Behaviour-Driven Development

@mattwynne | matt@mattwynne.net

BBC Academy

Big Picture

Day One:

- BDD Immersion

Day Two:

- Your first scenario

Tomorrow

- Bring a laptop with your system's development environment on it.
- Install Cucumber / SpecFlow / Cucumber-JVM
- Make sure we can get on the network.

Today's Goals

- Understand what BDD is, and why it might be useful to you.
- Provide an opportunity to practice identifying and writing examples.
- Build a consensus about when and how to use this technique in your team.

Agenda

09:30 - 10:15 Hello

10:15 - 11:15 How examples help

11:15 - 11:30 Break

11:30 - 12:30 Scenario-writing exercise

12:30 - 13:30 Lunch

13:30 - 14:15 Cucumber & Gherkin

14:15 - 14:30 Break

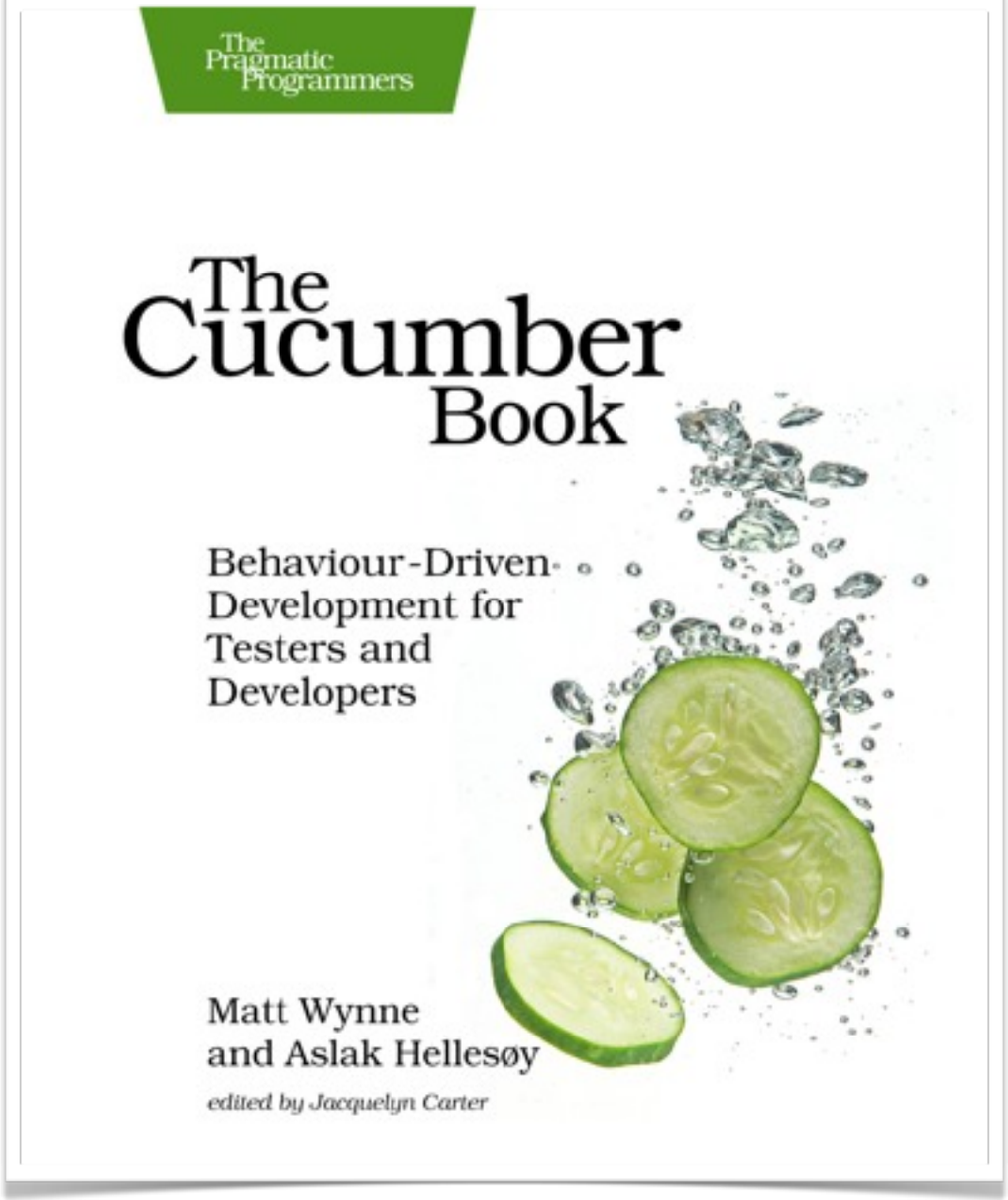
