

Software Test & Performance CONFERENCE

October 19 - 23, 2009

Next Steps to Test Automation

Robert Walsh - President - EnvisionWare, Inc.
rwalsh@envisionware.com

Software Test
& Performance
CONFERENCE

2

Introductions

- Robert Walsh
 - President - EnvisionWare, Inc.
 - Developer
 - Agile Proponent
 - Especially Test-Driven Development (TDD) and Incremental Development

Introductions

- Name
- Title or Role
- City and State or Country
- What do you hope to learn in this class?

Agenda

- Review
 - Autolt
 - Watir
 - Fitnessse
- Fitnessse + Autolt
- Fitnessse + Watir

Overview

- Goals and Objectives
 - To demonstrate how various open source testing tools can be combined and applied to achieve greater effect

Overview

- Ground Rules
 - Participate
 - Ask questions
 - Share ideas and experiences
 - Raise doubts and concerns

Review

Review: AutoIt

- <http://www.autoitscript.com/autoit3/>
- From the AutoIt website:

AutoIt v3 is a freeware BASIC-like scripting language designed for automating the Windows GUI and general scripting. It uses a combination of simulated keystrokes, mouse movement and window/control manipulation in order to automate tasks in a way not possible or reliable with other languages (e.g. VBScript and SendKeys). AutoIt is also very small, self-contained and will run on all versions of Windows out-of-the-box with no annoying "runtimes" required!

Review: Autot

- Additional information
 - There appear to be 64-bit versions of the Window Info tool and the script runner (C:\Program Files\Autot3)
 - There is a more robust version of ScITE for Autot that includes a record-and-playback utility
 - See <http://www.autoitscript.com/autoit3/scite/downloads.shtml>

Review: Watir

- <http://wtr.rubyforge.org/> (or <http://www.watir.com>)

- From the Watir website:

Watir is an open-source library for automating web browsers. It allows you to write tests that are easy to read and maintain. It is simple and flexible.

Watir drives browsers the same way people do. It clicks links, fills in forms, presses buttons. Watir also checks results, such as whether expected text appears on the page.

Watir is a family of Ruby libraries. They support Internet Explorer on Windows, Firefox on Windows, Mac and Linux, Safari on Mac, Chrome on Windows and Flash testing with Firefox.

Like other programming languages, Ruby gives you the power to connect to databases, read data files and spreadsheets, export XML, and structure your code as reusable libraries. Unlike other programming languages, Ruby is concise and often a joy to read.

Watir stands for "Web Application Testing in Ruby". It is pronounced *water*.

Review: Watir

- The key input speed appears to be controllable
 - Use `ie.typingSpeed=new value`
- You should be able to connect to existing browser windows (or popups)
 - Use `ie.attach(how, what)`
 - Examples
 - `ie.attach(:url, 'www.google.com')`
 - `ie.attach(:title, 'Google')`
- On-line documentation
 - <http://wtr.rubyforge.org/rdoc/classes/Watir/IE.html>

Review: Fitnesse

- <http://www.fitnesse.org/>
- Excerpts from the FitNesse wiki:

FitNesse is a software development collaboration tool

FitNesse enables customers, testers, and programmers to **learn what their software should do**, and to automatically compare that to **what it actually does do**. It compares customers' expectations to actual results.

FitNesse is a software testing tool.

From another perspective, FitNesse is a lightweight, open-source framework that makes it easy for software teams to:

- Collaboratively define **AcceptanceTests** -- web pages containing simple tables of inputs and expected outputs.
- Run those tests and see the results.

FitNesse is a wiki.

- You can easily create and edit pages.

FitNesse is a web server.

- It requires **no configuration or setup**.
- Just run it and then direct your browser to the machine where it is running.

