Behavior Driven Development

For

Ruby on Rails

Using

Cucumber, Capybara, Rspec, Selenium-WebDriver, Rcov, Launchy, etc...

James Mason, @bear454
Friday, June 18, 2010
SUSE Appliance Hack Week
It looks like you're making a presentation. Would you like some sarcasm for that?

:P
/me grumbles.
Development Methodologies

Extreme, Agile, Waterfall, Iterative, Formal, Rapid, Plan-Driven, Disciplined, etc.
Development Methodologies

Extreme, Agile, Waterfall, Iterative, Formal, Rapid, Plan-Driven, Disciplined, etc.

~= “Cowboy Coding”

when compared to

Test Driven Development
# Methodologies Compared

<table>
<thead>
<tr>
<th>Methodology</th>
<th>Developer's Role</th>
<th>QA's Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cowboy</td>
<td>Write the code, test it. Maybe.</td>
<td>Read the developer's mind. Test everything.</td>
</tr>
<tr>
<td>TDD</td>
<td>Write a test, Code to passing.</td>
<td>Expand the test suite to handle edge/corner cases and <strong>security</strong>.</td>
</tr>
</tbody>
</table>

Hey a table!
## Methodologies Compared, cont'd

<table>
<thead>
<tr>
<th>Methodology</th>
<th>Developer's Role</th>
<th>QA's Role</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cowboy</strong></td>
<td>Write the code, test it. Maybe.</td>
<td>Read the developer's mind. Test everything.</td>
</tr>
<tr>
<td><strong>TDD</strong></td>
<td>Write a test, Code to passing.</td>
<td>Expand the test suite to handle edge/corner cases and security.</td>
</tr>
<tr>
<td><strong>BDD</strong></td>
<td>Describe a feature, Code to passing.</td>
<td><em>Same as TDD.</em></td>
</tr>
</tbody>
</table>
“It’s one of those circle-of-life things”
Gherkin as a Second Language

Feature: First Things First

In order to test the cucumber stack
As a developer
I want to do something simple.

Scenario: Hello World

Given I am on the home page
Then I should see "Hello, World."

<= Name of the feature
<= Why?
<= Who?
<= What?

<= Like a test, many per Feature
<= Set the stage
<= Expectation

Also:
- "When" interim activity before "Then"
- "Background" before each scenario.
- "And" same as the prior step
- @tags – sorting and behavior
Given X, When Y, Then Z

Set the stage,
Perform some actions,
Test an expectation.

Lather, rinse, repeat.
What Cucumber does with Gherkin

Feature: First Things First
In order to test the cucumber stack
As a developer
I want to do something simple.

@wip
Scenario: Hello World
  Given I am on the home page
  And I am logged in as “Admin”
  When I follow “Click Me”
  Then I should see "Hello, World."

<= Displayed in long-form results
<= tested with 'rake cucumber:wip'
<= Resets the test environment
<= Rspec-based “step”
<= Rspec-based “step”
<= Rspec-based “step”
<= Rspec-based “step”
Setup

```
> sudo gem in cucumber cucumber-rails capybara rcov
> script/generate cucumber -capybara
  force  config/database.yml
create  config/cucumber.yml
create  config/environments/cucumber.rb
create  script/cucumber
create  features/step_definitions
create  features/step_definitions/web_steps.rb
create  features/support
create  features/support/paths.rb
create  features/support/env.rb
exists  lib/tasks
create  lib/tasks/cucumber.rake

> sudo rake gems:install RAILS_ENV=cucumber
```

You're supposed to use RPMs in home:ctso, you know. :P
config/database.yml

test: &test

cucumber:
  <<: *test
config/cucumber.yml

<% rerun = File.file?('rerun.txt') ? IO.read('rerun.txt') : "" rerun_opts = rerun.to_s.strip.empty? ? ? "--format #{{ENV['CUCUMBER_FORMAT']} | | 'progress'} features" : "--format #{{ENV['CUCUMBER_FORMAT']} | | 'pretty'}#{rerun}" std_opts = "--format #{{ENV['CUCUMBER_FORMAT']} | | 'progress'} --strict --tags ~@wip" %>
default: <%= std_opts %> features
wip: --tags @wip:3 --wip features
rerun: <%= rerun_opts %> --format rerun --out rerun.txt --strict --tags ~@wip

Is that really erb in yaml ?!? ZOMG HACK HACK HACK
:P
Cucumber defines its own environment

```ruby
config.gem 'cucumber-rails',
  :lib  => false,
  :version => '>=0.3.2' unless File.directory?(File.join(Rails.root, 'vendor/plugins/cucumber-rails'))
config.gem 'database_cleaner',
  :lib  => false,
  :version => '>=0.5.0' unless File.directory?(File.join(Rails.root, 'vendor/plugins/database_cleaner'))
config.gem 'capybara',
  :lib  => false,
  :version => '>=0.3.5' unless File.directory?(File.join(Rails.root, 'vendor/plugins/capybara'))
```

features/

your-app/
  features/
    step_definitions/
      web_steps.rb
  support/
    env.rb
    paths.rb