14:30 - 15:15 Readability & Style

15:15 - 15:30 Break

15:30 - 16:30 Traps, pitfalls, tips and discussion

About Me

- Programmer (Ruby, C#, JavaScript, VB, BBC BASIC)
- Coach / Team Lead (NHS, Songkick.com, BBC Worldwide)
- Agile / lean advocate
- Cucumber core team
- Author of *The Cucumber Book*



About You

- Name
- Job title / responsibilities
- The domain you work in
- Questions about acceptance testing / BDD
- Goals for today

This Morning

- How do examples help?
- What is BDD and how does it fit into this?
- Okay, can we do something now?

What is BDD?

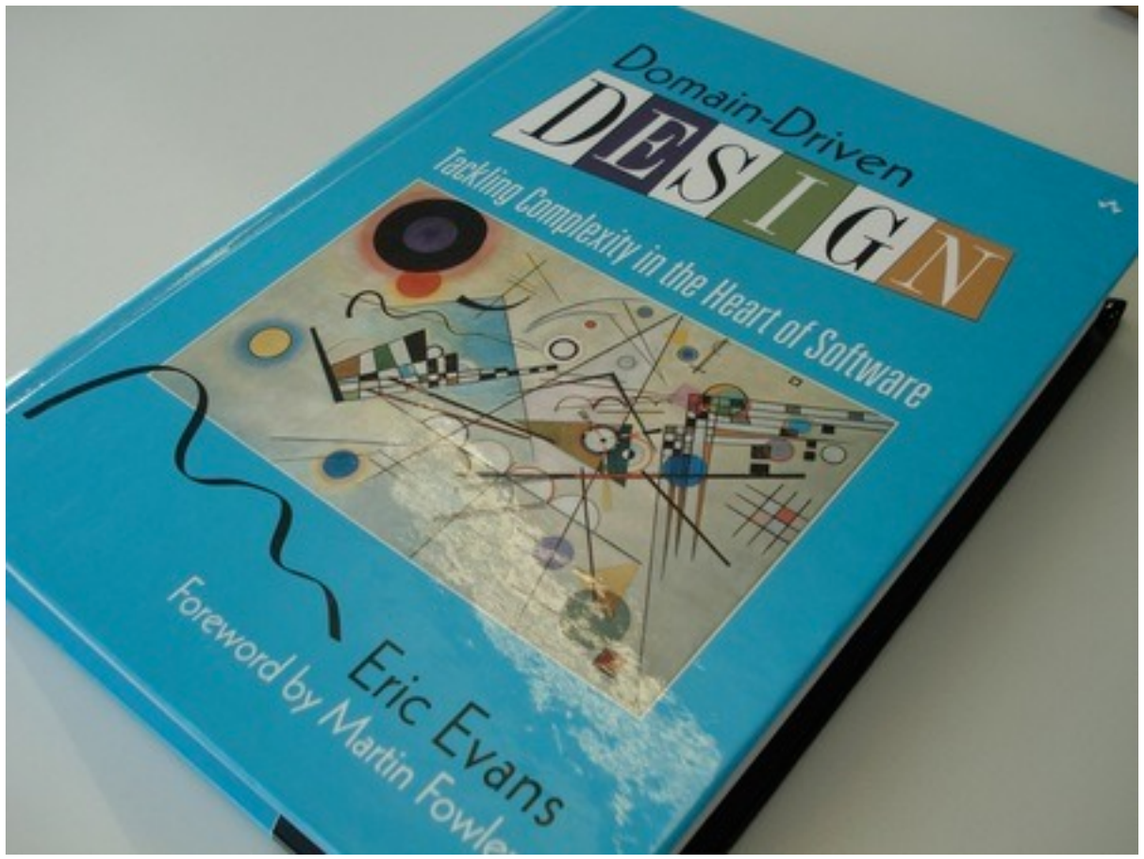
"Behaviour-Driven Development (BDD) builds upon Test-Driven Development (TDD) by formalising the good habits of the best TDD practitioners."

