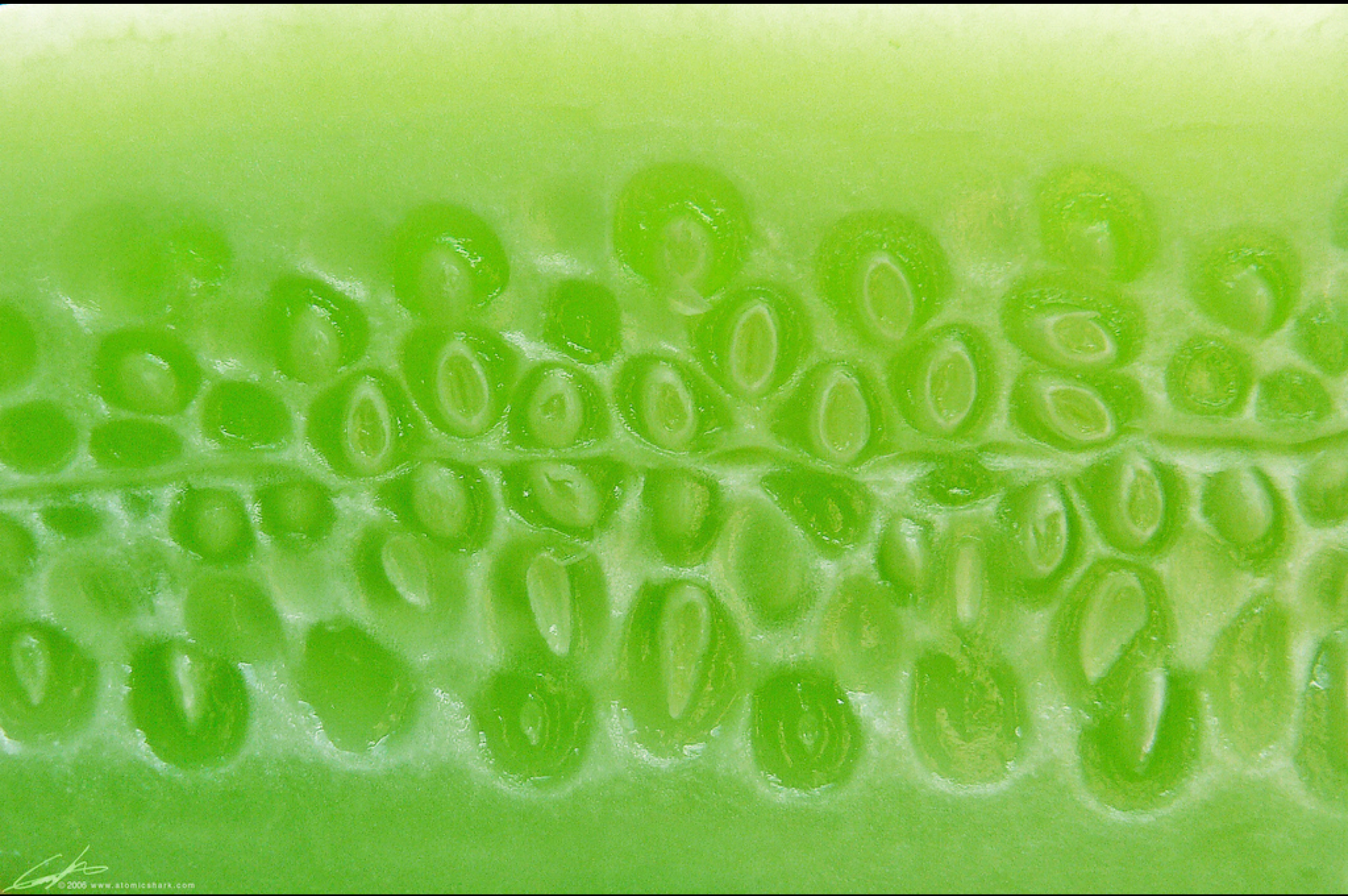
http://cukes.info/

Aslak Hellesøy - Chief Scientist

twitter.com/aslak_hellesoy
aslak.hellesoy@gmail.com



THIS IS REALLY BAD NEWS!!

