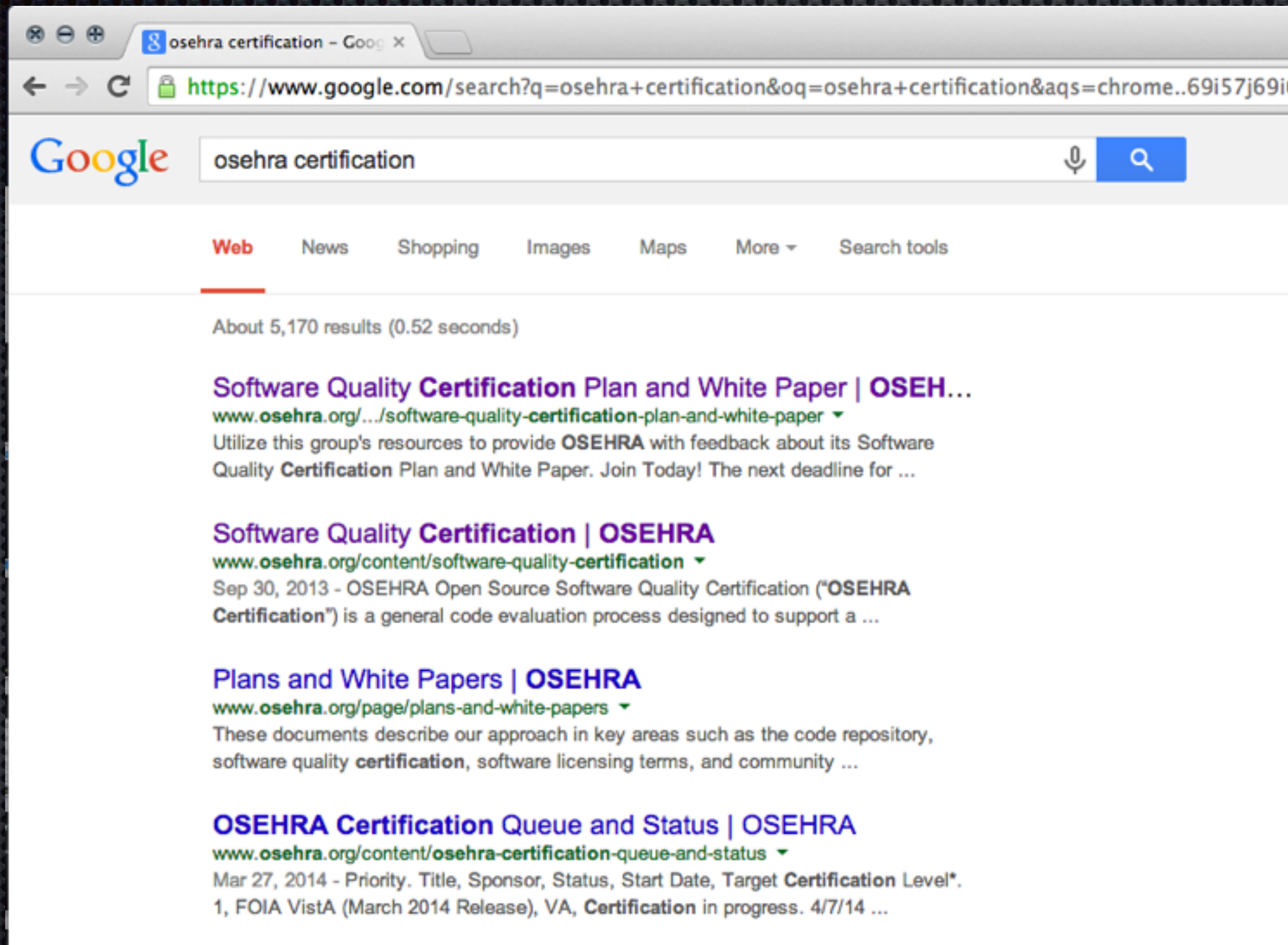


# OpenInfobutton OSEHRA Certification

Rick Bradshaw MS, PhDc  
Biomedical Informatics  
University of Utah



# Resources are Google-able





# Evaluation Areas

- ✦ Open Source License
- ✦ Documentation
- ✦ Testing
- ✦ Code Review

<http://www.osehra.org/content/certification-standards>



# Certification Levels

	<b>Name/ Number Space</b>	<b>Depen- dency / SAC</b>	<b>Open Source License</b>	<b>Documen- tation</b>	<b>Code Review</b>	<b>Test Install- tion</b>	<b>Regression Testing</b>	<b>Functional Testing</b>
<b>Level 1</b>	Pass	Pass	Apache 2	None	Large # Non- critical Issues	Large # Non- critical Issues	Existing Tests Pass	Large # Non-critical Issues
<b>Level 2</b>	Pass	Pass	Apache 2	Basic	Small # Non- critical Issues	Small # Non- critical Issues	Existing + Some R. Tests	Small # Non-critical Issues
<b>Level 3</b>	Pass	Pass	Apache 2	Substantial	No Issues	No Issues	Existing + >= 50% Coverage	No Issues
<b>Level 4</b>	Pass	Pass	Apache 2	All Required	No Issues	No Issues	Existing + >= 90% Coverage	No Issues

**Table 1 - OSEHRA software quality certification levels and minimum category expectations**



# Open Source Evaluation

- ✦ Review and validate that all dependency library licenses are Open Source
  - ✦ Apache 2.0, GPL v2.0 & 3.0, LGPL, ...
- ✦ Clearly state the license in the new software
  - ✦ Apache 2.0

<http://opensource.org>



# Documentation

- ✦ Requirements
  - ✦ Intended purpose - HL7 specification
  - ✦ Implementation-specific logic
- ✦ Installation instructions
- ✦ Testing instructions



# Documentation @ GitHub

The screenshot shows the GitHub interface for the repository **VHAINNOVATIONS / Innovation-182**. The page is for the **docs** directory in the **oib-responder** subdirectory on the **master** branch. The commit message is "Adding documentation updates; renamed files; removed unused code stubs." by RickSic, 14 days ago. The commit hash is 88f32dd774. The file list includes:

File	Commit Message	Time
maven-oracle-instructions	Adding documentation updates; renamed files; removed unused code stubs.	14 days ago
OpenInfobutton-HL7-URL-ba...	Adding documentation updates; renamed files; removed unused code stubs.	14 days ago