-- The Cucumber Book

BDD: Dual Audience

Stakeholders,
Customers,
Users

Programmers,
Testers


Domain-Driven
DESIGN
Tackling Complexity in the Heart of Software
Eric Evans
Foreword by Martin Fowle

UBIQUITOUS LANGUAGE

BBC Academy

OUTSIDE

IN

BBC Academy

Outside-in

Why?

Why?

Why?

Why?

Why?

Organisation Goal

Goal:
Increase
subscriptions

Product Feature / Epic

User / Stakeholder Story

User / Stakeholder Story

VISA subscriptions

In order to increase subscriptions

visitors should be able to

subscribe online with a VISA card

Acceptance Criteria

Credit Card Processing

Acceptance criteria:

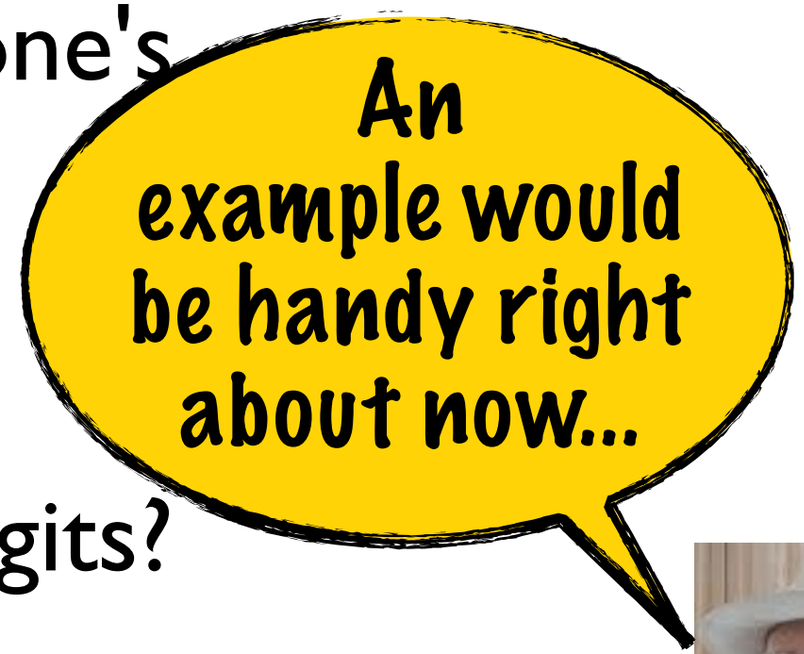
- Must support VISA
- Does not need to support MasterCard, Switch
- ...
- Customers should be prevented from entering invalid credit card details
- ...

Acceptance Criteria

"Customers should be prevented from entering invalid credit card details"

Really? So tell me...

- What exactly makes someone's credit card details invalid?
- Can they use spaces?
- Should we checksum the digits?
- How do we feed back that the details are invalid?



Scenarios

"Customers should be prevented from entering invalid credit card details"

For example, imagine this scenario:

1. Customer enters a credit card number that contains only 15 digits
2. Customer tries to submit the form
3. form is re-displayed with an error message advising Customer that the correct number of digits is 16

Can you think of some more scenarios?

Common understanding of

Done

User Story

Acceptance Criteria

Scenarios

Describe each of these in your own words

Break

The Three Amigos

BBC Academy

The Three Amigos

- Tester: thinks about what could go wrong
- Product Owner: thinks about scope
- Developer: thinks about solutions, details
- Facilitator:
  - Keeps it focussed
  - Defers things that are out of scope

The Three Amigos

1. Brainstorm
2. Defer
3. Agree

Squeaker

- We're going to build a Twitter clone
- Here are some high-level features:

- Pick a product owner
- Brainstorm a few stories that you could independently build and use.
- Make them really small
- No, even smaller than that
- Prioritise one of them

Squeaker

- For your story:
  - Now, brainstorm all the scenarios that could happen when someone tries to use this new functionality
  - Now, defer any scenarios that aren't core to this story. Put them into other user stories
  - Look at what you have left. Does this sum up the story now?

