



# Behaviour Driven Development with **Ruby**



Who am I?



# Zak Grant

@zakgrant

Senior Developer  
at Rumble Labs

# What is BDD?

*“BDD is a second-generation, outside-in, pull-based, multiple-stakeholder, multiple-scale, high-automation, agile methodology.”*

**- Dan North (Creator)**

Bit of a mouthful...

In English...

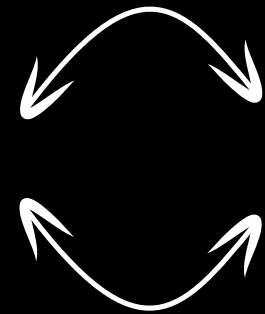
*“Describes a cycle of interactions with well-defined outputs, resulting in the delivery of working, tested software that matters.”*

**- Dan North**

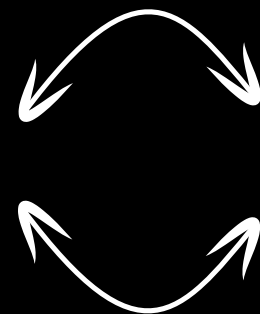


# The BDD Lifecycle

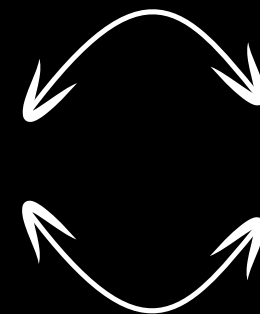
**Analyse**



**Design**



**Test**



**Code**

# Looking in from the outside

# What is required?

A feature

# A scenario for fulfilling a feature

**Feature:** Home page

**Scenario:** Reading the "hello world it's a beautiful \$TODAY" message on the home page

**Given** I have a valid URL "/"

**When** I go to the home page

**Then** I should see "Hello world it's a beautiful \$TODAY!"

**./features/index.feature**

# Run scenario



→ cucumber-sinatra-demo git:(master) ✕ cucumber

Using the default profile...

Feature: Home page

Scenario: Reading the "hello world it's a beautiful \$TODAY" message on the home page	# features/index.feature:3
Given I have a valid URL "/"	# features/index.feature:4
When I go to the home page	# features/index.feature:5
Then I should see "Hello world it's a beautiful \$TODAY!"	# features/index.feature:6

1 scenario (1 undefined)

3 steps (3 undefined)

0m0.002s

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

```
Given /^I have a valid URL "(.*?)"$/ do |arg1|
  pending # express the regexp above with the code you wish you had
end
```

```
When /^I go to the home page$/ do
  pending # express the regexp above with the code you wish you had
end
```

```
Then /^I should see "(.*?)"$/ do |arg1|
  pending # express the regexp above with the code you wish you had
end
```

# Create pending steps

```
Given /^I have a valid URL "([^"]*)"$/ do |url|  
  | @url = url  
end
```

```
When /^I go to (.*)$/ do |page|  
  | visit path_to(@url, page)  
end
```

```
Then /^I should see "([^\\"]*)"$/ do |text|  
  | today = Time.now.strftime '%A'  
  | response_body.should =~ Regexp.new(Regexp.escape(text.sub "$TODAY", today))  
end
```

**`./features/step_definitions/web_steps.rb`**

# Jump into Rspec

# Create code Example

```
require 'spec_helper'

describe TimeHelper do
  context 'Generating the Day Name for Today' do
    it 'should generate today as a string name' do
      TimeHelper.today.should eq(Time.now.strftime '%A')
    end
  end
end
```

**./spec/time\_helper\_spec.rb**

# Get example passing

```
require "rubygems"  
require "sinatra"  
require_relative "../lib/time_helper"  
  
get "/" do  
  "Hello world it's a beautiful #{TimeHelper.today}!"  
end
```

**./app.rb**



```
class TimeHelper  
  def self.today  
    Time.now.strftime '%d'  
  end  
end
```

**./lib/time\_helper.rb**

Run example...

→ cucumber-sinatra-demo git:(master) ✖ rspec

TimeHelper

Generating the Day Name for Today

should output today's name (FAILED - 1)

Failures:

1) TimeHelper Generating the Day Name for Today should output today's name

Failure/Error: TimeHelper.today.should eq(Time.now.strftime '%A')

expected: "Monday"

got: "21"

(compared using ==)

# ./spec/time\_helper\_spec.rb:6:in `block (3 levels) in <top (required)>'

Finished in 0.00179 seconds

1 example, 1 failure

Failed examples:

rspec ./spec/time\_helper\_spec.rb:5 # TimeHelper Generating the Day Name for Today should output today's name

Oops...

