45426: Teste e Qualidade de Software

BDD: behavior driven testing

Ilídio Oliveira

v2020-03-03

universidade de aveiro
departamento de eletrónica,
telecomunicações e informática
deti

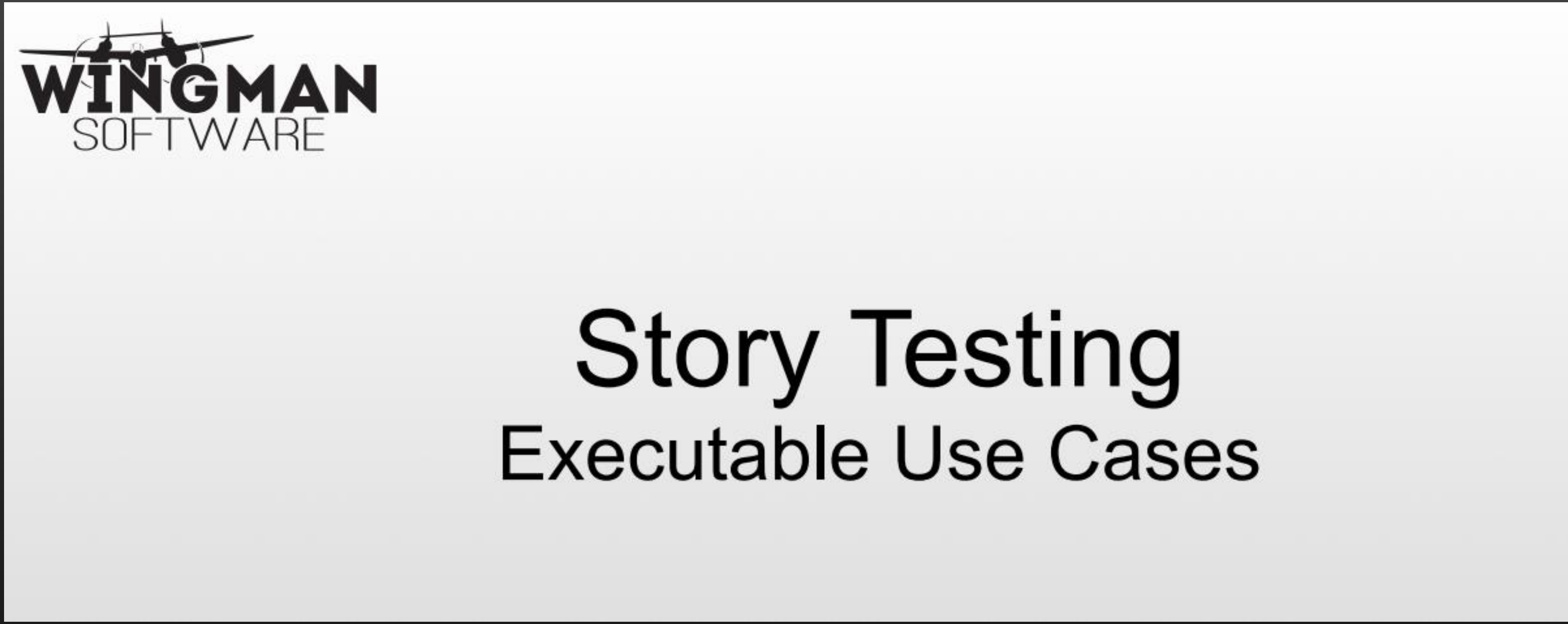
Learning objectives

Explain how “features/user-stories” are used as a conversation tool to build functional specifications

Write simple acceptance criteria for a user story in structured text

Write acceptance scenarios using the Gerking language

Describe the steps to implement BDD in Java using the Cucumber framework



WINGMAN SOFTWARE

Story Testing

Executable Use Cases

https://wingman-sw.com/articles/story-testing-embedded

James Grenning, founder of Wingman Software, trains, coaches and consults worldwide. With decades of software development experience, both technical and managerial, James brings a wealth of knowledge, skill, and creativity to software development teams and their management. As his professional roots are in embedded software, he is leading the way to introduce Agile development practices to that challenging world. See James' articles for applying Agile to embedded software development.

Stories and scenarios

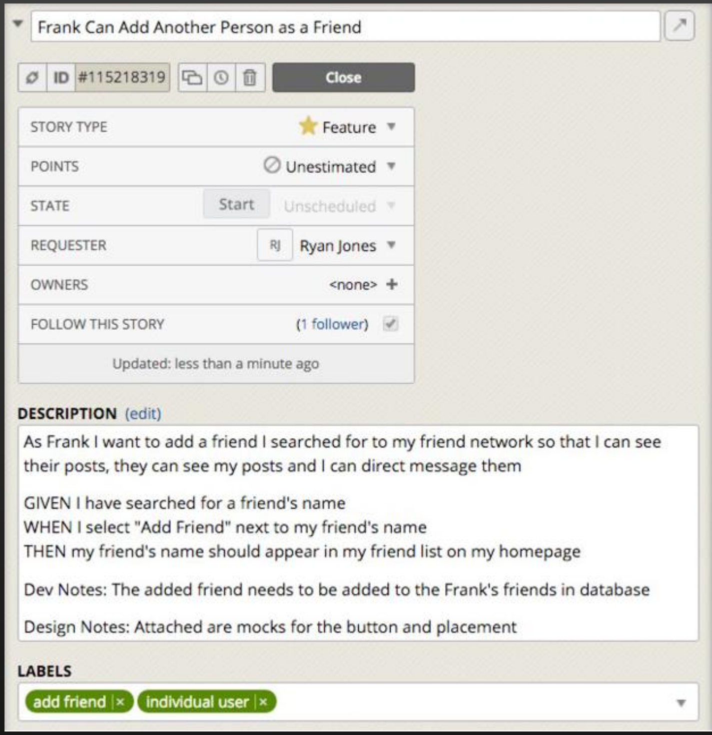
(User) Story as the basic unit of functionality, and therefore of delivery.

Captures a feature of the system

defines the scope of the feature and its acceptance criteria.