BBC Academy

How Scenarios Help

- Concrete examples help clarify misunderstandings
- Identifying one scenario helps us to see others
- Scenarios help us to explore scope

This Afternoon

- Automating Scenarios with Cucumber
- Writing Scenarios in Gherkin
- Group discussion & questions

Automating Scenarios

BBC Academy

Behaviour = Context + Action + Outcome

Given When Then

```
Scenario: Tickle a happy person

  Given I am in a good mood
  When you tickle me
  Then I will giggle
```

Scenario: Tickle a happy person

  Given I am in a good mood
  When you tickle me
  Then I will giggle

```
Scenario: Tickle a grumpy person

  Given I am in a bad mood
  When you tickle me
  Then I will giggle
```

```
Scenario: Tickle a grumpy person

  Given I am in a bad mood
  When you tickle me
  Then I will giggle
    expected: "tee hee hee",
         got: "bugger off"
```

Scenario: Tickle a grumpy person

  Given I am in a bad mood
  When you tickle me
  Then I will get cross

Scenario: Attack a happy person

Given I am in a good mood
When you kick me in the shins
Then I will get cross

Behaviour = Context + Action + Outcome

Given When Then

System = ∑(Behaviour)

Scenarios

- Automate scenarios to give developers a safety net
- Use automated scenarios to drive development - building just enough code and no more
- Scenarios written in the business domain language become *living documentation*

Automated Scenarios

http://cukes.info/


Microsoft
.NET


php


Ioke


Scala


Groovy


Javascript


Fx

Next:


ERLANG


python

Demo

BBC Academy

Gherkin

(language)

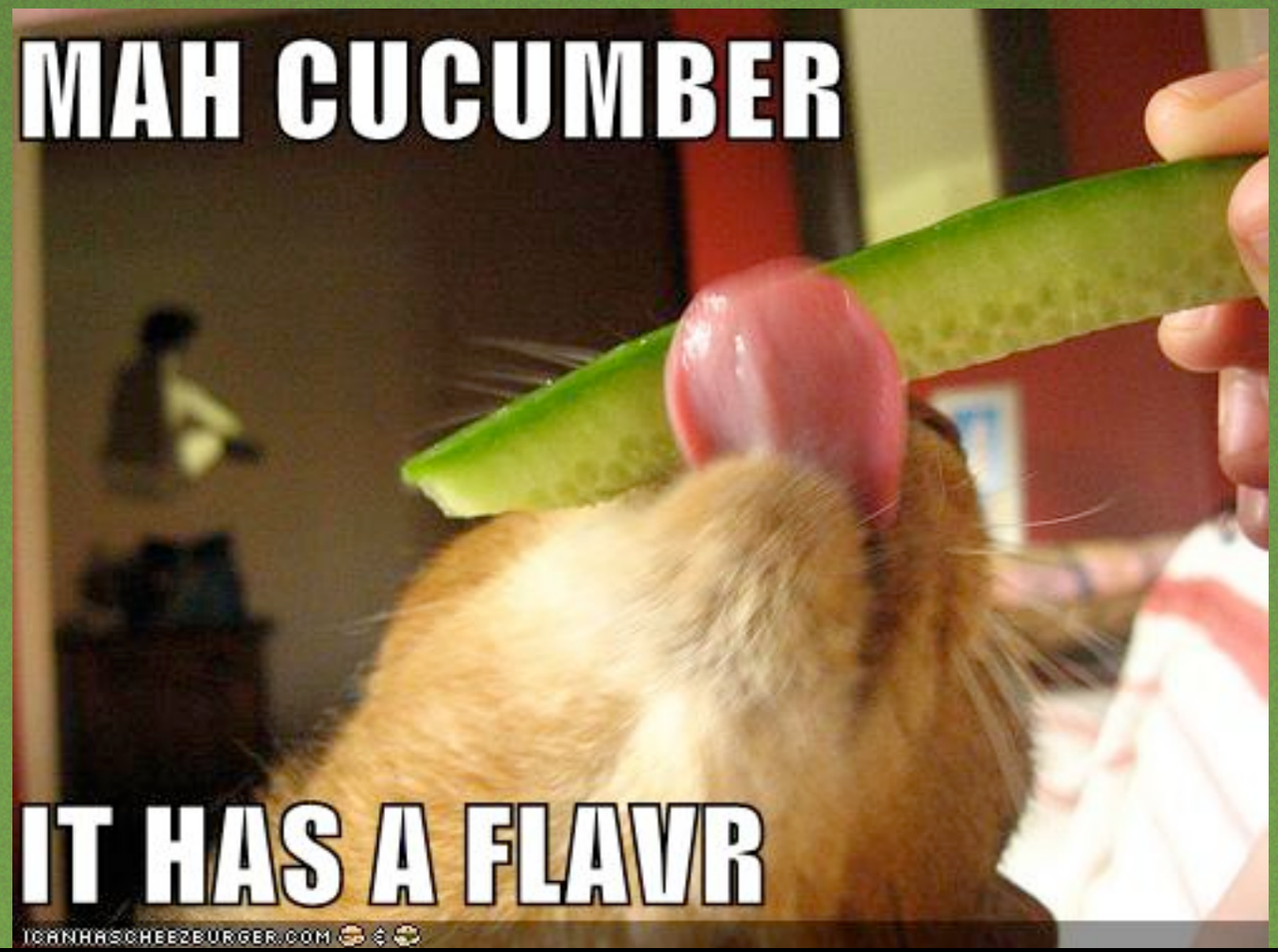
BBC Academy

```
# language: en
Feature: Division
  In order to avoid silly mistakes
  Cashiers must be able to calculate a fraction

  Scenario: Regular numbers
    Given I have entered 3 into the calculator
    And I have entered 2 into the calculator
    When I press divide
    Then the result should be 1.5 on the screen
```