http://www.flickr.com/photos/jelles/2656101758/

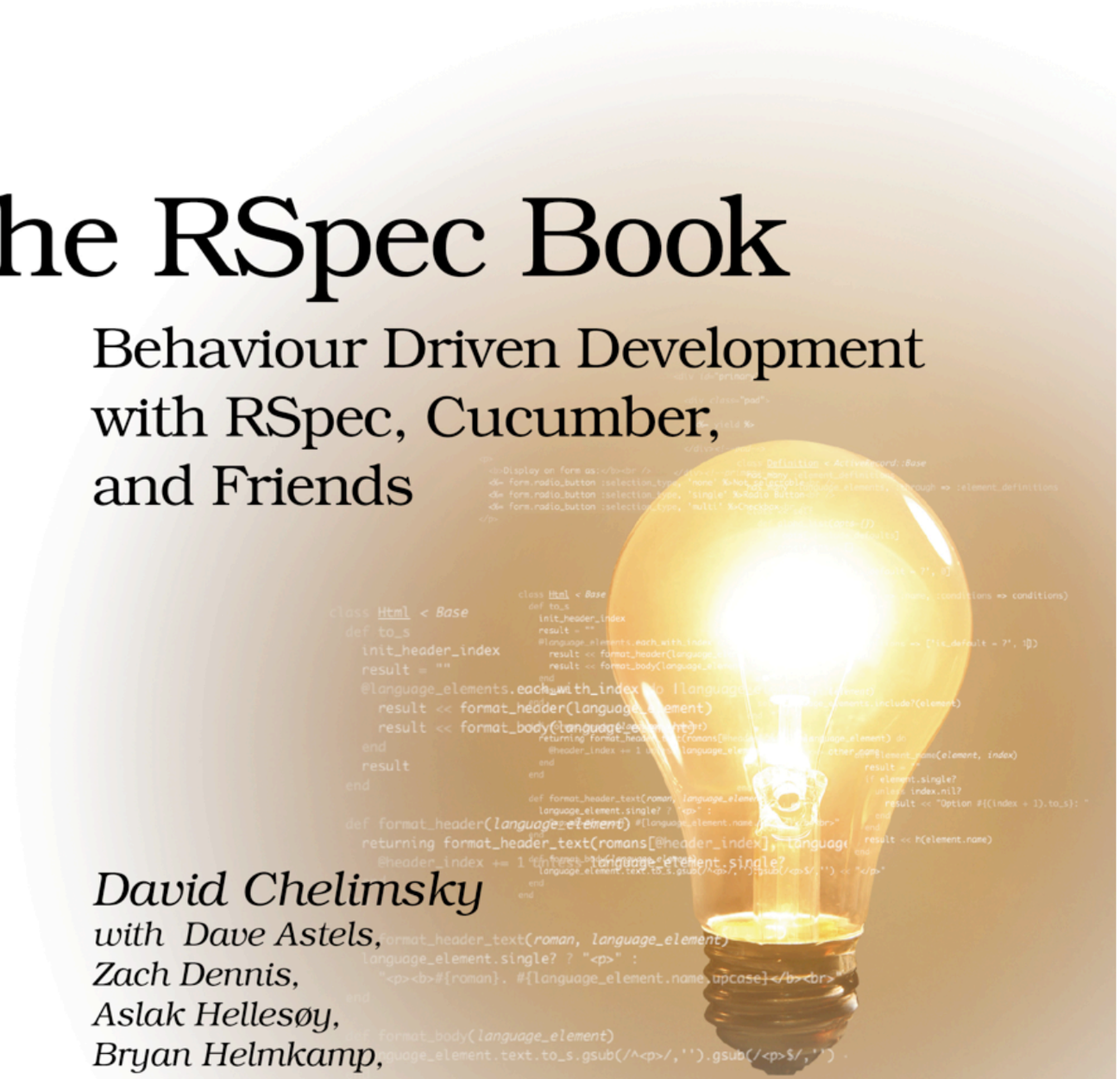
33000 gem downloads
1000 github followers
85 committers
47 wiki pages
30 related tools
6 screencasts