They are also used as the basis for estimation when we come to do our planning

Can be mapped on outcomes, requirements


Frank Can Add Another Person as a Friend

ID #115218319 Close

STORY TYPE	Feature
POINTS	Unestimated
STATE	Start Unscheduled
REQUESTER	RJ Ryan Jones
OWNERS	<none> +
FOLLOW THIS STORY	(1 follower)

Updated: less than a minute ago

DESCRIPTION (edit)

As Frank I want to add a friend I searched for to my friend network so that I can see their posts, they can see my posts and I can direct message them

GIVEN I have searched for a friend's name
WHEN I select "Add Friend" next to my friend's name
THEN my friend's name should appear in my friend list on my homepage

Dev Notes: The added friend needs to be added to the Frank's friends in database

Design Notes: Attached are mocks for the button and placement

LABELS

add friend × individual user ×

https://www.pivotaltracker.com/blog/principles-of-effective-story-writing-the-pivotal-labs-way

What's in a Story?
http://dannorth.net/whats-in-a-story/

Stories, use cases, scenarios

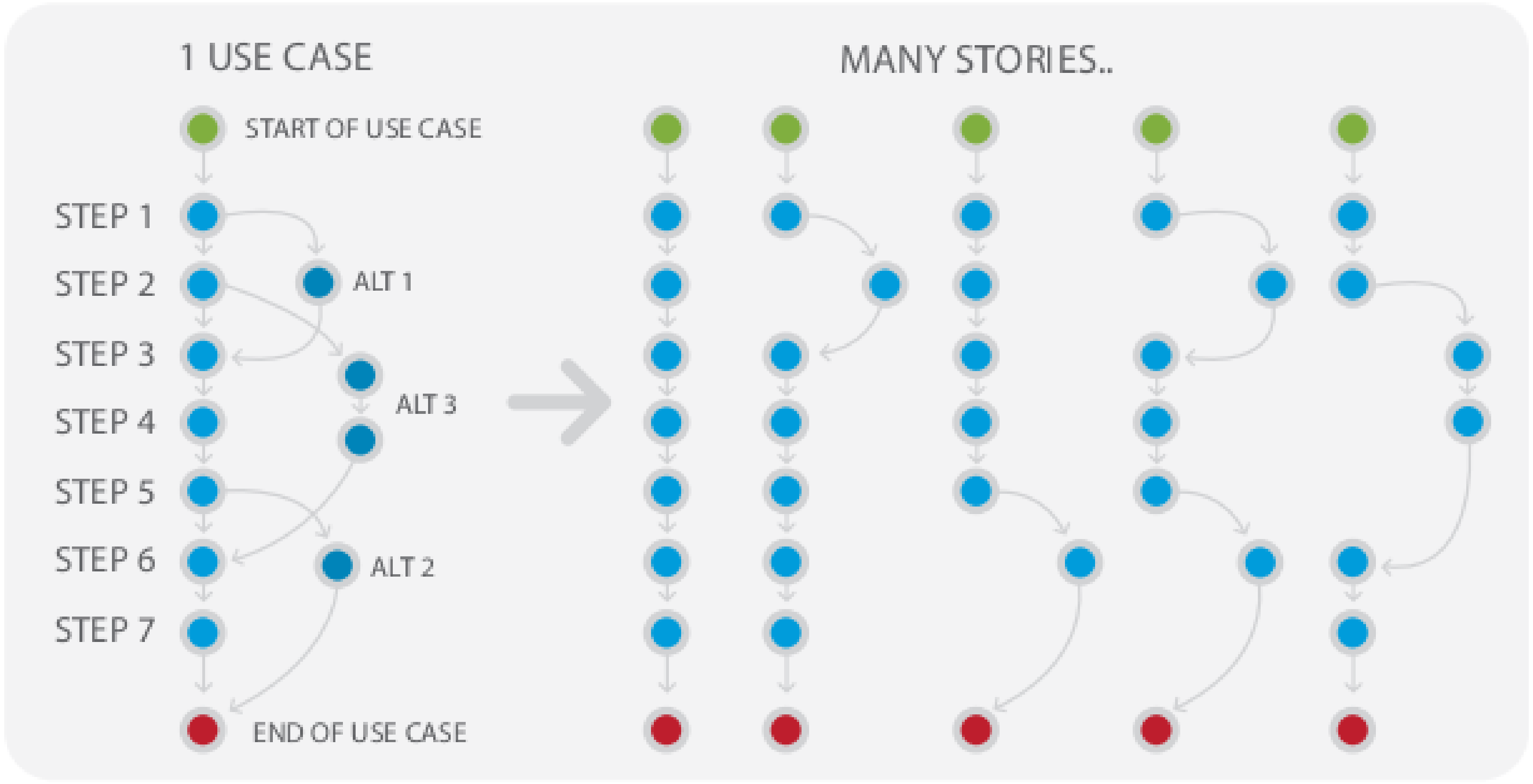
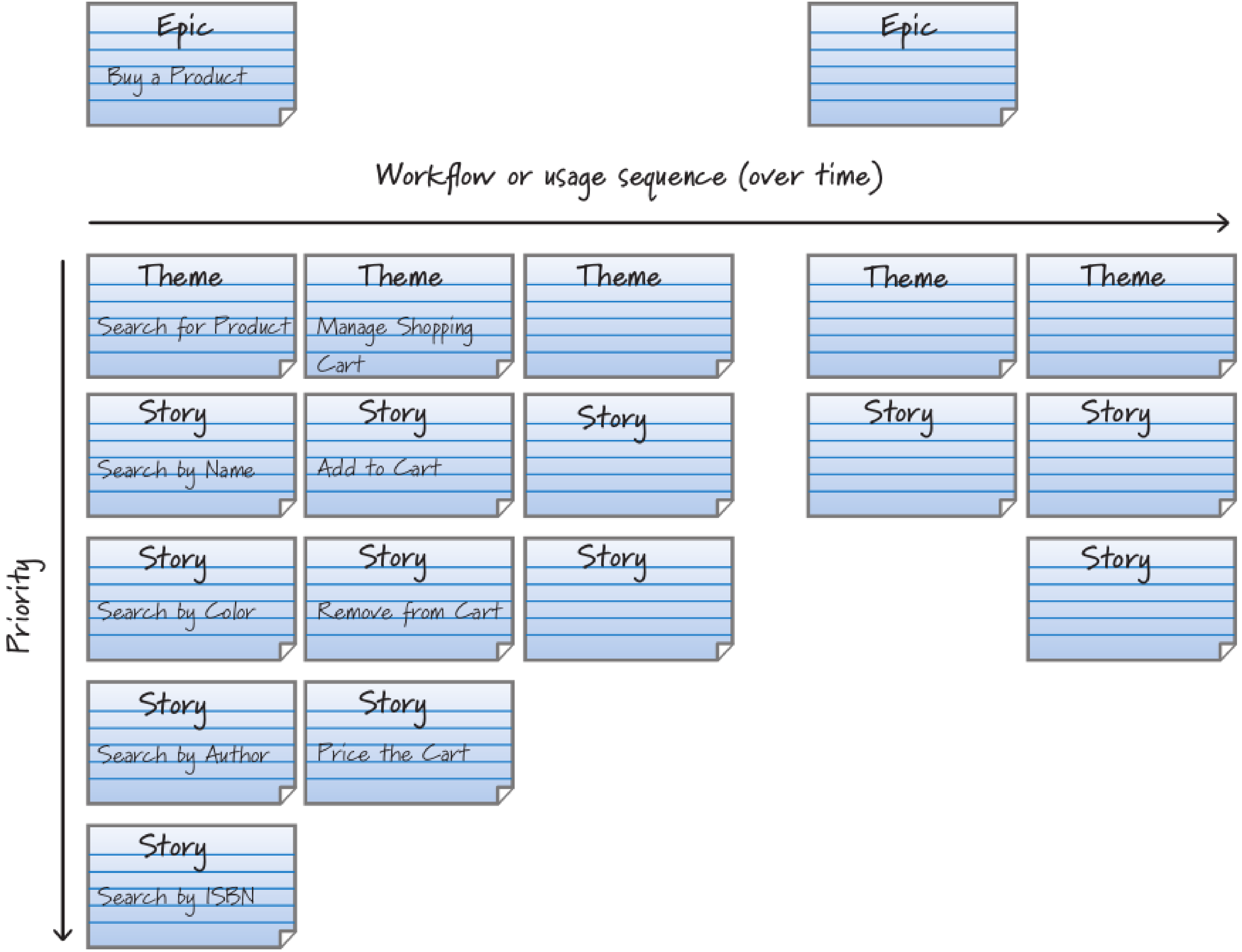
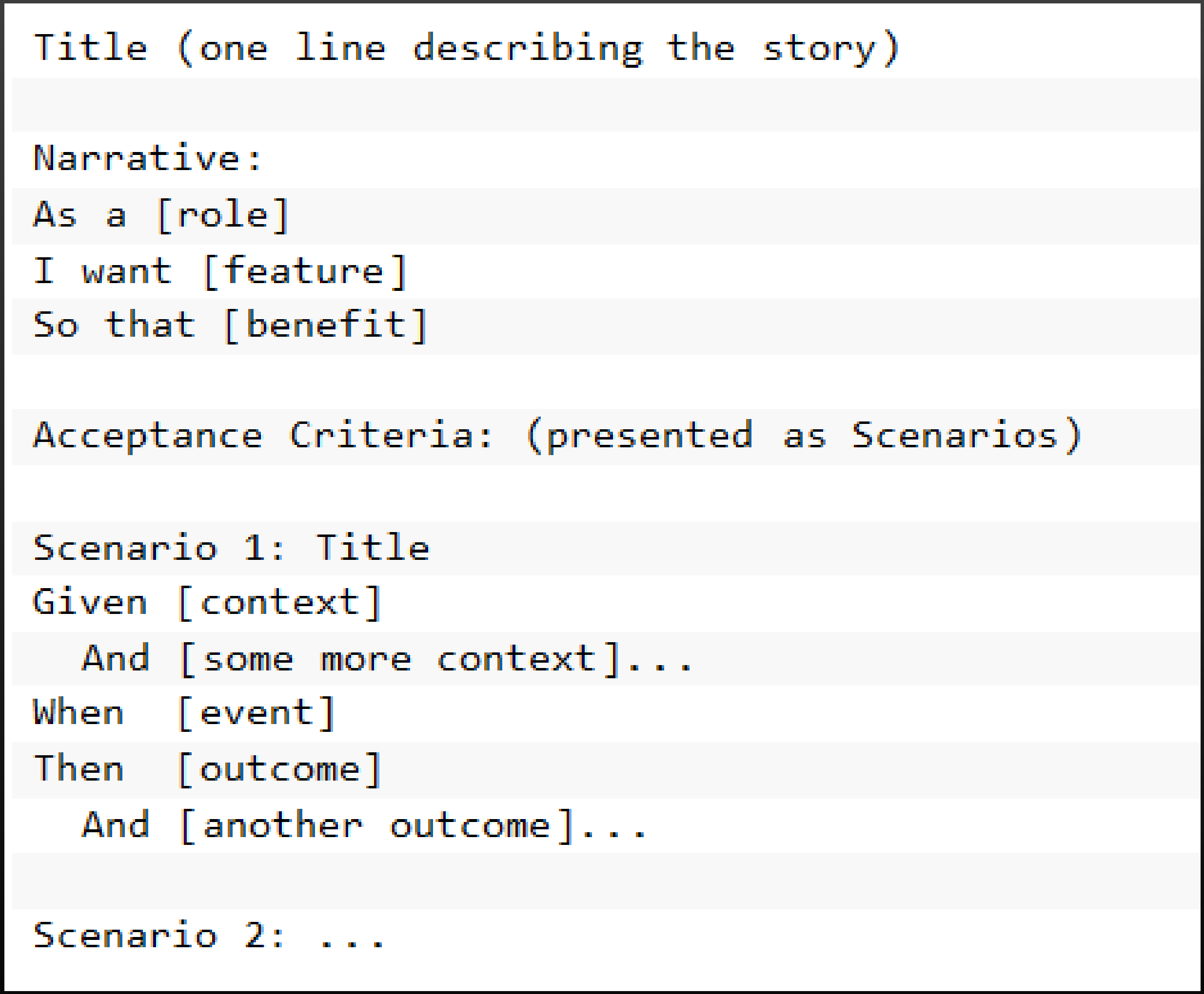