български
català
Cymraeg
Česky
dansk
Deutsch
English
Australian
LOLCAT
Texan
español
eesti keel
日本語
한국어
lietuvių kalba
latviesu
Nederlands
norsk
polski
português
romana
suomi
français
hrvatski
magyar
Bahasa Indonesia
italiano
русский
Svenska
Slovensky
Узбекча
Tiếng Việt
簡体中文
繁體中文
עברית
العربية

```
# language: ja
フィーチャ: 除算
    バカな間違いを避けるために
    有理数も計算できること

    シナリオ: ふつうの数値
        前提 3 を入力
        かつ 2 を入力
        もし divide を押した
        ならば 1.5 を表示
```

BBC Academy




```
OH HAI: STUFFING
  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!
```

Gherkin Fundamentals

```
Feature: Feature name
    Description of feature goes here

    Scenario: Scenario name
        Description of scenario goes here

        Given a certain context
        When something happens
        Then an outcome
        And something else
        But not this though

    Scenario: Another scenario name
        Description of another scenario goes here
        ...
```

Multi-line Arguments

```
Given the user has set their description:
  """
  My name is Matt

  I like Cucumbers
  """
```

```
Given the users are following each other like this:
  | User | Following  |
  | Dave | Matt, Pete |
  | Pete | Matt       |
  | Matt | Pete       |
```

BBC Academy

Avoiding Duplication

- Background
- Scenario Outline

Background

```
Feature: Follow other users
  Scenario: Show followers on my page
    Given I am logged in as "Matt"
    And there is a user "Dave"
    And I am following Dave
    When I view my page
    Then I should see that I'm following Dave

  Scenario: See followed user's messages in my feed
    Given I am logged in as "Matt"
    And there is a user "Dave"
    And I am following Dave
    And Dave has posted a message "Hello"
    When I view my feed page
    Then I should see a message "Hello" from Dave
```

Background

```
Feature: Follow other users
  Background:
    Given I am logged in as "Matt"
    And there is a user "Dave"
    And I am following Dave

  Scenario: Show followers on my page
    When I view my page
    Then I should see that I'm following Dave

  Scenario: See followed user's messages in my feed
    Given Dave has posted a message "Hello"
    When I view my feed page
    Then I should see a message "Hello" from Dave
```

Scenario Outline

```
Feature: Sign up
  Scenario: No numbers in password
    When I try to sign up with the password "abcd"
    Then I should see that my password is invalid

  Scenario: Password too short
    When I try to sign up with the password "a"
    Then I should see that my password is invalid

  Scenario: Valid password
    When I try to sign up with the password "abc1"
    Then I should see that my password is valid
```