http://www.flickr.com/photos/twose/887903401/

The Pragmatic Programmers

The RSpec Book

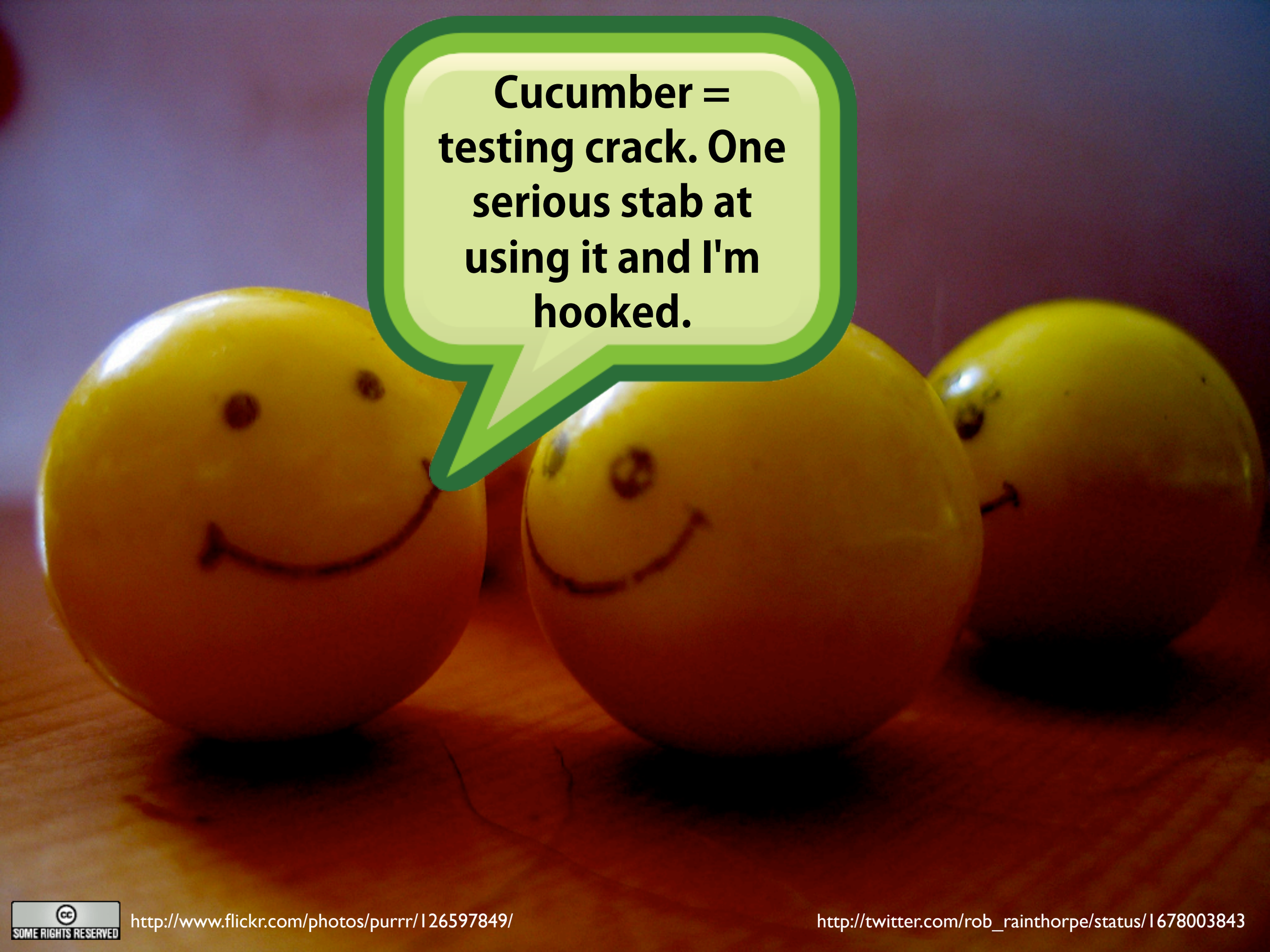
Behaviour Driven Development with RSpec, Cucumber, and Friends



David Chelimsky
with Dave Astels, Zach Dennis, Aslak Hellesøy, Bryan Helmkamp, and Dan North

Edited by Jacquelyn Carter

The Facets of Ruby Series



http://www.flickr.com/photos/purrr/126597849/

http://twitter.com/rob_rainthorpe/status/1678003843

Installing Cucumber

```
$ gem install cucumber webrat rspec-rails rspec
$ script/generate cucumber
```


features


step_definitions


webrat_steps.rb


support


env.rb


paths.rb

env.rb

```
ENV["RAILS_ENV"] ||= "test"
require File.expand_path(File.dirname(__FILE__) +
  '/../../config/environment')
require 'cucumber/rails/world'
require 'cucumber/formatter/unicode'
Cucumber::Rails.use_transactional_fixtures
Cucumber::Rails.bypass_rescue

# Plus some more setup stuff.....
```

webrat_steps.rb

```
Given /^I am on (.+)$/ do |page_name|
  visit path_to(page_name)
end

When /^I press "([^\"]*)"$/ do |button|
  click_button(button)
end

Then /^I should see "([^\"]*)"$/ do |text|
  response.should contain(text)
end
```

paths.rb

```
module NavigationHelpers
  # When /^I go to (.+)$/ do |page_name|
  def path_to(page_name)
    case page_name
    when /the homepage/
      '/'
    when /^(.*)'s profile page$/i
      user_profile_path(User.find_by_login($1))
    else
      raise "Can't find mapping for \"#{page_name}\""
    end
  end
end

World(NavigationHelpers)
```

