC).* Remote Control

mode also has a related capability called *Selenium* ***Grid*** that allows throwing hardware at tests to

make it all faster.

**Introduction to Selenium**

7/3/2008 Page 5 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

**2. Selenium IDE**

**Selenium IDE** is an integrated development environment for Selenium tests. It is implemented as a

Firefox extension, and allows to record, edit, and debug tests. Selenium IDE includes the entire

Selenium Core, allowing to easily and quickly record and play back tests in the actual environment that

they will run.

**Features:**

Easy record and playback,

Intelligent field selection will use IDs, names, or Xpath as needed,

Auto complete for all common Selenium commands,

Debug and set breakpoints,

Save tests as HTML, Ruby scripts, or any other format,

Option to automatically assert the title of every page,

**Selenium IDE**

**Introduction to Selenium**

7/3/2008 Page 6 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

**Selenium IDE in Action**

Selenium IDE would work with Browser which in tern interacts with application to simulate user

actions.

**Object Locators used in IDE**

**HTML-ID’s**

Id=LoginButton

**xpath= xpathExpression**

Locate an element using an XPath expression. XPath locators must

begin with "//".

xpath=//img[@alt='The image alt text']

xpath=//table[@id='table1']//tr[4]/td[2]

**link= textPattern**

Select the link (anchor) element which contains text matching the specified

pattern.

link=The link text

**Introduction to Selenium**

7/3/2008 Page 7 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

**3. Selenium Core**

Selenium Core is written in pure JavaScript/DHTML. Selenium Core uses JavaScript and Iframes to

embed a test automation engine in your browser. This technique should work with any JavaScriptenabled

browser.

Selenium Core tests directly into application web server, allowing the tests to run in any supported

browser on the client-side. Thus, one must have write access to server to install Selenium Core.

That means that one can't use Selenium Core (pure DHTML/JavaScript) to write a test of google.com

this is because Selenium Core is pure DHTML/JavaScript, and so it is bound by JavaScript's security

restrictions. This restriction is called same origin policy. The same origin policy states that JavaScript is

allowed only to read/modify HTML from the *same origin* as its source.

Despite the soundness of the policy, it creates a problem for JavaScript automated tests. If one writes

a .js file designed to test google.com, the same origin policy denies the right to run that .js file with

google.com; instead, one has to somehow install that .js file *on google.com* in order to write

automated tests against it.

**Selenium Core**

Selenium IDE embeds Selenium Core internally. Test Runner of Selenium can be driven from IDE itself.

**Introduction to Selenium**

7/3/2008 Page 8 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

**Selenium Test Runner in Action**

Test Runner is always used to run tests coded in HTML format. It is advisable to use this for our trial

and error exercise during our initial test case development. We would have a fair idea on user action

simulation of the recorded script.

**Introduction to Selenium**

7/3/2008 Page 9 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

**4. Selenium Remote Control**

Selenium Remote Control (RC) is a test tool that allows writing automated web application UI tests

in any programming language against any HTTP website using any mainstream JavaScript-enabled

browser.

Selenium RC comes in two parts.

1. A server which can automatically launch and kill supported browsers, and acts as a HTTP proxy

for web requests from those browsers. This Server bundles Selenium Core.

2. Client libraries for your favorite computer language. Using these libraries tests can be coded in

following programming languages - Java, .NET, Perl, PHP, Python or Ruby.

**Selenium RC**

**Introduction to Selenium**

7/3/2008 Page 10 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

Selenium server acts as a Client Configured Proxy for the browser. Selenium Server doesn't

simply fetch the page from the remote server, but instead automatically returns its own

page. That makes the browser think that the remote server served up JS, which allows

selenium to "inject" arbitrary JavaScript in to the domain being tested with out actually

modifying the domain.

**Selenium RC in Action**

Selenium Server is written in Java, and requires the Java Runtime Environment (JRE)

version 1.5.0 or higher in order to start. Selenium RC has two modes of operations –

**Interactive Mode:**

In interactive mode, commands are typed into the Selenium Server command window;

this allows to immediately seeing the results of running command in a working browser.

**Proxy Injection Mode:**

Selenium Tests can not be run against multiple domains using Interactive mode. If tests

are to be run against multiple domains then proxy injection mode should be used.

The two experimental "elevated security privilege" browser launchers allow to test

applications on any web site, including SSL/HTTPS websites, and allow your tests to

freely change domains. These browsers are:

\*iehta: Launches Internet Explorer as an HTML Application (HTA).

\*chrome: Launches Firefox using a chrome URL.

**Introduction to Selenium**

7/3/2008 Page 11 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

The two experimental "proxy injection mode" browser launchers are:

\*piiexplore

\*pifirefox

"Proxy injection" mode is a new highly experimental feature for 0.9.0. In "normal mode"

there are two automated test windows--one for Selenium, and one for application

under test (AUT)--in proxy injection mode we eliminate the separate Selenium window,

in favor of "injecting" Selenium into every HTML page. By injecting ourselves into the

HTML, we have increased control over the AUT, but this comes with some risk, because

we're also modifying the AUT in order to test it.

To use ProxyInjection mode, we need to start the Selenium Server with a special

command line argument, like this:

*Java -jar selenium-server. jar -proxyInjectionMode*

**5. Selenium Grid**

Selenium Grid is an open-source tool that dramatically speeds up web testing by leveraging

existing computing infrastructure. It allows you to run multiple tests in parallel and on

multiple machines, cutting down the time required for running web acceptance tests.

Selenium Grid allows you to run multiple instances of Selenium Remote Control in parallel.

