

Cucumber

Behavior-Driven Development Simplified

What is Cucumber?

- High-Level Integration Testing (Full-Stack)
- Plain-Text specifications readable by stakeholders
- Not just Ruby!
- Almost always used with browser automation
- Lots of uses outside of web application development

Domain Language

- Feature files should use consistent business-domain language
- Preferably readable by actual stake holders
- Crosses the bridge between requirements gathering and runnable specifications
- Can be useful in language ports

Why do we
test?

It's all about Confidence



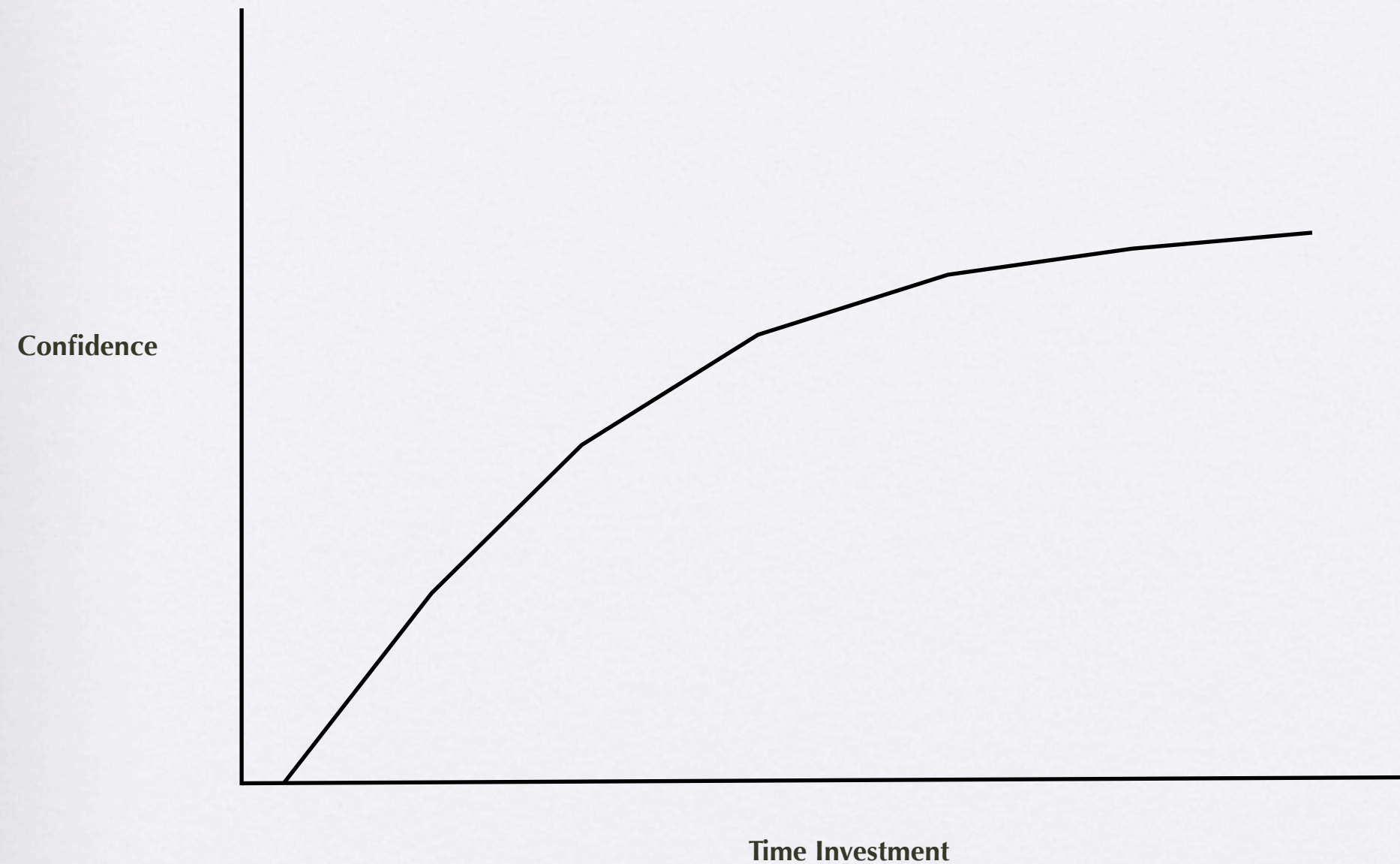
Benefits of Confidence:

- Nerves during “cap production deploy”
- Protection from regression errors when making changes
- Reduces barrier to entry to a project (others’ code)

No Free Lunch: The Costs of Confidence

- Time writing tests
- Time maintaining and refactoring tests
- Potentially greater barrier to entry for developers (testing is hard!)
- Time is worth (a lot of) money
- Opportunity Cost of Time is usually more than developer pay

The Trade-off:



Where does Cucumber fit in?

- **Ideal: Supplement to RSpec Tests (High Confidence, High Costs):**
<http://www.sarahmei.com/blog/2010/05/29/outside-in-bdd/>
- **Standalone - Very High Level (Low Confidence, Low Costs)**
- Standalone - Detailed (Medium Confidence, Medium Costs)
- Lots of “in-betweens”

Browser Automation

- Webrat
- Capybara (Multiple Drivers)

```
visit('/projects')
visit(post_comments_path(post))
click_link('id-of-link')
click_link('Link Text')
click_button('Save')
fill_in('First Name', :with => 'John')
fill_in('Password', :with => 'Seekrit')
fill_in('Description', :with => 'Really Long Text...')
choose('A Radio Button')
check('A Checkbox')
unchecked('A Checkbox')
attach_file('Image', '/path/to/image.jpg')
select('Option', :from => 'Select Box')
```

```
page.should have_xpath('//table/tr')
page.should have_css('table tr.foo')
page.should have_content('foo')
page.should have_no_content('foo')
```


Testing Javascript

@javascript

Scenario: Endless pagination with AJAX

Given 10 articles exist

When I go to the articles page

Then I should see 5 articles

When I follow "More articles"

Then I should see 10 articles

The BDD Cycle of Development

Demo

Organizing Scenarios

- @wip
- @focus
- When they should be run: @checkin, @hourly, @daily
- External dependencies: @local, @database, @network

Compound Steps

```
Given /^the user (.*) exists$/ do |name|  
  # ...  
end
```

```
Given /^I log in as (.*)$/ do |name|  
  # ...  
end
```

```
Given /^(.*) is logged in$/ do |name|  
  Given "the user #{name} exists"  
  Given "I log in as #{name}"  
end
```


Tables of Data

Feature: Manage Users

In order manage user details

As a security enthusiast

I want to edit user profiles only when authorized

Scenario: Show edit link as admin

Given the following user records

username	password	admin
bob	secret	false
admin	secret	true

And I am logged in as "admin" with password "secret"

When I visit profile for "bob"

Then I should see "Edit Profile"

Step Definition:

Given /^the following (.+) records?\$/ do |factory, table|

table.hashes.each do |hash|

Factory(factory, hash)

end

end

Managing Data: Pickle

Given a user exists with name: "Fred"

Given a user exists with name: "Fred", activated: false

Given a user exists

And a post exists with author: the user

Given a person: "fred" exists

And a person: "ethel" exists

And a fatherhood exists with parent: user "fred", child: user "ethel"

Given the following users exist

name	activated	
Fred	false	
Ethel	true	

Many-to-one assocs: "Then **a model** should be [in]one of **other model's association**", e.g.:

Then the person: "ethel" should be one of person: "fred"'s children

Then the comment should be in the post's comments

Cucumber is Slow

- Use tags to segment what should be run when
- Use “cucover” to intelligently run stories that hit code you’ve changed:
<http://github.com/mattwynne/cucover>
- Parallelize with “testjour”:
<http://github.com/brynary/testjour/>

Other Applications

- Testing Excel Formulas:
<http://adomokos.blogspot.com/2010/03/testing-excel-with-cucumber.html>
- “Behavior Driven Infrastructure”:
http://blogs.sun.com/martin/entry/behavior_driven_infrastructure
- Cucumber as a Scripting Language:
<http://holmwood.id.au/~lindsay/2009/11/01/using-cucumber-as-a-scripting-language/>
- Testing Rake Tasks with Cucumber:
<http://www.frykt.no/posts/1-2009-6-7-testing-rake-tasks-with-cucumber>