FIGURE 8:
THE RELATIONSHIP BETWEEN THE FLOWS AND THE STORIES

https://www.ivarjacobson.com/publications/white-papers/use-case-ebook


Epic
Buy a Product
Epic
Workflow or usage sequence (over time)
Priority
Theme
Search for Product
Theme
Manage Shopping Cart
Theme
Theme
Theme
Story
Search by Name
Story
Add to Cart
Story
Story
Story
Story
Search by Color
Story
Remove from Cart
Story
Story
Story
Search by Author
Story
Price the Cart
Story
Search by ISBN

A story and the tests...

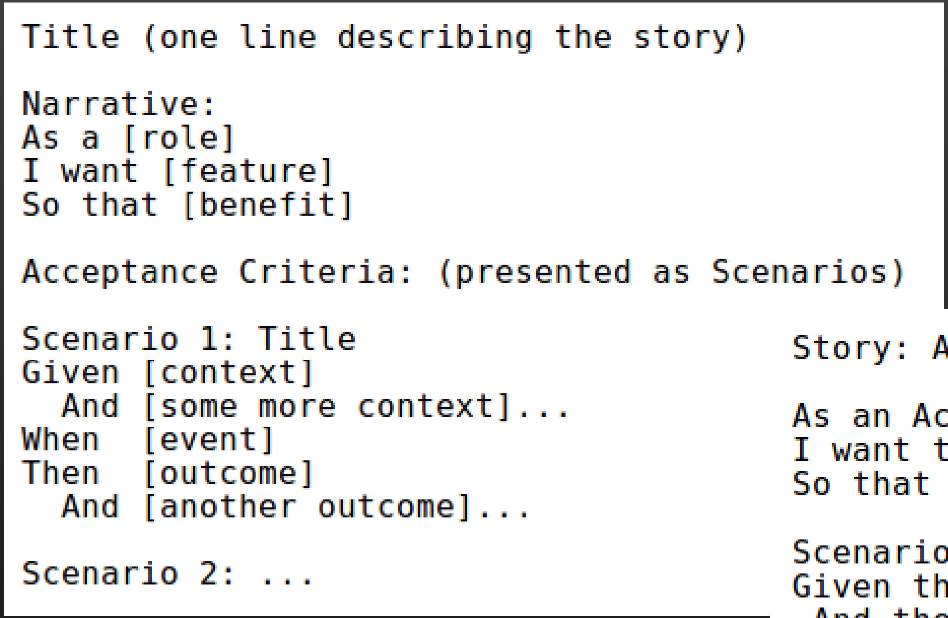


```
Title (one line describing the story)

Narrative:
As a [role]
I want [feature]
So that [benefit]

Acceptance Criteria: (presented as Scenarios)

Scenario 1: Title
Given [context]
  And [some more context]...
When  [event]
Then  [outcome]
  And [another outcome]...

Scenario 2: ...
```

Can we write the acceptance criteria in a way that it is <u>executable</u>?

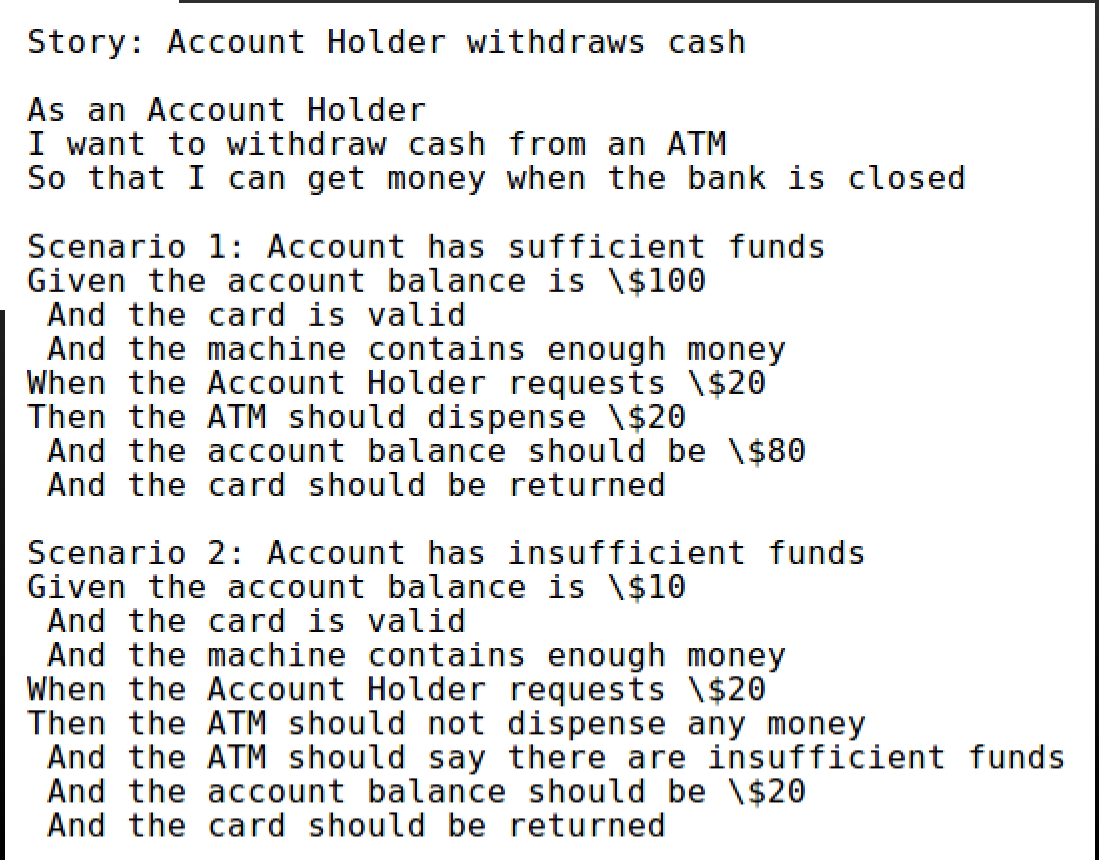
Oliveira (2017)

Story: the scope of a feature + its acceptance criteria.



```
Title (one line describing the story)

Narrative:
As a [role]
I want [feature]
So that [benefit]

Acceptance Criteria: (presented as Scenarios)

Scenario 1: Title
Given [context]
  And [some more context]...
When  [event]
Then  [outcome]
  And [another outcome]...

Scenario 2: ...
```

Functional view.
Value for the user.
Specification by examples.



```
Story: Account Holder withdraws cash

As an Account Holder
I want to withdraw cash from an ATM
So that I can get money when the bank is closed

Scenario 1: Account has sufficient funds
Given the account balance is \$100
 And the card is valid
 And the machine contains enough money
When the Account Holder requests \$20
Then the ATM should dispense \$20
 And the account balance should be \$80
 And the card should be returned

Scenario 2: Account has insufficient funds
Given the account balance is \$10
 And the card is valid
 And the machine contains enough money
When the Account Holder requests \$20
Then the ATM should not dispense any money
 And the ATM should say there are insufficient funds
 And the account balance should be \$20
 And the card should be returned
```

Credit: http://dannorth.net/whats-in-a-story/

Features are described in the Gherkin Language

Feature: Some terse yet descriptive text of what is desired

In order to realize a named business value

As an explicit system actor

I want to gain some beneficial outcome which furthers the goal

Scenario: Some determinable business situation

Given some precondition

And some other precondition

When some action by the actor

And some other action

And yet another action

Then some testable outcome is achieved

And something else we can check happens too

Scenario: A different situation

...

writing features - gherkin language ¶

BDD Given, When, Then style

Structured syntax (Gherkin) to describe a feature (for testing):

Feature: what

Scenario: some determinable business situation

Given: preparation/setup (e.g.: required data)

- And…

When: the set of actions (execute).