It makes all these Selenium Remote Controls appear as a single one, so your tests do not

have to worry about the actual infrastructure. Selenium Grid cuts down on the time

required to run a Selenium test suite to a fraction of the time that a single instance of

Selenium instance would take to run.

**Introduction to Selenium**

7/3/2008 Page 12 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

**Selenium Grid**

We can choose the number of nodes, the versions of each operating system. Mac OS Tiger

and Leopard can be nodes in the same farm. Similarly Windows XP, Vista and Server 2003

can be nodes in the same farm.

**Introduction to Selenium**

7/3/2008 Page 13 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

**6. Which Selenium Tool to Use**

\* = experimental support is available in Selenium RC

\*\* = theoretically possible, but not currently implemented

Note that even though Selenium RC requires Java, one can write RC tests in .NET, Perl, Python,

and Ruby as well, but one still needs Java around to run the proxy.

Not requiring remote installation and Language Support make Selenium RC most viable

Selenium Tool.

Since Selenium RC provides proxy server hence it needs not be installed on the server

on which application is deployed.

Selenium Remote Control allows to write tests in any programming language, including

Java, .NET, Perl, Python and Ruby. Selenese has a number of strict limitations: it has

no conditionals (no "if" statements), and it has no loops (no "for" statements). This can

make writing complicated tests difficult

**7. Selenium tool constraints.**

This tool is used only of web applications.

It does not have inbuilt reporting functionality.

As it is an open source tool, this does not has commercial support.

Cross domain testing browsers are experimental as of now.

**Introduction to Selenium**

7/3/2008 Page 14 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

**8. Selenium Configuration with IntelliJ**

Download Selenium RC from OpenQA site

Start any java IDE.

Create new project.

Add to your project classpath selenium-java-client-driver.jar

Record your test to from Selenium IDE and translate it to java code (Selenium IDE has

automation translation feature).

Run selenium server from console (You need initialized java environment variable to do

this) like: java -jar selenium-server -proxyInjectionMode.

Run your test in IDE.

These points have been delineated below with reference to IntelliJ IDEA:

**Open a New Project in IntelliJ IDEA -**

**Introduction to Selenium**

7/3/2008 Page 15 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

Give name and location to Project -

Click Next and provide compiler output path –

**Introduction to Selenium**

7/3/2008 Page 16 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

Click Next and select the JDK to be used –

Click Next and select Single Module Project –

**Introduction to Selenium**

7/3/2008 Page 17 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

Click Next and select Java module –

Click Next and provide Module name and Module content root –

**Introduction to Selenium**

7/3/2008 Page 18 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

Click Next and select Source directory –

At last click Finish. This will launch the Project Pan.

**Introduction to Selenium**

7/3/2008 Page 19 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

**Adding Libraries to Project:**

Click on Settings button in the Project Tool bar –

Click on Project Structure in Settings pan –

**Introduction to Selenium**

7/3/2008 Page 20 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

Select Module in Project Structure and browse to Dependencies tab –

Click on Add button followed by the click on Module Library –

**Introduction to Selenium**

7/3/2008 Page 21 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

Browse to the Selenium directory and select selenium-java-client-driver.jar and seleniumserver.

jar. (Multiple Jars can be selected b holding down the control key.) –

**Introduction to Selenium**

7/3/2008 Page 22 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

Select both jar files in project pan and click on Apply button –

**Introduction to Selenium**

7/3/2008 Page 23 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

Now click ok on Project Structure followed by click on Close on Project Settings pan. Added

jars would appear in project Library as following –

**Introduction to Selenium**

7/3/2008 Page 24 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

Create the directory structure in src folder as following –

***Note: This is not any hard and fast convention and just a convention hence it might differ***

***from project to project.***

**Introduction to Selenium**

7/3/2008 Page 25 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

Most basically *‘core’* contains the SelTestCase class which is used to create Selenium object

and fire up the browser. *‘testscripts’* package contains the test classes which extend the

SelTestCase class. Hence extended structure would look as following –

Here SelTestCase class extends **SeleneseTestCase** class of Selenium API which in turns extends

**TestCase** class of Junit API. Hence capabilities of Junit would be available in Test Scripts. Most

basic SelTestCase class can have following statements –

**Introduction to Selenium**

7/3/2008 Page 26 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

Here ***setup*** and ***tearDown*** methods of SeleneseTestCase class are overridden in SelTestCase

class.

***setup*** method fires up browser before each test method and ***tearDown*** method closes the

browser after each test method.

**Introduction to Selenium**

7/3/2008 Page 27 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

Test script for one of test method is as following –

Notice that test class GoogleTest extends the SelTestCase class and going by this way each test

class would extend the SelTestCase class. Going forward in a project all common methods

would be kept in SelTestCase class hence each of the class can use it. This is known as

***abstraction*** in java. Establishing JDBC connection can be its example.

**Running the Test:**

Click on Edit configuration in Tool Bar -

**Introduction to Selenium**

7/3/2008 Page 28 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

Click on Add symbol and select JUnit from drop down –

**Introduction to Selenium**

7/3/2008 Page 29 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

Select the package for which tests are to be run and click on Apply button followed by click on

Ok button –

**Introduction to Selenium**

7/3/2008 Page 30 of 30

Copyright 2008 - All Rights Reserved, Any Information in this document is confidential and proprietary to Iteamic

Now click on **Run** button to run the tests –

**9. Further Readings and Scope**

http://www.openqa.org/

http://safsdev.sourceforge.net/FRAMESDataDrivenTestAutomationFrameworks.htm

http://www.junit.org/

http://www.testng.org/

http://release.openqa.org/selenium-remotecontrol/

0.9.2/doc/java/com/thoughtworks/selenium/package-summary.html

http://junit.sourceforge.net/javadoc/