

# 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

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

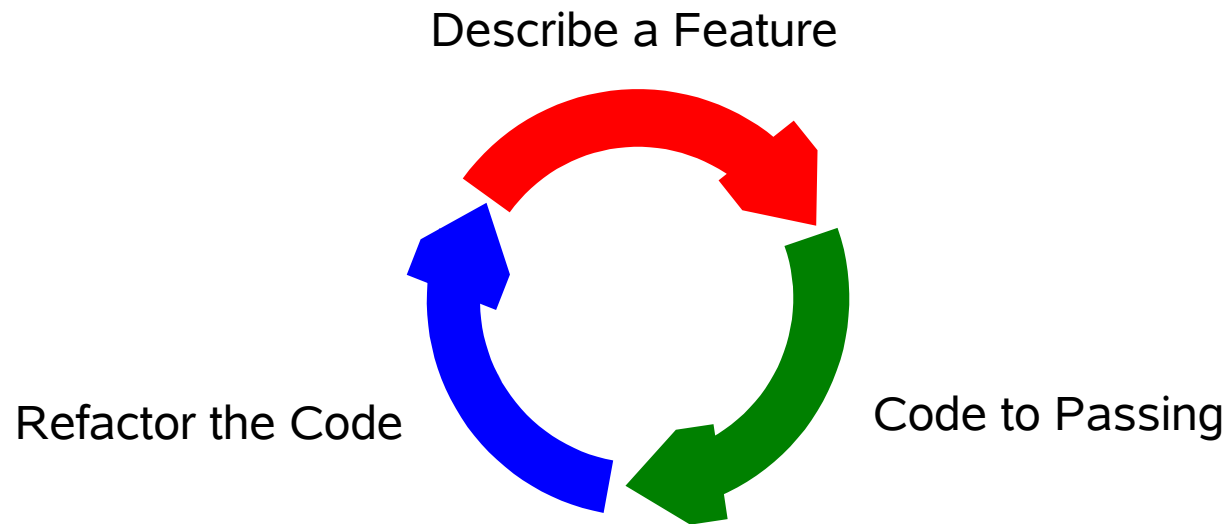
Hey a table!

:P

# Methodologies Compared, cont'd

<b>Methodology</b>	<b>Developer's Role</b>	<b>QA's Role</b>
Cowboy	Write the code, test it. Maybe.	Read the developer's mind. Test everything.
TDD	Write a test, Code to passing.	Expand the test suite to handle edge/corner cases and <u>security</u> .
BDD	Describe a feature, Code to passing.	<i>Same as TDD.</i>

“Its 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.

<= Name of the feature

<= Why?

<= Who?

<= What?

**Scenario:** Hello World

**Given** I am on the home page

**Then** I should see "Hello, World."

<= 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.

Bad jokes are my job.  
Step off.

:P

# 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.

<= Displayed in long-form results

@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."

<= 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

# config/environments/cucumber.rb

## Cucumber defines its own environment

```
...
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/

<= Cucumber's root

step\_definitions/

<= rcov test methods

web\_steps.rb

<= a good start

support/

<= loaded by cucumber

env.rb

<= cucumber's setup

paths.rb

<= 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
...
```

# features/step\_definitions/web\_steps.rb

## 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  
...
```

# features/support/paths.rb

## Helper for web\_steps.rb

```
...
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 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

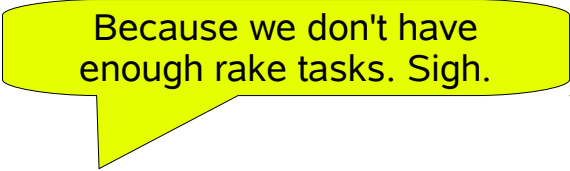
```
...
# ex: Then I should see "Thank you for registering"
# ex: Then I should see "Thank you for registering" within "#flash-notice"
Then /^(?:|I )should see "([^"]*)"(?: within "([^"]*)"?)?$/ do |text, selector|
  with_scope(selector) do
    if page.respond_to? :should
      page.should have_content(text)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

:P

# “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.

```
config/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
```

Mozilla Firefox

File Edit View Tools Help

file:///home/jmason/projects/studio/sdk/capybara-20100617142137.html

file:///home/jmas...00617142137.html

Hello, World.

Console HTML CSS Script DOM Net Firefinder

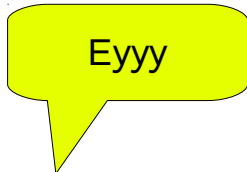
Edit | p < body < html