- And…

Then: specifies the expected resulting state (assert).

- And…

Sample

brainstorm

Section	
Scenario	
Given	
When	
Then	

Cucumber tool

Goal

common understanding of the problem ⇒ simplify the communication between all parties

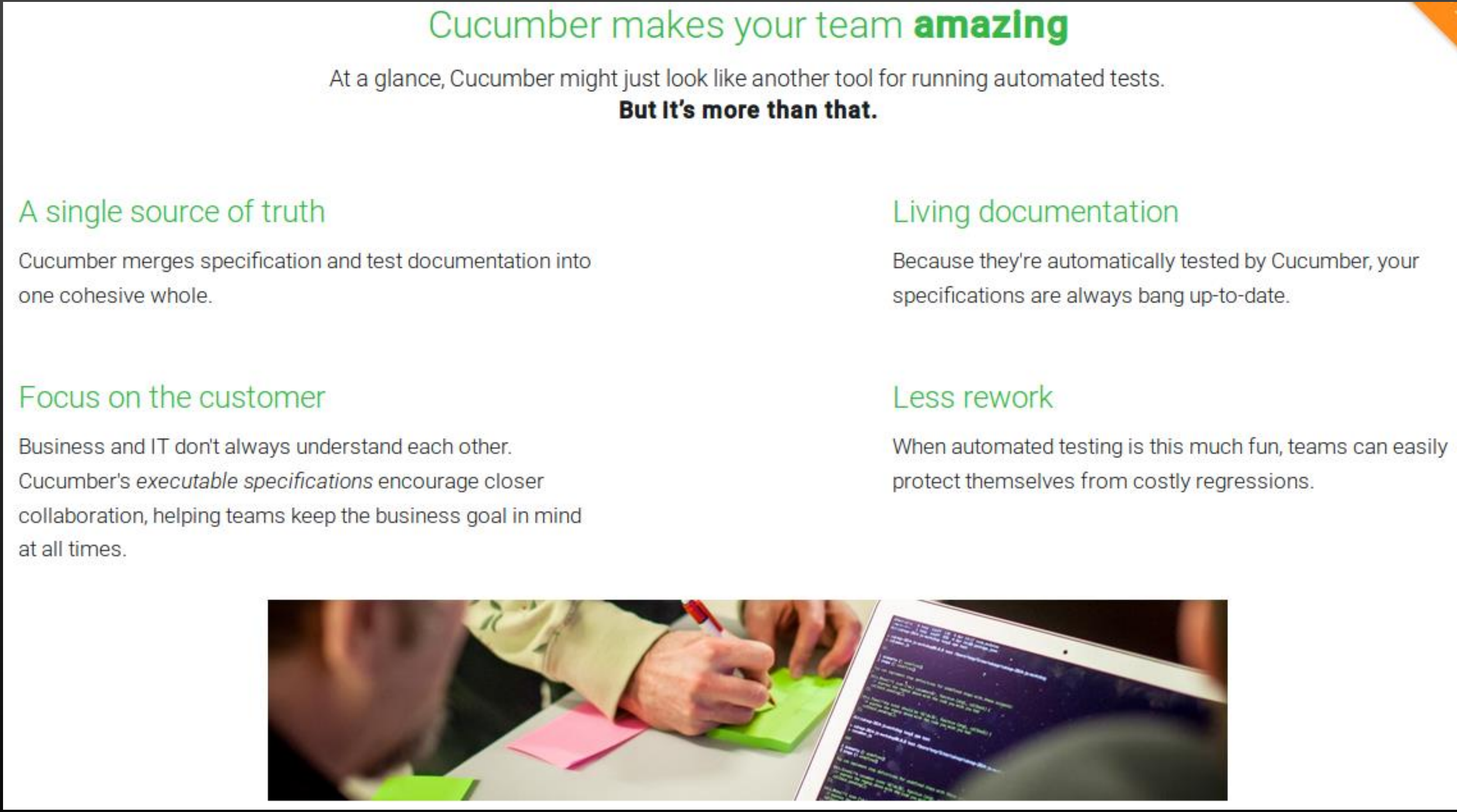
Cucumber way

express requirements using concrete examples

create examples of behavior that are executable

examples are found in a collaborative way (business analysts, testers and developers)

examples can be used as acceptance tests (with additional preparation steps)



Cucumber makes your team **amazing**

At a glance, Cucumber might just look like another tool for running automated tests.
But It's more than that.

A single source of truth

Cucumber merges specification and test documentation into one cohesive whole.

Living documentation

Because they're automatically tested by Cucumber, your specifications are always bang up-to-date.

Focus on the customer

Business and IT don't always understand each other. Cucumber's *executable specifications* encourage closer collaboration, helping teams keep the business goal in mind at all times.

Less rework

When automated testing is this much fun, teams can easily protect themselves from costly regressions.