Fix mistake...

```
class TimeHelper  
  def self.today  
    Time.now.strftime '%A'  
  end  
end
```

**./lib/time\_helper.rb**

# Re-run example

→ cucumber-sinatra-demo git:(master) ✖ rspec

TimeHelper

Generating the Day Name for Today  
should output today's name

Finished in 0.00164 seconds  
1 example, 0 failures



# Schweet!

# Jump back up to Cucumber

# Re-run steps

```
→ cucumber-sinatra-demo git:(master) ✕ cucumber
```

```
Using the default profile...
```

```
Feature: Home page
```

```
  Scenario: Reading the "hello world it's a beautiful $TODAY" message on the home page # features/index.feature:3
    Given I have a valid URL "/" # features/step_definitions/web_steps.rb:1
    When I go to the home page # features/step_definitions/web_steps.rb:5
    Then I should see "Hello world it's a beautiful $TODAY!" # features/step_definitions/web_steps.rb:9
```

```
1 scenario (1 passed)
```

```
3 steps (3 passed)
```

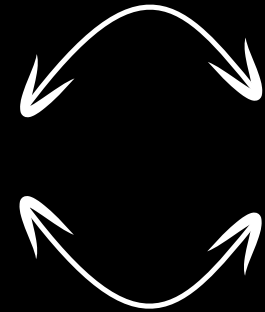
```
0m0.099s
```

Refactor & Repeat

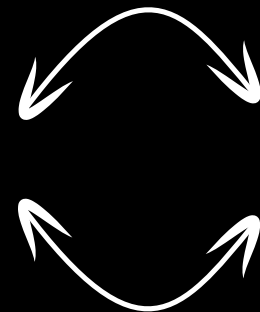
*or*

Onto the next scenario...

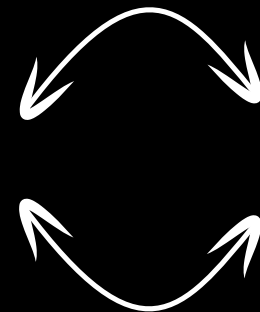
**Analyse**



**Design**



**Test**



**Code**

# Tooling

# Gherkin



**Feature:** Home page

**Scenario:** Reading the "hello world it's a beautiful \$TODAY" message on the home page

**Given** I have a valid URL "/"

**When** I go to the home page

**Then** I should see "Hello world it's a beautiful \$TODAY!"

**./features/index.feature**

# Cucumber

```
Given /^I have a valid URL "([^"]*)"$/ do |url|  
  @url = url  
end
```

```
When /^I go to (.*)$/ do |page|  
  visit path_to(@url, page)  
end
```

```
Then /^I should see "([^\\"]*)"$/ do |text|  
  today = Time.now.strftime '%A'  
  response_body.should =~ Regexp.new(Regexp.escape(text.sub "$TODAY", today))  
end
```

**`./features/step_definitions/web_steps.rb`**

# RSpec

```
require 'spec_helper'

describe TimeHelper do
  context 'Generating the Day Name for Today' do
    it 'should generate today as a string name' do
      TimeHelper.today.should eq(Time.now.strftime '%A')
    end
  end
end
```

**./spec/time\_helper\_spec.rb**

# Artefacts

# Command-line

```
→ cucumber-sinatra-demo git:(master) x cucumber
```

```
Using the default profile...
```

```
Feature: Home page
```

```
Scenario: Reading the "hello world it's a beautiful $TODAY" message on the home page # features/index.feature:3
```

```
  Given I have a valid URL "/"
```

```
# features/step_definitions/web_steps.rb:1
```

```
  When I go to the home page
```

```
# features/step_definitions/web_steps.rb:5
```

```
  Then I should see "Hello world it's a beautiful $TODAY!"
```

```
# features/step_definitions/web_steps.rb:9
```

```
1 scenario (1 passed)
```

```
3 steps (3 passed)
```

```
0m0.042s
```

—



# HTML + CSS

# Cucumber Features

1 scenario (1 passed)  
3 steps (3 passed)  
Finished in 0m0.034s seconds  
[Collapse All](#) [Expand All](#)

## Feature: Home page

Scenario: Reading the "hello world it's a beautiful \$TODAY" message on the home page		/Users/zak/Development/cucumber-sinatra-demo/features/index.feature:3
Given I have a valid URL "/"		features/step_definitions/web_steps.rb:1
When I go to the home page		features/step_definitions/web_steps.rb:5
Then I should see "Hello world it's a beautiful \$TODAY!"		features/step_definitions/web_steps.rb:9

**What ever you want!**

# In summary

**It's a mind set not a tool set**

*“Tools are just tools... using them doesn’t automatically mean you’re doing BDD”*

**- RSpec Book**

# Ubiquitous language

“Common language between all stakeholders.”



Provides a starting point

“Projects can start quickly without the need for waiting on verbose and complex requirements documents.”

# Working software

“Not only will you have working software, you will have the ability to determine easily if future changes break your already working software.”

# Documentation is free

“Most other methodologies have documentation artifacts as an intrinsic component of their lifecycle. With BDD, documentation is provided for free!”

# Resources

GitHub Cucumber Repo

GitHub Rspec Repo

Google

Pragprog Cucumber Book

Pragprog Rspec Book

The power of your imagination!

# Code

[https://github.com/zakgrant/  
cucumber-sinatra-demo](https://github.com/zakgrant/cucumber-sinatra-demo)



# BDD

Why not take it for a spin?



# Thanks & good luck!

@zakgrant