Outside-In

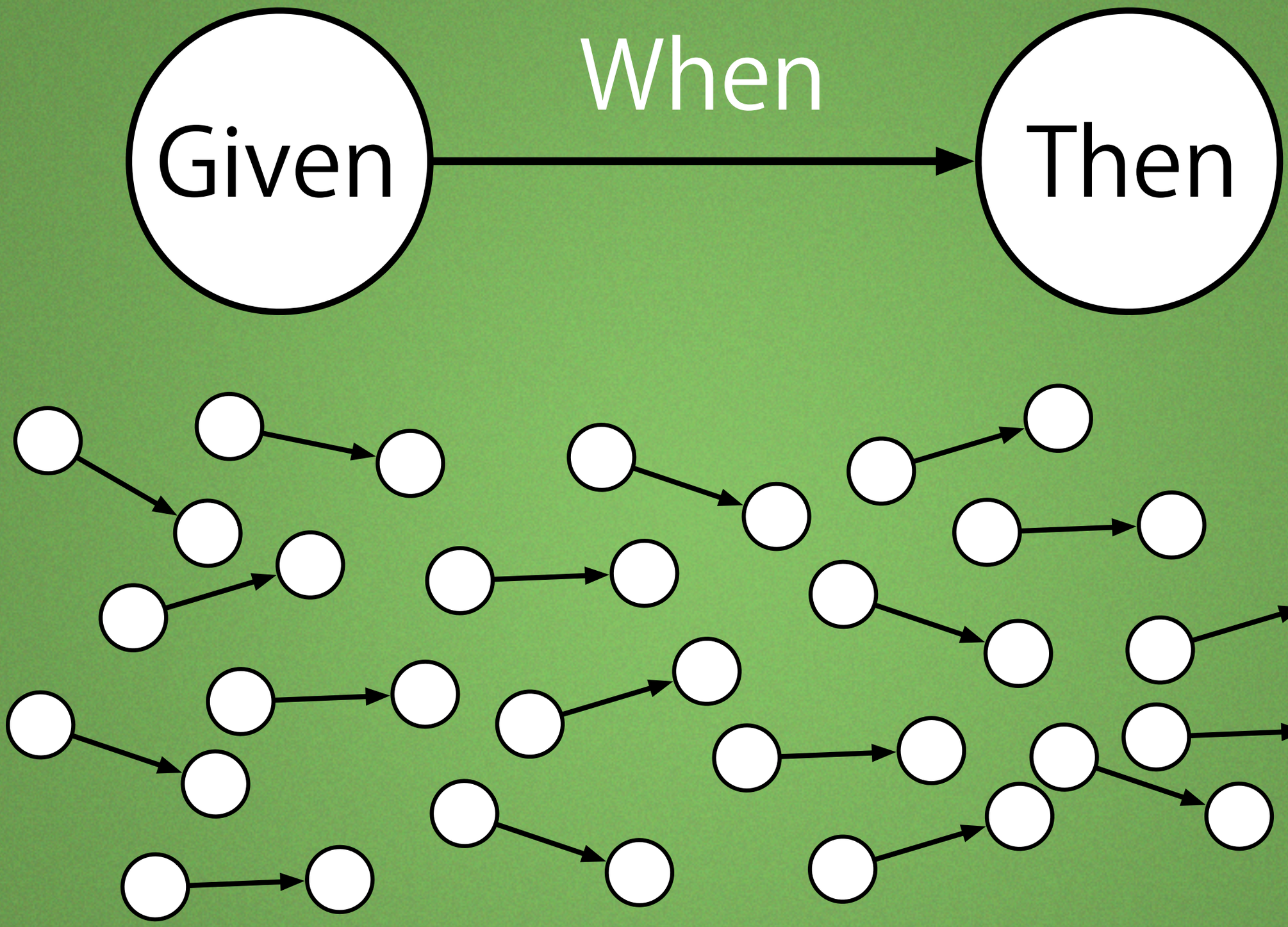
We had a glitch in our system accepting RailsConf proposals and acccidently accepted all of them. Please disregard if you got an email.

8:57 PM Feb 9th from twhirl

railsconf
RailsConf

Proposal notification

In order to reduce time spent on emailing Administrators should be able to mail proposal status to all owners


Given
When
Then

Our First Feature

```
Feature: Proposal notification
  In order to reduce time spent on emailing
  Administrators should be able to mail
  all proposals owners depending on status

  Scenario: Email accepted proposal
    Given aslak.hellesoy@gmail.com proposed Cucumber
    And the Cucumber proposal is approved
    When I send proposal emails
    Then aslak.hellesoy@gmail.com should get email
      """
      Hi aslak.hellesoy@gmail.com
      Congratulations, Cucumber was accepted.
      See you at RailsConf!
      """
```