<= Cucumber's root
<= rcov test methods
<= a good start
<= loaded by cucumber
<= cucumber's setup
<= web_steps helper
features/support/env.rb

Defines cucumber's behavior and 'world'

# IMPORTANT: This file is generated by cucumber-rails -
# edit at your own peril.
...
require 'cucumber/formatter/unicode'
# Remove this line if you don't want Cucumber Unicode support
...
require 'cucumber/rails/capybara_javascript_emulation'
# Lets you click links with onclick javascript handlers without using
# @culerity or @javascript tags
...
Capybara.default_selector = :css
# Remove this line if you prefer XPath
...
DatabaseCleaner.strategy = :truncation
# see http://github.com/bmabey/database_cleaner
...
Prebuilt steps to get you started.

# IMPORTANT: This file is generated by cucumber-rails -
# edit at your own peril.
...

# ex: Given I am on the home page
Given /^(?::|I )am on (.+)$/ do |page_name|
  visit path_to(page_name)
end
...
...
def path_to(page_name)
  case page_name

  when /the home\s?page/
    # Add more mappings here.
    # Here is an example that pulls values out of the Regexp:
    #
    # when /^([^\s]+)\s+profile page$/i
    #   user_profile_path(User.find_by_login($1))

  ...
features/step_definitions/web_steps.rb

...  
# Use this to fill in an entire form with data from a table. Example:  
#  
#     When I fill in the following:  
#     | Account Number | 5002     |  
#     | Expiry date    | 2009-11-01 |  
#     | Note           | Nice guy |  
#     | Wants Email?   |           |  
#  
# TODO: Add support for checkbox, select og option  
# based on naming conventions.  
#  
When /^\(?:|I )fill in the following(?: within "([\"]*)")?:$/ do |selector, fields|  
  with_scope(selector) do  
    fields.rows_hash.each do |name, value|  
      When %{I fill in "#{name}" with "#{value}"}  
    end  
  end  
end  

When /^\(?:|I )fill in "([^\"]*)" with "([^\"]*)"(?: within "([^\"]*)")?:$/ do |field, value, selector|  
  with_scope(selector) do  
    fill_in(field, :with => value)  
  end  
end  
...
features/step_definitions/web_steps.rb

... # ex: Then I should see "Thank you for registering"
# ex: Then I should see "Thank you for registering" within "#flash-notice"
Then /^(?:|I )should see "([^"\*]+)"(?: within "([^"\*]+)")?$/ do |text, selector|
  with_scope(selector) do
    if page.respond_to? :should
      page.should have_content(text)
    else
      assert page.has_content?(text)
    end
  end
end
...

I thought they had to use rspec?
:P
lib/tasks/cucumber.rake

> rake -T cucumber
rake cucumber        # Alias for cucumber:ok
rake cucumber:all    # Run all features
rake cucumber:ok     # Run features that should pass
rake cucumber:rerun  # Record failing features and run only them if any exist
rake cucumber:wip    # Run features that are being worked on

Because we don't have enough rake tasks. Sigh.
“Hello, World.”

First describe a feature.

features/first_things_first.feature =>

Feature: First Things First
    In order to test the cucumber stack
    As a developer
    I want to do something simple.

Scenario: Hello World
    Given I am on the home page
    Then I should see "Hello, World."
"Hello, World."

> rake cucumber
/usr/bin/ruby -I "/usr/lib64/ruby/gems/1.8/gems/cucumber-0.8.1/lib:lib" -S rcov
--rails --exclude osx\objc,gems\ ":/usr/lib64/ruby/gems/1.8/gems/cucumber-0.8.1/bin/cucumber" -- --profile default
Using the default profile...

. F

:: failed steps (::)

<false> is not true. (Test::Unit::AssertionFailedError)
/usr/lib64/ruby/1.8/test/unit/assertions.rb:48:in `assert_block'
/usr/lib64/ruby/1.8/test/unit/assertions.rb:500:in `_wrap_assertion'
/usr/lib64/ruby/1.8/test/unit/assertions.rb:46:in `assert_block'
/usr/lib64/ruby/1.8/test/unit/assertions.rb:63:in `assert'
/usr/lib64/ruby/1.8/test/unit/assertions.rb:495:in `_wrap_assertion'
/usr/lib64/ruby/1.8/test/unit/assertions.rb:61:in `assert'
/.features/step_definitions/web_steps.rb:112
/.features/step_definitions/web_steps.rb:14:in `with_scope'
/.features/step_definitions/web_steps.rb:108:in `~/^(?::|I )should see "([^\"]*)"(?: within "([^\"]*)")?$'/.features/first_things_first.feature:8:in `Then I should see "Hello, World."'