OpenInfobutton-Responder-...	Adding documentation updates; renamed files; removed unused code stubs.	14 days ago
OpenInfobutton-Responder-...	Adding documentation updates; renamed files; removed unused code stubs.	14 days ago
OpenInfobutton-Responder-...	Adding documentation updates; renamed files; removed unused code stubs.	14 days ago

<https://github.com/VHAINNOVATIONS/Innovation-182/tree/master/Manuals>  
<https://github.com/VHAINNOVATIONS/Innovation-182/tree/master/oib-responder/docs>



# Testing - Java IDE

The screenshot displays a Java IDE interface with three main panels:

- Left Panel (Project Explorer):** Shows a project structure with various modules. The 'Test Packages' section is expanded, highlighting the 'OpenInfobuttonResponderControllerTest.java' file.
- Top Panel (Source Editor):** Displays the source code of 'OpenInfobuttonResponderControllerTest.java'. The code includes two test methods: `testOpenInfobuttonRequestHandlerReturnsAtomPage()` and `testOpenInfobuttonRequestHandlerReturnsNoCache()`, both using Mockito for mocking and JUnit for assertions.
- Bottom Panel (Test Results):** Shows the results of the test run. A green progress bar indicates 100.00% success. The text 'All 11 tests passed.(2.801 s)' is displayed. The list of tests includes:
  - org.openinfobutton.request.service.impl.RequestServiceImplTest passed
  - testGetIndexInterpretationMap passed (0.022 s)
  - testGetFlatRequestMap passed (0.0 s)
  - testInvalidArgumentGetFlatRequestMap passed (0.0 s)
  - org.openinfobutton.responder.controller.OpenInfobuttonResponderControllerTest passed
  - testOpenInfobuttonRequestHandleMissingServletRequestParamException passed (2.253 s)
  - testOpenInfobuttonRequestHandleHttpMediaTypeNotSupportedException passed (0.043 s)
  - testOibRequestHandleIllegalArgumentException passed (0.076 s)
  - testOpenInfobuttonRequestHandlerReturnsHtmlPage passed (0.054 s)
  - testOpenInfobuttonRequestHandlerGetIndexPage passed (0.032 s)
  - testOpenInfobuttonRequestHandlerReturnsAtomPage passed (0.032 s)
  - testOpenInfobuttonRequestHandlerReturnsNoCache passed (0.033 s)
  - SanityTest passed
  - testEmptyMap passed (0.0 s)