features/proposal_notification.feature

Run the feature

```
$ cucumber features/proposal_notification.feature
```

```
Feature: Proposal notification
  In order to reduce time spent on emailing
  Administrators should be able to mail
  all proposals owners depending on status

  Scenario: Email accepted proposal
    Given aslak.hellesoy@gmail.com proposed Cucumber
    And the Cucumber proposal is approved
    When I send proposal emails
    Then aslak.hellesoy@gmail.com should get email
      """
      Hi aslak.hellesoy@gmail.com
      Congratulations, Cucumber was accepted.
      See you at RailsConf!
      """

1 scenario (1 undefined)
4 steps (4 undefined)
```

```
Feature: Proposal notification
  In order to reduce time spent on emailing
  Administrators should be able to mail
  all proposals owners depending on status

  Scenario: Email accepted proposal
    Given aslak.hellesoy@gmail.com proposed Cucumber
    And the Cucumber proposal is approved
    When I send proposal emails
    Then aslak.hellesoy@gmail.com should get email
      """
      Hi aslak.hellesoy@gmail.com
      Congratulations, Cucumber was accepted.
      See you at RailsConf!
      """

1 scenario (1 undefined)
4 steps (4 undefined)
```

```
You can implement step definitions for undefined steps
with these snippets:

Given /^aslak\.hellesoy@gmail\.com proposed Cucumber$/ do
  pending
end

Given /^the Cucumber proposal is approved$/ do
  pending
end

When /^I send proposal emails$/ do
  pending
end

Then /^aslak\.hellesoy@gmail\.com should get email$/ do
  |string|
  pending
end
```

Step Definitions

```
Given /^aslak\.hellesoy@gmail\.com proposed Cucumber$/ do
  pending
end

Given /^the Cucumber proposal is approved$/ do
  pending
end

When /^I send proposal emails$/ do
  pending
end

Then /^aslak\.hellesoy@gmail\.com should get email$/ do
  |string|
  pending
end
```

features/step_definitions/proposal_steps.rb

features
proposal_n..n.feature
step_definitions
proposal_steps.rb
webrat_steps.rb
support
env.rb
paths.rb

```
Feature: Proposal notification
  In order to reduce time spent on emailing
  Administrators should be able to mail
  all proposals owners depending on status

  Scenario: Email accepted proposal
    Given aslak.hellesoy@gmail.com proposed Cucumber
      TODO (Cucumber::Pending)
      features/step_definitions/proposal_steps.rb:2
      features/proposal_notification.feature:7
    And the Cucumber proposal is approved
    When I send proposal emails
    Then aslak.hellesoy@gmail.com should get email
      """
      Hi aslak.hellesoy@gmail.com
      Congratulations, Cucumber was accepted.
      See you at RailsConf!
      """
```

Do what Regexp says

```
Given /^aslak\.hellesoy@gmail\.com proposed Cucumber$/ do
  Proposal.create!({
    :email => 'aslak.hellesoy@gmail.com',
    :title => 'Cucumber'
  })
end
```

features/step_definitions/proposal_steps.rb

```
Feature: Proposal notification
  In order to reduce time spent on emailing
  Administrators should be able to mail
  all proposals owners depending on status

  Scenario: Email accepted proposal
    Given aslak.hellesoy@gmail.com proposed Cucumber
      uninitialized constant Proposal (NameError)
      features/step_definitions/proposal_steps.rb:2
      features/proposal_notification.feature:7
    And the Cucumber proposal is approved
    When I send proposal emails
    Then aslak.hellesoy@gmail.com should get email
      """
      Hi aslak.hellesoy@gmail.com
      Congratulations, Cucumber was accepted.
      See you at RailsConf!
      """
```

Make it Pass

```
$ script/generate rspec_scaffold proposal \
  email:string title:string approved:boolean

$ rake db:migrate db:test:clone

$ cucumber features --no-source
```

```
Feature: Proposal notification
  In order to reduce time spent on emailing
  Administrators should be able to mail
  all proposals owners depending on status

  Scenario: Email accepted proposal
    Given aslak.hellesoy@gmail.com proposed Cucumber
    And the Cucumber proposal is approved
      TODO (Cucumber::Pending)
      features/step_definitions/proposal_steps.rb:9
      features/proposal_notification.feature:8
    When I send proposal emails
    Then aslak.hellesoy@gmail.com should get email
      """
      Hi aslak.hellesoy@gmail.com
      Congratulations, Cucumber was accepted.
      See you at RailsConf!
      """
```

Implement Intention

```
Given /^the Cucumber proposal is approved$/ do
  proposal = Proposal.find_by_title('Cucumber')
  proposal.approved = true
  proposal.save!
end
```

features/step_definitions/proposal_steps.rb

```
Feature: Proposal notification
  In order to reduce time spent on emailing
  Administrators should be able to mail
  all proposals owners depending on status

  Scenario: Email accepted proposal
    Given aslak.hellesoy@gmail.com proposed Cucumber
    And the Cucumber proposal is approved
    When I send proposal emails
      TODO (Cucumber::Pending)
      features/step_definitions/proposal_steps.rb:15
      features/proposal_notification.feature:9
    Then aslak.hellesoy@gmail.com should get email
      """
      Hi aslak.hellesoy@gmail.com
      Congratulations, Cucumber was accepted.
      See you at RailsConf!
      """
```