Failing Scenarios:
cucumber features/first_things_first.feature:6 # Scenario: Hello World

1 scenario (1 failed)
2 steps (1 failed, 1 passed)
0m0.151s
“Hello, World.”

… and watch it fail.
Then code to the test.

public/index.html =>
+ <p>Hello, World.</p>
“Hello, World.”

> rake cucumber
/usr/bin/ruby -I "/usr/lib64/ruby/gems/1.8/gems/cucumber-0.8.1/lib:lib" -S rcov
--rails --exclude osx\objc,gems\ "/usr/lib64/ruby/gems/1.8/gems/cucumber-
0.8.1/bin/cucumber" -- --profile default
Using the default profile...
..

1 scenario (1 passed)
2 steps (2 passed)
0m0.183s

... and watch it pass!
“Hello, World”

So we know:

(1) Our complete rack stack is parsed by Ruby, and runs.
(2) We can make a request to the root path, and get a document back with a 2xx:OK status code (3xx:Redirects are automatically followed).
(3) The document returns contains the content we expected it to.
“I don't believe you. Show me.”

Cucumber-rails missed the 'launchy' gem, so we have to add it.

```
cfg/environments/cucumber.rb =>
+ config.gem 'launchy', :lib => false

> rake gems:install RAILS_ENV=cucumber

features/first_things_first.feature =>
  Scenario: Hello World
    Given I am on the home page
    Then I should see "Hello, World."
+   And show me the page
Hello, World.
Test Coverage with Rcov

lib/tasks/cucumber.rake =>
@@ -18,4 +18,5 @@ begin
    t.binary = vendored_cucumber_bin
    t.fork = true
    t.profile = 'default'
+   t.rcov = true
end

The first time you run 'rake cucumber:ok'
rcov will output to the new top-level folder 'coverage'
file:///path-to-rails-root/coverage/index.html

Umm... where's the hard part?
:P
<table>
<thead>
<tr>
<th>NAME</th>
<th>TOTAL LINES</th>
<th>LINES OF CODE</th>
<th>TOTAL COVERAGE</th>
<th>CODE COVERAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>app/controllers/appliance_templates_controller.rb</td>
<td>82</td>
<td>57</td>
<td>32.93%</td>
<td>14.04%</td>
</tr>
<tr>
<td>app/controllers/appliances_controller.rb</td>
<td>120</td>
<td>89</td>
<td>29.17%</td>
<td>11.24%</td>
</tr>
<tr>
<td>app/controllers/application_controller.rb</td>
<td>32</td>
<td>20</td>
<td>62.50%</td>
<td>50.00%</td>
</tr>
<tr>
<td>app/helpers/appliance_templates_helper.rb</td>
<td>2</td>
<td>2</td>
<td>100.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>app/helpers/appliances_helper.rb</td>
<td>2</td>
<td>2</td>
<td>100.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>app/helpers/application_helper.rb</td>
<td>3</td>
<td>2</td>
<td>100.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>app/models/appliance.rb</td>
<td>82</td>
<td>56</td>
<td>41.46%</td>
<td>33.93%</td>
</tr>
<tr>
<td>app/models/appliance_template.rb</td>
<td>3</td>
<td>3</td>
<td>100.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>app/models/easy_mac.rb</td>
<td>8</td>
<td>7</td>
<td>62.50%</td>
<td>57.14%</td>
</tr>
<tr>
<td>app/models/user.rb</td>
<td>13</td>
<td>11</td>
<td>53.85%</td>
<td>45.45%</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>656</strong></td>
<td><strong>462</strong></td>
<td><strong>44.66%</strong></td>
<td><strong>28.57%</strong></td>
</tr>
</tbody>
</table>

Not bad for 'Hello, World'.

:P
Browser-based Tests

Step 1
features/first_things_first.feature =>

Feature: First Things First
In order to test the cucumber stack
As a developer
I want to do something simple.

+ @javascript
  Scenario: Hello World
    Given I am on the home page
    Then I should see "Hello, World."
Browser-based Tests

Step 2
> rake cucumber
There is no step 3.

Hello, World.

Again with the jokes.

:P
Some Links

http://cukes.info
http://wiki.github.com/aslakhellesoy/cucumber
http://github.com/jnicklas/capybara
http://code.google.com/p/selenium
http://www.pragprog.com/titles/achbd <= A book, sort of
http://railscasts.com/episodes/155-beginning-with-cucumber
http://railscasts.com/episodes/159-more-on-cucumber

Links are good, right?:P