Cucumber

Presentation to Victoria Ruby Users Group on using Cucumber

"Cucumber executes

plain text documentation of code
against that code."
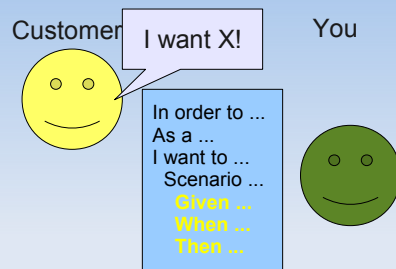
http://github.com/aslakhellesoy/cucumber/tree/master   1

Common names

- Story Driven Development
- Executable Acceptance Test Driven Development

- Write a "feature"

Customer

I want X!

You

In order to ...
As a ...
I want to ...
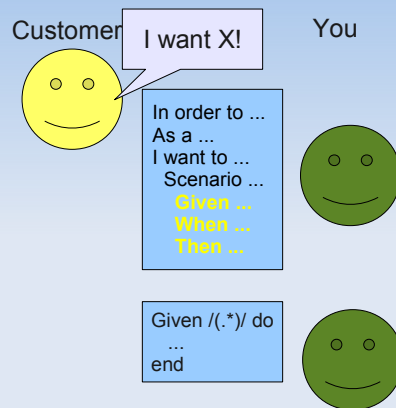 Scenario ...
   Given ...
   When ...
   Then ...

3

Basic workflow
- Write a "feature" with the customer
   - The cucumber team discourages the work "story" because this is not the same thing as an XP Story
- Implement "steps" that make the story executable
- Write the code to make the stories pass

Basic workflow
- Write a "feature" with the customer
  - The cucumber team discourages the work "story" because this is not the same thing as an XP Story
- Implement "steps" that make the story executable
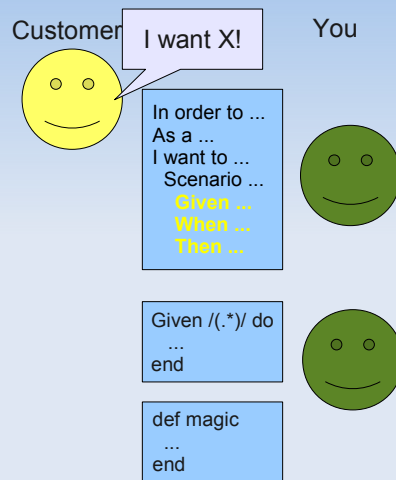- Write the code to make the stories pass

Basic workflow
- Write a "feature" with the customer
  - The cucumber team discourages the work "story"
    because this is not the same thing as an XP Story
- Implement "steps" that make the story executable
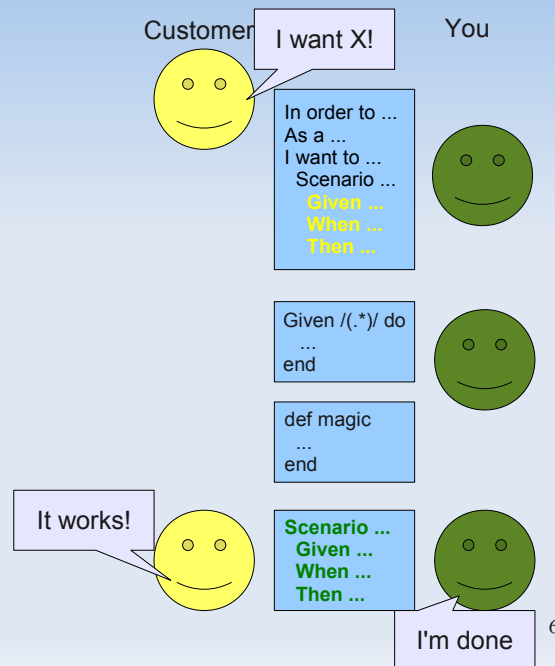- Write the code to make the stories pass

Basic workflow
- Write a "feature" with the customer
    - The cucumber team discourages the work "story"
      because this is not the same thing as an XP Story
- Implement "steps" that make the story executable
- Write the code to make the stories pass

Business Value

```
In order to be very popular
As a normal user
I want to send invites to my friends

  Scenario: Send a friend request
    Given there is a user named "Fred"
    And I am logged in as a normal user named "Joe"
    When I go to "the add a friend page"
    And I fill in "Email" with "fred@example.com"
    And I press "Add"
    Then I should see "A request has been sent to "Fred"
```