Next Steps to Test Automation

Fitnessse + Autolt

Fitnessse + Autolt

- Autolt has a command-line interface and writing a FitNesse fixture to drive it is trivial (or so I thought...)
- While the CLI is functional for most Autolt commands, I struggled with ControlGetText
 - Hard to get it to write to the console
 - Had to compile helper script to a console exe
 - Hard to identify control using all available syntax options
 - Had to use “[ID:xxx]” syntax
 - Fitnessse struggle[s | d] with leading and trailing spaces (or multiple internal spaces) in verification text
 - Had to use Ruby's String.gsub method to convert leading and trailing spaces to _

Fitnessse + Autolt

- auto_it.rb
 - Fitnessse fixture that drives Autolt through its command line interface
- Commands:
 - run(path) - launches the application under test
 - wait_for_active_window(title) - waits until a window with "title" is ready for input
 - wait_for_active_window_with_text(title, text) - waits until a window with "title" containing "text"
 - type(input) - sends key input to the focused control
 - in_window_click_on_control(title, control) - click the specified control
 - text_from_window_in_control(title, control) - get the text from the specified control

Copyright©2008-2009 Robert Walsh - All Rights Reserved

Fitnessse + Autolt

- Example

script				
start	Autolt			
run	bcp\build\InstallBarcodePlus.exe			
wait_for_active_window	EnvisionWare BarcodePlus@ Setup			
type	!n			
wait_for_active_window	EnvisionWare BarcodePlus@ Setup	with_text	License Agreement	
type	{TAB}			
type	{SPACE}			
type	!n			
wait_for_active_window	EnvisionWare BarcodePlus@ Setup	with_text	Choose Install Location	
type	<i>undefined variable: install_dir</i>			
type	{ENTER}			
wait_for_active_window	EnvisionWare BarcodePlus@ Setup	with_text	Download EnvisionWare license	
type	!i			
wait_for_active_window	EnvisionWare BarcodePlus@ Setup	with_text	Completing the EnvisionWare	
type	!f			

Copyright©2008-2009 Robert Walsh - All Rights Reserved

Fitnessse + Autolt

- Demonstration

Fitnessse + Autolt

- Lab

Next Steps to Test Automation

Fitnessse + Watir

Fitnessse + Watir

- Watir can be driven through FitNesse with a tool called Faucets (<http://faucets.rubyforge.org>)
 - From the Faucets website

Faucets is a set of fixtures that let you write web acceptance tests that will run on the RubyFit server using Watir. Tests are independent from screen layout and therefore can be used in test-first development. The syntax is simple and only require minimal understanding of HTML. No programming knowledge is required. Watir uses an OLE object to script and query Internet Explorer.

- Watin can be driven through FitNesse with a Watin plug-in for FitNesse
 - See <http://watin.sourceforge.net/faq.html> for more information

Fitnessse + Watir

- Documentation is a bit scarce
 - See "Commands.rb" in {install_path}/faucets/bin/faucets
- Key commands:
 - open - open a page
 - close - close the page
 - click - click a button or other element
 - type - enter text
 - getText - get the text of a control
 - assertText - verify that a control contains given text
 - findElement - get the element with a given id

Fitnessse + Watir

- Example

Set Up: RubyTests.FaucetUnitTests.SetUp (edit)		
Faucets::Commands		
closeAll	faucetsTestPages	
open	http://localhost:8080/files/faucetsTestPages/Base_Page.html	
Faucets::Commands		
assertValue	id=txtId	
click	id=linkId	
assertValue	id=txtId	from link with id
click	id=buttonId	
assertValue	id=txtId	from button with id
click	id=tableCellId	
assertValue	id=txtId	from cell with id
click	id=checkboxId	
assertValue	id=txtId	checkbox checked = true
assertValue	id=checkboxId	checkboxValue
click	id=checkboxId	
assertValue	id=txtId	checkbox checked = false

Fitnessse + Watir

- Demonstration

Fitnessse + Watir

- Lab

**Software Test
& Performance**
CONFERENCE

Next Steps to Test Automation

Questions?

Thank you!

**Software Test
& Performance**
CONFERENCE

26

For more information:

Robert Walsh
President
EnvisionWare, Inc.
678-584-5911
rwalsh@envisionware.com
<http://www.envisionware.com>