Scenario Outline

```
Feature: Sign up

  Scenario Outline: Password validation
    When I try to sign up with the password "<password>"
    Then I should see that my password is <valid or invalid>

  Examples:
  | password | valid or invalid |
  | a        | invalid          |
  | abcd     | invalid          |
  | abc1     | valid            |
```

Description

```
Feature: Sign up

  Scenario Outline: Password validation
    A password is valid if:
      - it's at least 4 characters long
      - it contains both letters and numbers

    When I try to sign up with the password "<password>"
    Then I should see that my password is "<valid or invalid>"

    Examples:
    | password | valid or invalid |
    | a        | invalid          |
    | abcd     | invalid          |
    | abc1     | valid            |
```

BBC Academy

Tags

```
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
```

BBC Academy

When should each artefact be ready?

Stakeholder/User Stories

↓

Acceptance Criteria

↓

Scenario Headlines
(*Friends* Episodes)

↓

Fleshed-out Scenarios
(Given / When / Then)

↓

Automation Code
(Step Definitions)

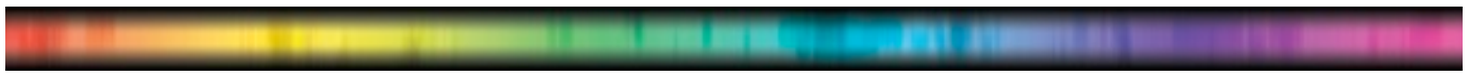
↓

System Code &
Unit tests

Who would you like to be involved at each stage?

Readability

Keep it Focussed:
Avoid incidental details

Imperative  Declarative

Imperative vs Declarative Style

```
Feature: Sign up
  Scenario: New user redirected to their own page
    Given I am not logged in
    And I visit the homepage
    And I follow "Sign up"
    And I fill in "Username" with "Matt"
    And I fill in "Password" with "password"
    And I fill in "Confirm password" with "password"
    When I press "Sign up"
    Then I should be on my feeds page
    And I should see "Hello, Matt"
```

BBC Academy

Imperative vs **Declarative** Style

```
Feature: Sign up
  Scenario: New user redirected to their own page
    When I sign up for a new account
    Then I should be taken to my feeds page
    And I should see a greeting message
```

Imperative vs **Declarative** Style

```
Feature: The entire system

  This feature illustrates what can happen when you
  take the declarative style too far.

  Scenario: It works
    When I use the system
    Then it should work perfectly
```

Imperative vs Declarative Style

```
Feature: Sign up
  Scenario: New user redirected to their own page
    Given I am not logged in
    And I visit the homepage
    And I follow "Sign up"
    And I fill in "Username" with "Matt"
    And I fill in "Password" with "password"
    And I fill in "Confirm password" with "password"
    And I press "Sign up"
    Then I should be on my feeds page
    And I should see "Hello, Matt"
```

```
Feature: Sign up
  Scenario: New user redirected to their own page
    When I sign up for a new account
    Then I should be taken to my feeds page
    And I should see a greeting message
```

BBC Academy

Whose domain is it anyway?

BBC Academy

Your Turn

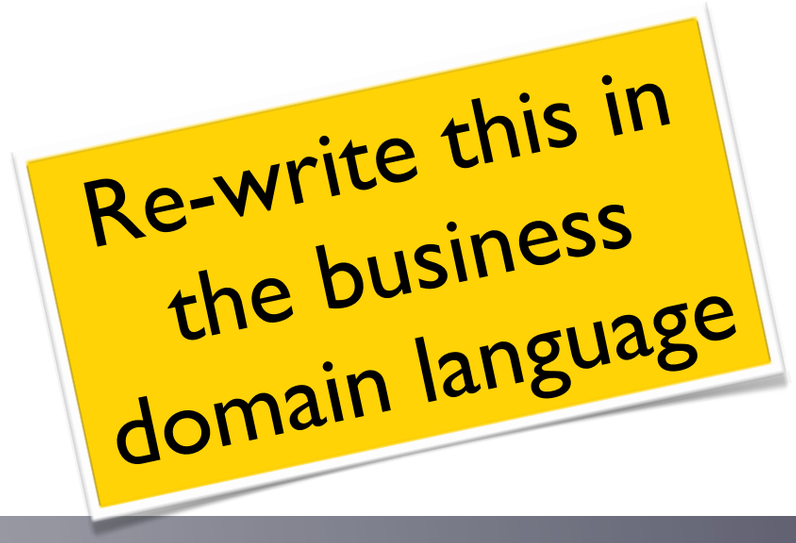
```
Feature: Sharing Links

  We want it to be easy to share guest access links, so each page of documentation
  will have a guest access link that users can copy / paste.

  Background:
    Given there is a private project
    And guest access has been enabled for the project
    And I am signed in
    And I am a collaborator on the project

  Scenario: See the Guest Access link for a single feature
    Given the project has a feature
    When I visit the page for the feature
    And I copy the "Guest Access" link
    And I sign out
    And I follow the link that I copied earlier
    Then I should be shown the page for the feature
    And I should see that I am authenticated as the project's guest user
    And I should not see the "Guest Access" link
```