Webrat

```
When /^I send mass proposal email$/ do
  visit('admin')
  click_button("Send proposal emails")
end
```

```
Feature: Proposal notification
  In order to reduce time spent on emailing
  Administrators should be able to mail
  all proposals owners depending on status

  Scenario: Email accepted proposal
    Given aslak.hellesoy@gmail.com proposed Cucumber
    And the Cucumber proposal is approved
    When I send mass proposal email
      No route matches "/admin" with {:method=>:get}
      features/step_definitions/proposal_steps.rb:16
      features/proposal_notification.feature:9
    Then aslak.hellesoy@gmail.com should get email
      """
      Hi aslak.hellesoy@gmail.com
      Congratulations, Cucumber was accepted.
      See you at RailsConf!
      """
```


JOHN DEERE

Add the Controller

```
class AdminController < ApplicationController
  def index
  end
end
```

```
Feature: Proposal notification
  In order to reduce time spent on emailing
  Administrators should be able to mail
  all proposals owners depending on status

  Scenario: Email accepted proposal
    Given aslak.hellesoy@gmail.com proposed Cucumber
    And the Cucumber proposal is approved
    When I send mass proposal email
    Could not find link "Send proposal emails"
    features/step_definitions/proposal_steps.rb:16
    features/proposal_notification.feature:9
    Then aslak.hellesoy@gmail.com should get email
      """
      Hi aslak.hellesoy@gmail.com
      Congratulations, Cucumber was accepted.
      See you at RailsConf!
      """
```

Add the link

```
<%= link_to("Send proposal emails",
    :action => 'mass_email') %>
```

And #mass_email

```
class AdminController < ApplicationController
  def index
  end

  def mass_email
  end
end
```

```
Feature: Proposal notification
  In order to reduce time spent on emailing
  Administrators should be able to mail
  all proposals owners depending on status

  Scenario: Email accepted proposal
    Given aslak.hellesoy@gmail.com proposed Cucumber
    And the Cucumber proposal is approved
    When I send mass proposal email
    Then aslak.hellesoy@gmail.com should get email
      """

      Hi aslak.hellesoy@gmail.com
      Congratulations, Cucumber was accepted.
      See you at RailsConf!
      """
```

Email-Spec

```
Then /^... should get email$/ do |body|
  open_email('aslak.hellesoy@gmail.com')
  current_email.body.should == body
end
```

Controller

```
class AdminController < ApplicationController
  def index
  end

  def mass_email
    approved = Proposal.find_all_by_approved(true)
    approved.each do |proposal|
      AdminMailer.deliver_notification_email(proposal)
    end
  end
end
```

Mailer

```
class AdminMailer < ActionMailer::Base
  def notification_email(proposal)
    recipients proposal.email
    from       "confreg@oreilly.com"
    subject    "Your Railsconf proposal"
    body       :proposal => proposal
  end
end
```

Mailer template

```
Hi <%= @proposal.email %>
Congratulations, <%= @proposal.title %> was accepted.
See you at RailsConf!
```

```
Feature: Proposal notification
  In order to reduce time spent on emailing
  Administrators should be able to mail
  all proposals owners depending on status

  Scenario: Email accepted proposal
    Given aslak.hellesoy@gmail.com proposed Cucumber
    And the Cucumber proposal is approved
    When I send mass proposal email
    Then aslak.hellesoy@gmail.com should get email
    """
    Hi aslak.hellesoy@gmail.com
    Congratulations, Cucumber was accepted.
    See you at RailsConf!
    """
```

Scenarios? Steps?

Steps & Step Definitions

Step == Method invocation

```
Given aslak.hellesoy@gmail.com proposed Cucumber
```

Step Definition == Method defnition

```
Given /^aslak\.hellesoy@gmail\.com proposed Cucumber$/ do
end
```

Regexp group arguments

```
Given aslak.hellesoy@gmail.com proposed Cucumber
```

```
Given /^(.+) proposed (.+)$/ do |email, proposal_name|
end
```

```
Given aslak.hellesoy@gmail.com proposed Cucumber
```

$CUCUMBER_COLORS

Quoted arguments

```
Given I have "22" cukes in my belly
```

```
Given /^I have "([^\"]*)" cukes in my belly$/ do |n|
end
```

```
Given I have "2" cukes in my belly
```

Multiline args (String)

```
Then aslak.hellesoy@gmail.com should get email
  """
  Hi aslak.hellesoy@gmail.com
  Congratulations, Cucumber was accepted.
  See you at RailsConf!
  """
```

```
Then /^(.+) should get email$/ do |email, body|
end
```

Multiline args (Tables)