- 1. Business Value
  - Not executed
  - It is there to get you (and the customer) into the right mind set
  - Answers three questions
    - What value does this feature provide (pop the 'why' stack)
    - Who will use this
    - What is the requested feature

Scenario

```
In order to be very popular
As a normal user
I want to send invites to my friends

  Scenario: Send a friend request
    Given there is a user named "Fred"
    And I am logged in as a normal user named "Joe"
    When I go to "the add a friend page"
    And I fill in "Email" with "fred@example.com"
    And I press "Add"
    Then I should see "A request has been sent to "Fred"
```

8

- 2. Scenario Title
  - Just a short description of how this feature is used in a particular scenario

The Story

```
In order to be very popular
As a normal user
I want to send invites to my friends

  Scenario: Send a friend request
    Given there is a user named "Fred"
    And I am logged in as a normal user named "Joe"
    When I go to "the add a friend page"
    And I fill in "Email" with "fred@example.com"
    And I press "Add"
    Then I should see "A request has been sent to "Fred"
```

9

- 3. The story
  - This is actually executed
  - A collection of Given, When, and Then statements
  - Should read almost like English
  - The customer should be able to write and understand these

```
Given /I am logged in as a user named "(.*)"/ do |user|
  u = UserFactory.new(:name => user)
  visits '/login'
  fills_in('Email', :with => u.email)
  fills_in('Password', :with => 'password')
  clicks_button 'Login'
end

When /^I press "(.*)"$/ do |button|
  clicks_button(button)
end

Then /^I should see "(.*)"$/ do |text|
  response.body.should =~ /#{Regexp.escape(text)}/m
end
```

10

What do the step matchers look like?
- Simple regex
- I usually use "" to note where I will be matching
  things
  - It makes the regex more precise (less chance of a
    bug)
  - I can quickly see in my feature files what will be
    matched and what will be hard coded
- You have the choice between
  - always using the web UI
  - direct db access (i.e. ActiveRecord objects)
  - somewhere inbetween
- The choice depends on what you are testing
- I usually do web UI for what ever the 'point' of the
  story/feature is, but then ActiveRecord objects for
  the setup.
  - For example, I might create the user with active
    record objects, but then log in with form

More Examples

```
In order to not be bother by unusable links
As a normal user
I should not be allow to see the admin links

  Scenario: Hidden admin links
    Given I am logged in as a normal user named "Joe"
    When I go to "the home page"
    Then I should not see an "edit" link

  More Examples
    | User | Page                 | Link Name      |
    | Joe  | the posts index page | add post       |
    | Joe  | the reviews page     | delete comment |
```

- More Examples
  - Use the same steps, but with different values

GivenScenario

```
Scenario: Login
  Given a user with login "Paul" and password "secret"
  When I go to "the login page"
  And I fill in "Login" with "Paul"
  And I fill in "Password" with "secret"
  And I press "Login"
  Then I should see "Welcome back, Paul"

Scenario: Logout
  GivenScenario: Login
  When I click the "Logout" link
  Then I should see "You have been logged out"
```

- GivenScenario
  - Rerun the steps from previous scenarios
  - Reference it using the Scenario Title
  - It can clean things up quite a lot
  - It does couple your scenarios somewhat
  - I don't use this very often

Installation

- gem install cucumber

  For rails

  - gem install rspec rspec-rails cucumber webrat
  - ruby script/generate cucumber

  Dependencies

  - gem install term-ansicolor treetop diff-lcs hpricot

What else can you do (that I haven't done)
- Hook it up to Selenium, watir,
- Respond to emails that you app sends
  - For example, to click on an activation link in an
    email

What cucumber is not
- not a replacement for unit tests
  - It helps do TDD at a higher level, but it would be
    too cumbersome to test every edge case
- not dependent on RSpec
- not dependent no Rails

Resources

- http://github.com/aslakhellesoy/cucumber/tree/master
- http://goruco2008.confreaks.com/01_helmkamp.html
- http://www.ruby-forum.com/forum/32
- http://github.com/aslakhellesoy/cucumber/wikis
- http://smartic.us/2008/11/24/cucubmer-and-test-unit/
- #rspec
- paul@thoughtless.ca
-

What else can you do (that I haven't done)
- Hook it up to Selenium, watir,
- Respond to emails that you app sends
  - For example, to click on an activation link in an
   email

What cucumber is not
- not a replacement for unit tests
  - It helps do TDD at a higher level, but it would be
   too cumbersome to test every edge case
- not dependent on RSpec
- not dependent no Rails