```
<html>
  <head></head>
  <body>
    <p>Hello, World.</p>
  </body>
</html>
```

Style Computed Layout DOM

This element has no style rules.

Done



:P

# 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

Sdk C0 Coverage Information - RCov - Mozilla Firefox

file:///home/jmason/projects/studio/sdk/coverage/index.html

Sdk C0 Coverage Information - RCov

File Filter:  Code Coverage Threshold:

NAME	TOTAL LINES	LINES OF CODE	TOTAL COVERAGE	CODE COVERAGE
app/controllers/appliance_templates_controller.rb	82	57	32.93%	14.04%
app/controllers/appliances_controller.rb	120	89	29.17%	11.24%
app/controllers/application_controller.rb	32	20	62.50%	50.00%
app/helpers/appliance_templates_helper.rb	2	2	100.00%	100.00%
app/helpers/appliances_helper.rb	2	2	100.00%	100.00%
app/helpers/application_helper.rb	3	2	100.00%	100.00%
app/models/appliance.rb	82	56	41.46%	33.93%
app/models/appliance_template.rb	3	3	100.00%	100.00%
app/models/easy_mac.rb	8	7	62.50%	57.14%
app/models/user.rb	13	11	53.85%	45.45%
<b>TOTAL</b>	<b>656</b>	<b>462</b>	<b>44.66%</b>	<b>28.57%</b>

Generated on Thu Jun 17 14:21:38 -0700 2010 with [rcov 0.9.8](#)

Done

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 cucumbe stack

As a developer

I want to do something simple.

+ @javascript

Scenario: Hello World

Given I am on the home page

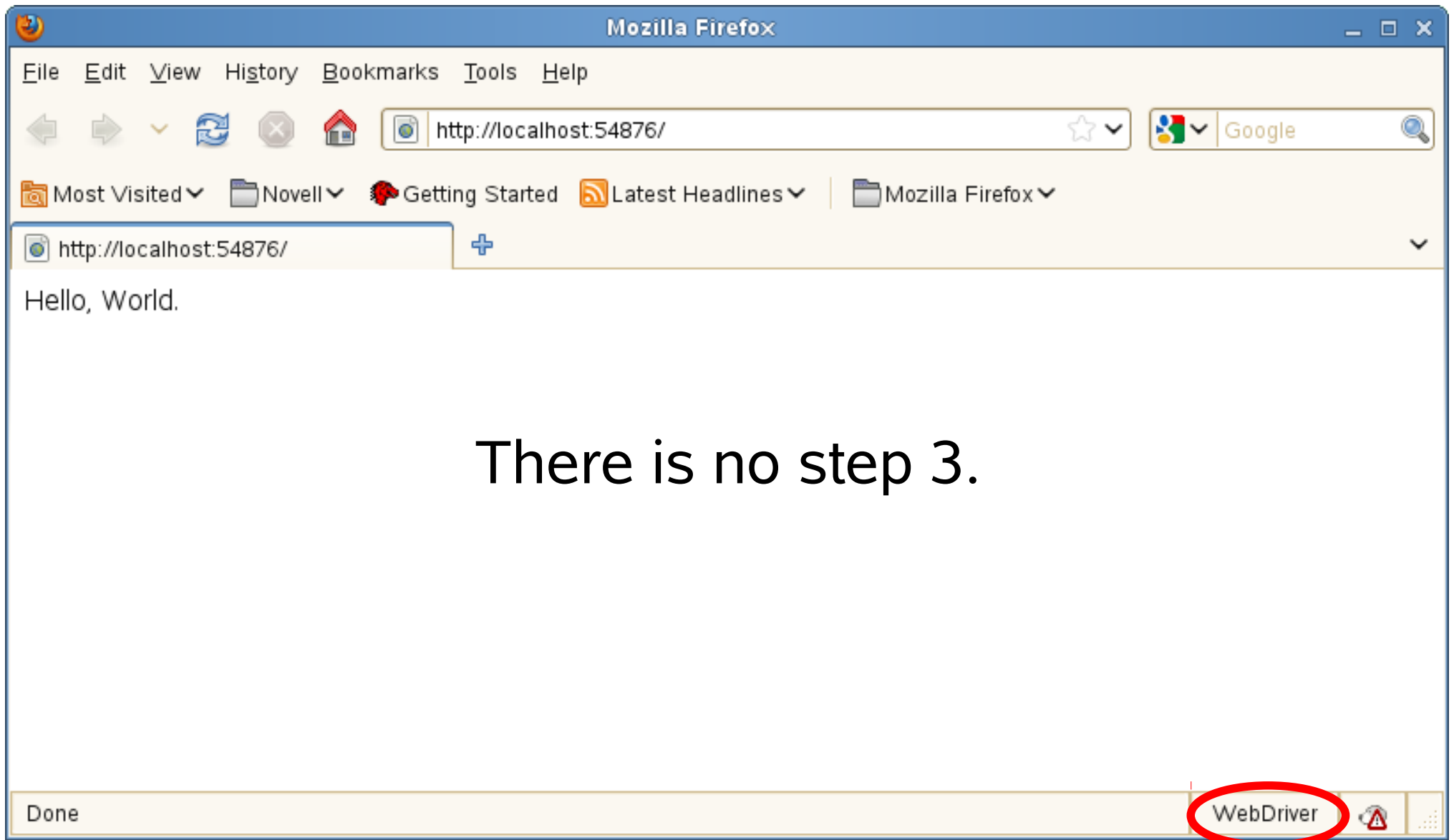
Then I should see "Hello, World."



# Browser-based Tests

## Step 2

```
> rake cucumber
```



There is no step 3.

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

« *fin* »