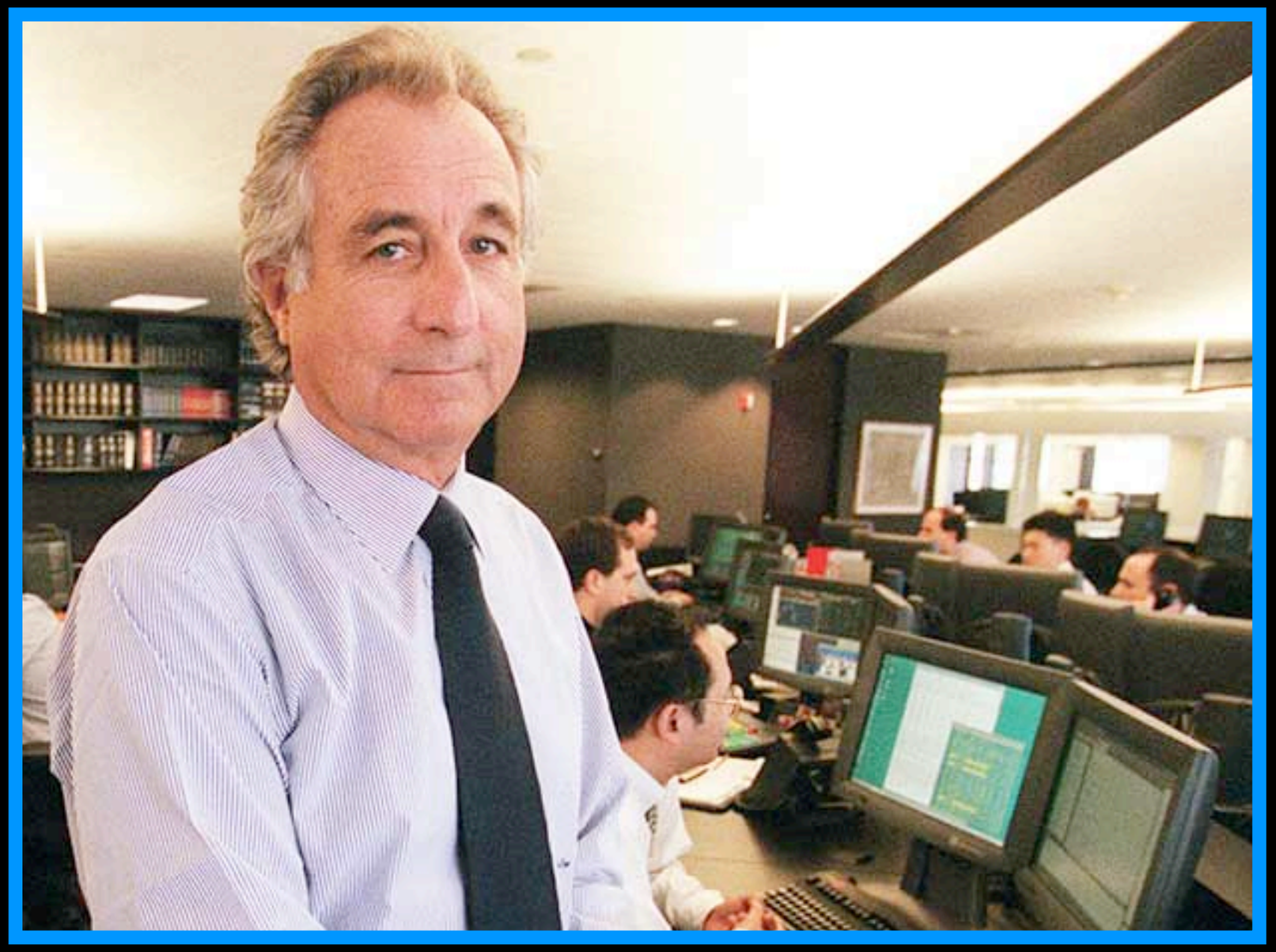
```
Given the following proposals
  | email                          | title     |
  | aslak.hellesoy@gmail.com | Cucumber |
  | bryan@brynary.com              | Webrat    |
```

```
Given /the following proposals$/ do |proposals|
  Proposal.create!(proposals.hashes)
end
```

Scenario Outline

```
Scenario Outline: Email accepted proposals
  Given the following proposals
    | email                       | title    |
    | aslak.hellesoy@gmail.com | Cucumber |
    | bryan@brynary.com           | Webrat   |
  And the <proposal> proposal is approved
  When I send proposal emails
  Then <email> should <what>

  Examples:
    | proposal | email                       | what           |
    | Cucumber | aslak.hellesoy@gmail.com | get email      |
    | Cucumber | bryan@brynary.com           | not get email |
    | Webrat   | bryan@brynary.com           | get email      |
```


```
OH HAI: STUFFING                $ cucumber -l en-lol stuffing.feature
  MISHUN: CUCUMBR
    I CAN HAZ IN TEH BEGINNIN "3" CUCUMBRZ
    WEN I EAT "2" CUCUMBRZ
    DEN I HAZ "2" CUCUMBERZ IN MAH BELLY
    AN IN TEH END "1" CUCUMBRZ KTHXBAI!
```