# Automated Tests - OSEHRA

- ✦ Java tests integrated with CMake - OSEHRA's build & test framework
- ✦ OSEHRA CMake Dashboard

Login

All Dashboards



OSEHRA

Dashboard

Open Source EHR

Calendar

Previous

Current

Project

No update data as of Wednesday, May 08 2013 - 00:00 CDT

Show Filters

Advanced View

Auto-refresh

Help

Nightly Expected

Site	Build Name	Update	Configure		Build		Test			Build Time
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass	
osehravm1.kitware	 Ubuntu11.10-GTM5.4 	0	0	1	0	0	0	71	76	4 hours ago
TUCHANKA.kitware	 Win32-Cache 	0	0	1	0	1	0	68 <sup>+1</sup>	80 <sub>-4</sub>	5 hours ago
osehra-dashboard.krminc.com	 Centos5.7_x64-Cache	0			0	0	0	0	0	8 hours ago

Nightly

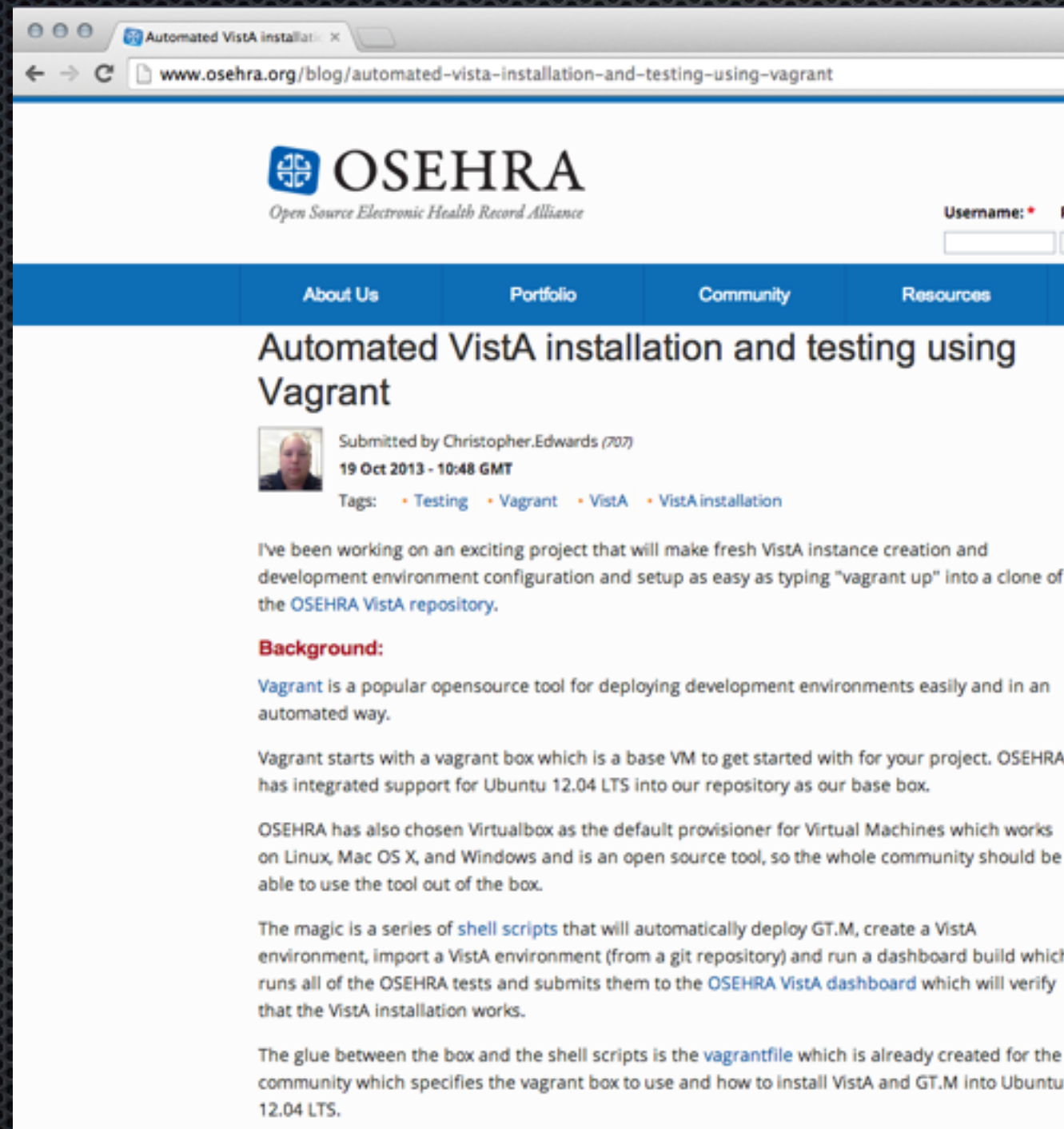
Site	Build Name	Update	Configure		Build		Test			Build Time
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass	
osehravm2.kitware	 Ubuntu13.04-GTM6.1 	0	0	1	0	0	0	73	70	8 hours ago
osehravm2.kitware	 Ubuntu13.04-GTM6.1-SupressedWarnings 	0	0	1	0	0	0	70	73	4 hours ago
osehravm1.kitware	 Ubuntu11.10-GTM5.4-SupressedWarnings 	0	0	1	0	0	0	68 <sup>+18</sup>	79 <sup>+19</sup>	8 hours ago
TUCHANKA.kitware	 Win32-Cache-SupressedWarnings 	0	0	1	0	1	0	62 <sub>-4</sub>	86 <sup>+12</sup>	8 hours ago