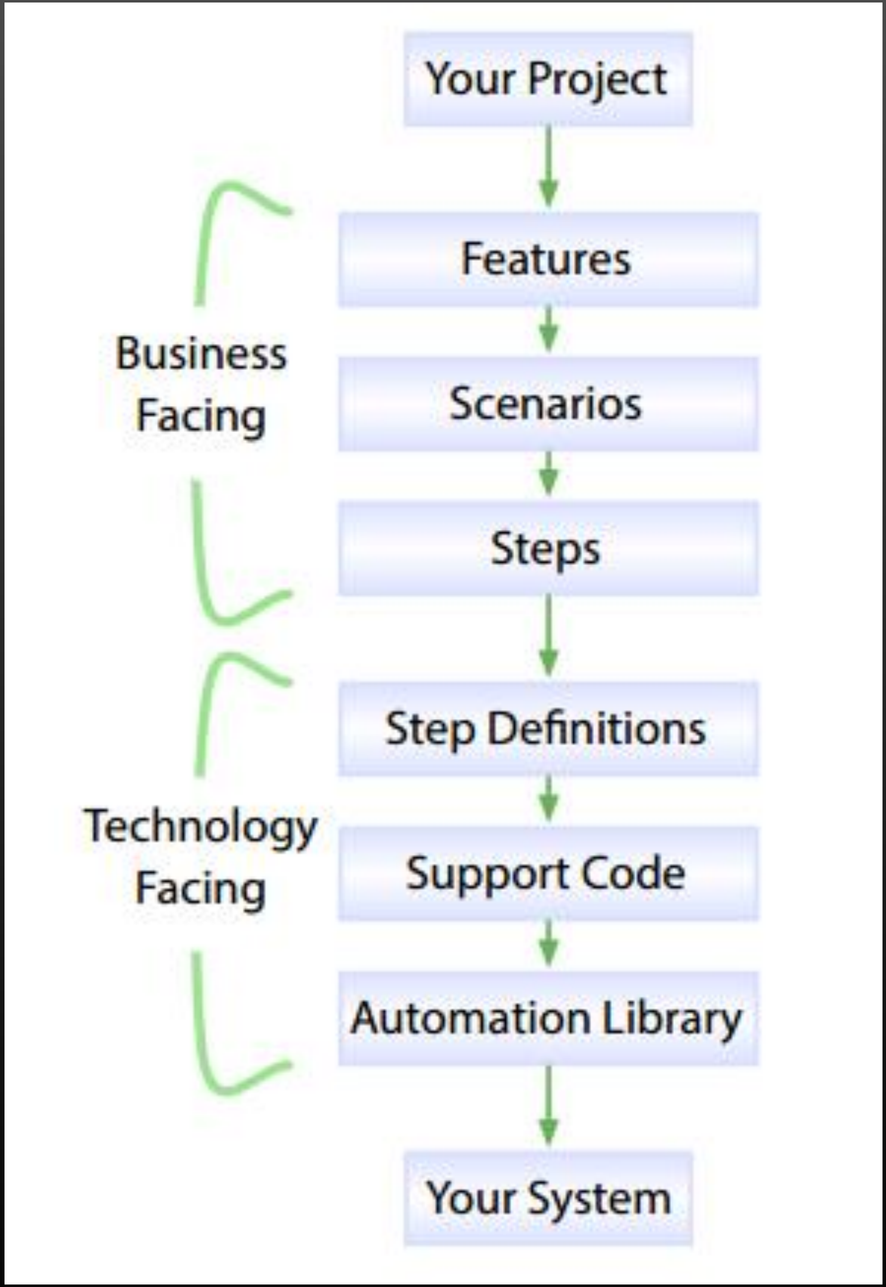
Cucumber reads specifications from plain-language text files called *features*, examines them for *scenarios* to test.

Each scenario is a list of *steps* for Cucumber to work through.

Along with the features, you give Cucumber a set of *step definitions*, which map the business-readable language of each step into code to carry out whatever action is being described by the step.

The step definition itself will probably just be one or two lines of code that delegate to a library of *support code*, specific to the domain of your application.

Sometimes that may involve using an *automation library*, like the browser automation library Selenium.



Implementing Cucumber in JVM

Cucumber is first and foremost a conversation tool.

The most important part is the conversations that must take place before we can implement something that our users want

Add a few features that will define our wanted behavior

The features must be located in the same package or a subpackage below the package where the runner is located.

in Maven, anything found in a directory called resources at the same level as the java directory, will be a part of the classpath

Cucumber can be executed using JUnit through a specific JUnit runner.

Feature - a short description of the feature. Try to express yourself in one sentence

Scenario - the actual business scenario that should be working

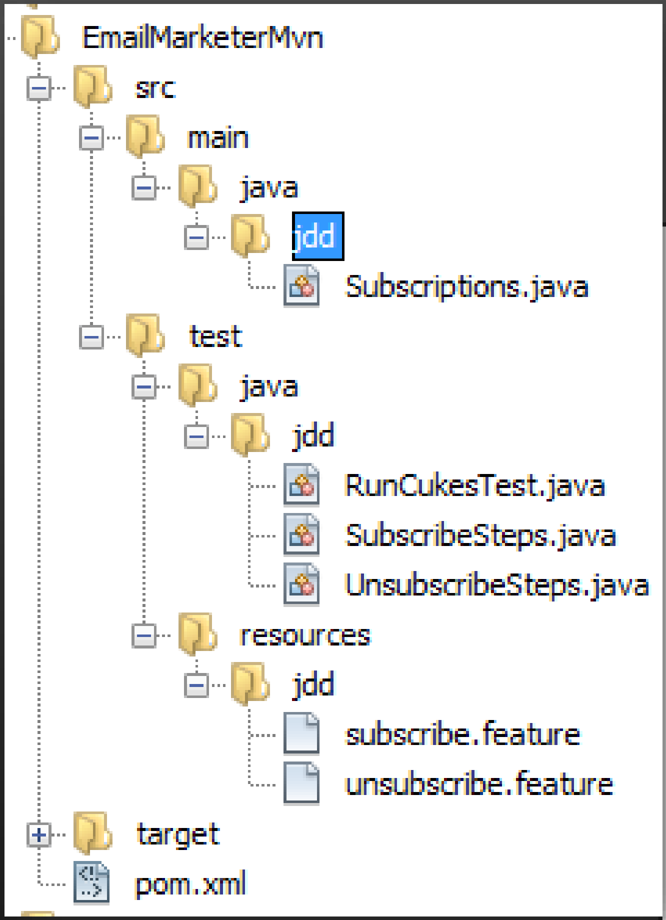
Given - the setup step. Define the preconditions for the wanted behavior