FAIL WITH STYLE

RSpec

```
Then /^I should have "(\d+)" cukes my belly$/ do |cukes|
  @belly.cukes.length.should == cukes.to_i
end
```

Kosher RSpec

```
Then /^I should have "(\d+)" cukes my belly$/ do |cukes|
  @belly.should have(cukes.to_i).cukes
end
```

Test::Unit

```
Then /^I should have "(\d+)" cukes my belly$/ do |cukes|
  assert_equal(@cukes.to_i, @belly.cukes.length)
end
```

Line numbers

```
Then bla bla # features/step_definitions/bla_steps.rb:16
```

Stack traces

```
When I send mass proposal email
  Could not find link "Send proposal emails"
  features/step_definitions/proposal_steps.rb:16
  features/notification.feature:9
```

```
$ cucumber features/notifications.rb:9
```

Hooks

```
Before do
end

After do |scenario|
end

World do
end

World(MyModule)
World(HerModule)
```

support/hooks.rb or support/env.rb

Background

```
Feature: Notification emails
  Background:
    Given the following proposals
      | email                      | title     |
      | aslak.hellesoy@gmail.com   | Cucumber  |
      | bryan@brynary.com          | Webrat    |

  Scenario: Approved

  Background: Rejected
```

Tagged Hooks

```
Before('@im_special', '@me_too') do
  @icecream = true
end
```

```
@me_too
Feature: Lorem
  Scenario: Ipsum
  Scenario: Dolor
```

```
Feature: Sit
  @im_special
  Scenario: Amet
  Scenario: Consec
```

Tagged Execution

```
cucumber -t spanish doit.feature
```

```
Feature: Take over the world
  I want it all

  @spanish @french @english
  Scenario: Take over Europe

  @spanish @english
  Scenario: Take over America

  @english
  Scenario: Take over Australia
```

Tagged Execution

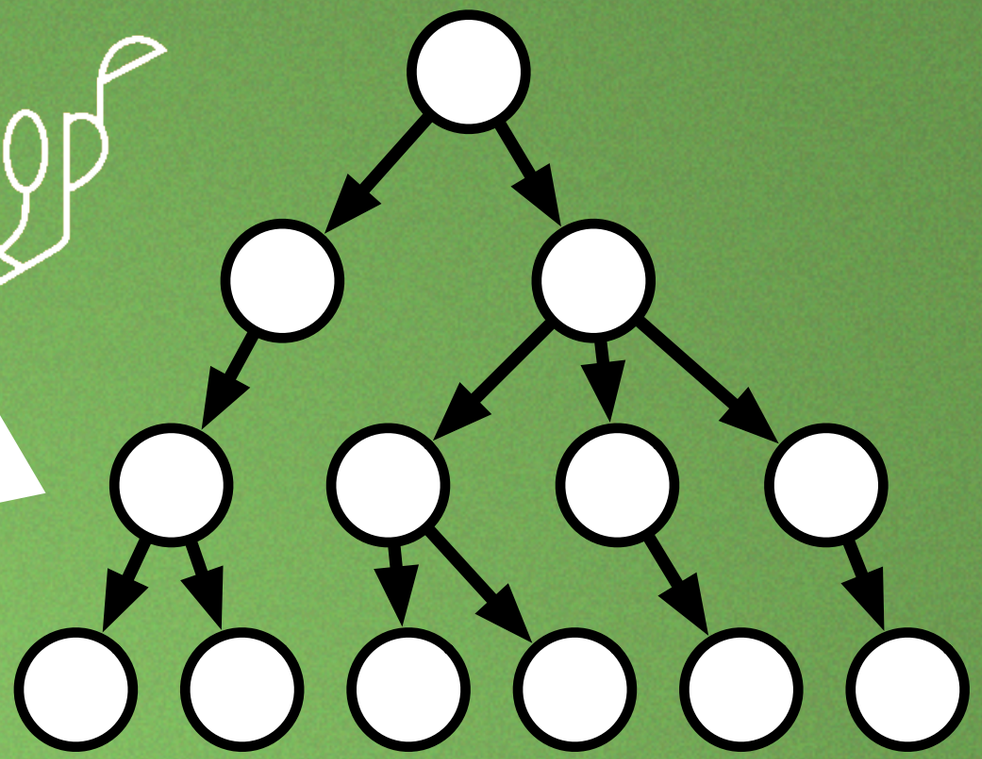
```
cucumber -t ~french doit.feature
```

```
Feature: Take over the world
  I want it all

  @spanish @french @english
  Scenario: Take over Europe

  @spanish @english
  Scenario: Take over America

  @english
  Scenario: Take over Australia
```


What's
inside?

a.feature
b.feature

treetop



x_steps.rb

y_steps.rb

RSpec/Test::Unit/Shoulda

Your Code

World Domination