BBC Academy

Your Turn

```
Scenario: Create an invoice
  Given I am an authenticated user with an admin role
  And a client "test client" exists with name: "test client", initials: "TTC"
  And a project "test project" exists with name: "test project", client: client "test client"
  And a ticket "test ticket" exists with project: project "test project", name: "test ticket"
  And a work_unit "test work unit" exists with ticket: ticket "test ticket", scheduled_at: "2010-01-01", hours: "1"
  And I am on the admin invoices page
  Then I should see "test client"
  And I follow "test client"
  And I fill in "global_invoiced" with "123"
  And I press "Submit"
  Given I am on the admin invoices page
  Then I should not see "test client"
```

https://gist.github.com/825967



BBC Academy

Are you writing tests, or documentation?

Traps, Pitfalls, Tips

Write the scenarios before the code

BBC Academy

The three amigos

Embrace debate!

Don't bury the examples away

Documentation for
Rallyround

Invitee registers

A person who is not already registered with Rallyround receives an email inviting them to help in someone's network. After completing the registration process they will find they are now a helper in the network.

Accepting an emailed invitation is thus a two-stage process:

Step 1: Click the emailed link

Step 2: Complete the registration form

Background:

Given a network for "Yoko Ono" was created by Member "Bat Man"

And I was invited via email to "dave@smith.com" to join the network "Yoko Ono"

Scenario: Step 1 - follow the link in the invitation email

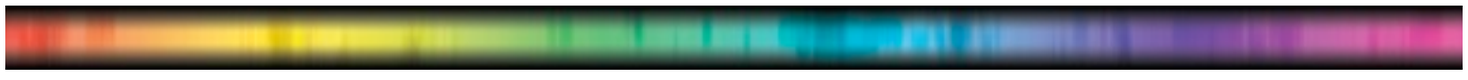
When I follow the first link in the invitation email

Then I will see a personalised Acceptance Page

Avoid brittle tests

BBC Academy

Avoid incidental details

Imperative  Declarative

BBC Academy

Feedback

Feedback

Feedback

What have we learned?

What are we going to do now?

Thanks :)

BBC Academy

BDD References

Specification By Example by Gojko Adzic
http://manning.com/adzic/

The Cucumber Book by Matt Wynne & Aslak Hellesoy
http://pragprog.com/book/hwcuc/the-cucumber-book

Domain-Driven Design by Eric Evans
http://domaindrivendesign.org/books/evans_2003

GOOS by Steve Freeman & Nat Pryce
http://www.growing-object-oriented-software.com/

Working Effectively with Legacy Code by Michael Feathers
http://www.amazon.co.uk/Working-Effectively-Legacy-Robert-Martin/dp/0131177052

Other blogs / articles
http://dhemery.com/pdf/writing_maintainable_automated_acceptance_tests.pdf
http://blog.gdinwiddie.com/
http://www.exampler.com/blog/
http://blog.mattwynne.net
http://www.agileproductdesign.com/
http://dannorth.net/2011/01/31/whose-domain-is-it-anyway/
http://agilecoach.typepad.com/agile-coaching/2011/07/when-to-write-story-tests.html
http://lizkeogh.com/2011/06/20/acceptance-criteria-vs-scenarios/