When - the execution step. This is where you use the system in some way

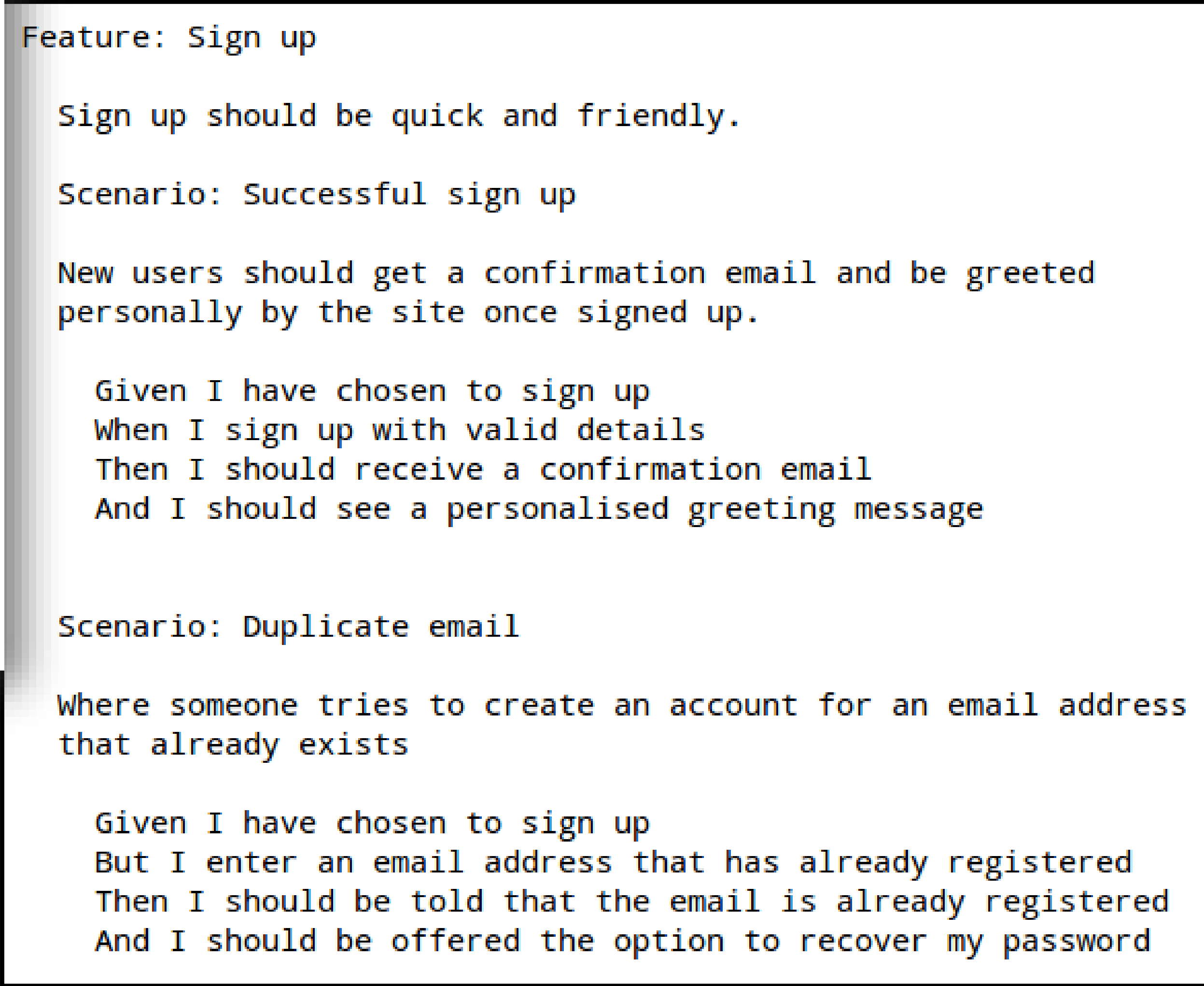
Then - the assertion step. This is where you observe the system and assert that the wanted change has occurred

I Oliveira (2017)

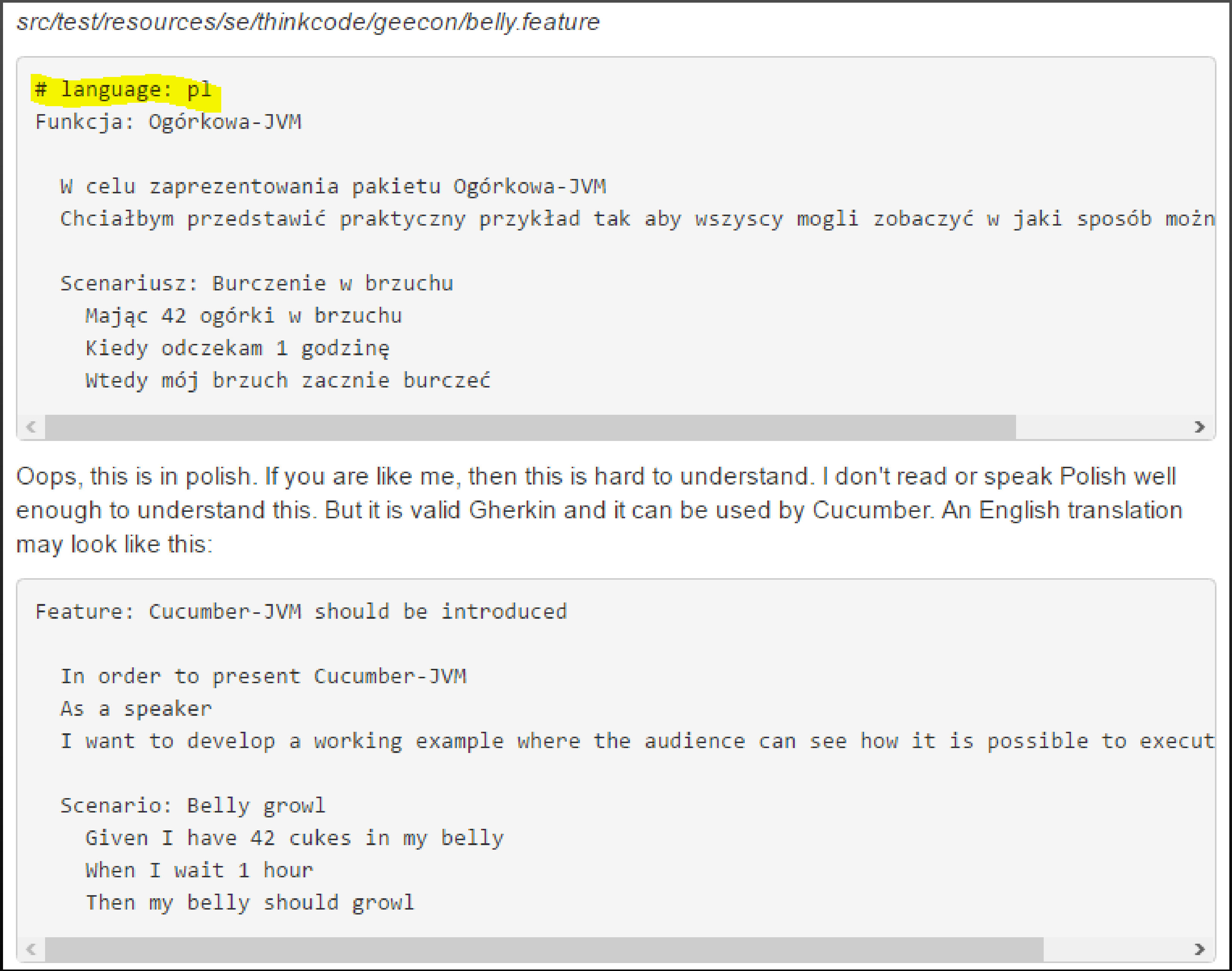
Hands-on: cucumber jvm



```
EmailMarketerMvn
  src
    main
      java
        jdd
          Subscriptions.java
    test
      java
        jdd
          RunCukesTest.java
          SubscribeSteps.java
          UnsubscribeSteps.java
      resources
        jdd
          subscribe.feature
          unsubscribe.feature
  target
  pom.xml
```



```
Feature: Sign up

  Sign up should be quick and friendly.

  Scenario: Successful sign up

  New users should get a confirmation email and be greeted
  personally by the site once signed up.

    Given I have chosen to sign up
    When I sign up with valid details
    Then I should receive a confirmation email
    And I should see a personalised greeting message


  Scenario: Duplicate email

  Where someone tries to create an account for an email address
  that already exists

    Given I have chosen to sign up
    But I enter an email address that has already registered
    Then I should be told that the email is already registered
    And I should be offered the option to recover my password
```