Experimental

Site	Build Name	Update	Configure		Build		Test			Build Time
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass	
PALAVEN	 Win32-Cache-VAEnterpriseIncrement2 		0	0	0	1	0	9 <sup>+1</sup>	15 <sub>-1</sub>	3 hours ago
TUCHANKA	 Win32-Delphi-CPRSV28		0	0	0	50	0	0	1	4 hours ago



# OSEHRA's VistA VM



The screenshot shows a web browser window with the address bar displaying `www.osehra.org/blog/automated-vista-installation-and-testing-using-vagrant`. The page features the OSEHRA logo and navigation links: About Us, Portfolio, Community, and Resources. The main content is a blog post titled "Automated VistA installation and testing using Vagrant" submitted by Christopher Edwards on October 19, 2013. The post includes tags for Testing, Vagrant, VistA, and VistA installation. The text describes a project to simplify VistA instance creation and development environment configuration using Vagrant. It mentions that Vagrant starts with a base VM (vagrant box) and that OSEHRA has integrated support for Ubuntu 12.04 LTS. It also notes that OSEHRA has chosen Virtualbox as the default provisioner for Virtual Machines. The post concludes by stating that the magic is a series of shell scripts that automatically deploy GT.M, create a VistA environment, import a VistA environment (from a git repository), and run a dashboard build which runs all of the OSEHRA tests and submits them to the OSEHRA VistA dashboard which will verify that the VistA installation works. The glue between the box and the shell scripts is the vagrantfile which is already created for the community which specifies the vagrant box to use and how to install VistA and GT.M into Ubuntu 12.04 LTS.

Automated VistA installation and testing using Vagrant

Submitted by Christopher Edwards (707)  
19 Oct 2013 - 10:48 GMT

Tags: • Testing • Vagrant • VistA • VistA installation

I've been working on an exciting project that will make fresh VistA instance creation and development environment configuration and setup as easy as typing "vagrant up" into a clone of the [OSEHRA VistA repository](#).

**Background:**

[Vagrant](#) is a popular opensource tool for deploying development environments easily and in an automated way.

Vagrant starts with a vagrant box which is a base VM to get started with for your project. OSEHRA has integrated support for Ubuntu 12.04 LTS into our repository as our base box.

OSEHRA has also chosen Virtualbox as the default provisioner for Virtual Machines which works on Linux, Mac OS X, and Windows and is an open source tool, so the whole community should be able to use the tool out of the box.

The magic is a series of [shell scripts](#) that will automatically deploy GT.M, create a VistA environment, import a VistA environment (from a git repository) and run a dashboard build which runs all of the OSEHRA tests and submits them to the [OSEHRA VistA dashboard](#) which will verify that the VistA installation works.

The glue between the box and the shell scripts is the [vagrantfile](#) which is already created for the community which specifies the vagrant box to use and how to install VistA and GT.M into Ubuntu 12.04 LTS.



# Code Review

- ✦ Engineer(s) manually reviews the code
- ✦ **Status:** OIB Code submitted. Patiently waiting ...