src/test/resources/se/thinkcode/geecon/belly.feature

```
# language: pl
Funkcja: Ogórkowa-JVM

  W celu zaprezentowania pakietu Ogórkowa-JVM
  Chciałbym przedstawić praktyczny przykład tak aby wszyscy mogli zobaczyć w jaki sposób możn

  Scenariusz: Burczenie w brzuchu
    Mając 42 ogórki w brzuchu
    Kiedy odczekam 1 godzinę
    Wtedy mój brzuch zacznie burczeć
```

Oops, this is in polish. If you are like me, then this is hard to understand. I don't read or speak Polish well enough to understand this. But it is valid Gherkin and it can be used by Cucumber. An English translation may look like this:

```
Feature: Cucumber-JVM should be introduced

  In order to present Cucumber-JVM
  As a speaker
  I want to develop a working example where the audience can see how it is possible to execut

  Scenario: Belly growl
    Given I have 42 cukes in my belly
    When I wait 1 hour
    Then my belly should growl
```

http://www.thinkcode.se/blog/2015/01/30/bdd-with-cucumberjvm-at-geecon-tdd-2015

Views from Robert C. Martin

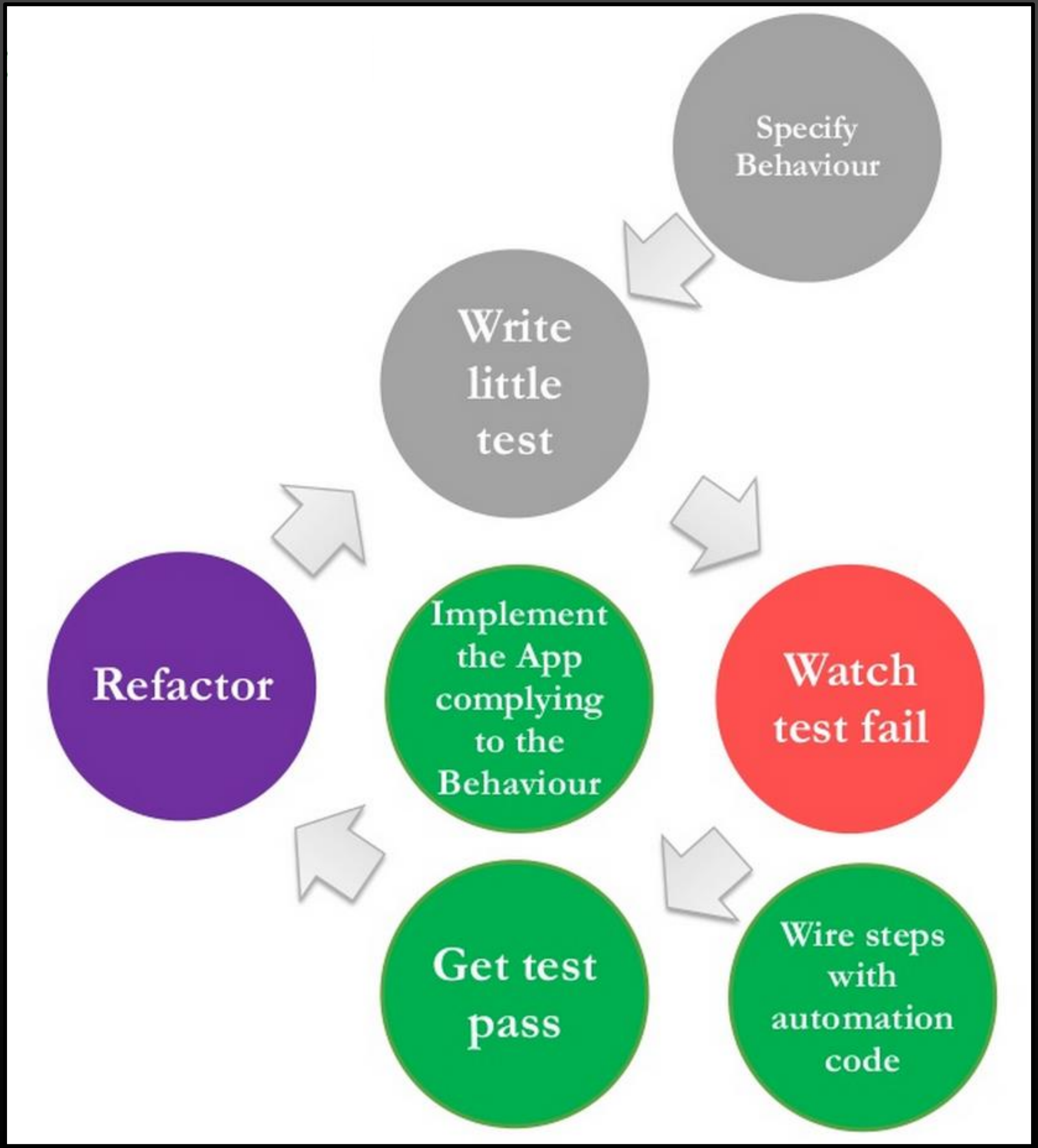
BDD is a variation on TDD.

Whereas in TDD we drive the development of a module by "first" stating the requirements as unit tests, in BDD we drive that development by first stating the requirements as, well, *requirements*.

The form of those requirements is fairly rigid, allowing them to be interpreted by a tool that can execute them in a manner that is similar to unit tests.

https://sites.google.com/site/unclebobconsultingllc/the-truth-about-bdd

BDD: Behaviour-driven development


Specify Behaviour
Write little test
Watch test fail
Wire steps with automation code
Get test pass
Implement the App complying to the Behaviour
Refactor

Credit: Nalin Goonawardana

Oliveira (2017)

BDD frameworks

Cucumber (Ruby framework)

Cucumber-JVM

Behat (PHP framework)

Fitness

...

Resources and readings

Sundberd, T., “Where should you use Behaviour Driven Development, BDD?”

Kops, “**BDD Testing with Cucumber, Java and Junit**”