Cucumber Session 1 - Practice 1 - Behavior Driven Development - Slide 19

Author: Yesica Acha

1. Research

When and why BDD was defined?

BDD was defined in 2008 by Dan North. It was defined to close the breach that exists between the developers, POs and Testers, and solve some problems. It was born from the weaknesses that TDD has.

The most important aspects of BDD

The most important aspects are:

* Its purpose is clearly related to business outcomes
* Thinking "from the outside in", in other words implement only those behaviors which contribute most directly to these business outcomes, so as to minimize waste.
* Describes behaviors in a single notation which is directly accessible to domain experts, testers and developers, so as to improve communication
* Apply these techniques all the way down to the lowest levels of abstraction of the software, paying particular attention to the distribution of behavior, so that evolution remains cheap

Why is useful and when could be applied?

It is useful because it is:

* Collaborative => All the members of the team (devlopers, testers, stakeholders, POs) can work together to review the scenario.
* Simple specification => since it uses Ubiquitous language, it is a common language and all the members of the team can understand.
* Easy feedback => All the members of the team can understand the scenario because it is written in a common language, and it is easy to get feedback from all the members of the team
* Avoid rework in requirements definitions
* Self-documentation => tools targeting a BDD approach generally afford the automatic generation of technical and end user documentation from BDD "specifications"
* Focus on behavior => thinking "from the outside in", in other words implement only those behaviors which contribute most directly to these business outcomes, so as to minimize waste

Which type of teams could implement BDD?

Any team could implement BDD.

2. Structure of a feature - Give an example

A feature should have the following structure: name, size (optional) and Priority

e.g.

Name: Create Users

3. Structure of a user story - Give an example

A user story should have the following structure:

* Title (one line describing the story)
* Narrative:
  + As a [role]
  + I want [feature]
  + So that [benefit]

e.g.

As a user

I can indicate folders not to backup

So that my backup drive isn't filled up with things I don't need saved.

4. Structure of a scenario - Give an example

A scenario should have the following structure:

* Given some initial context (the givens)
* When an event occurs
* then ensure some outcomes

e.g.

Scenario: Buy a coffee

Given there are 1 coffees left in the machine

And I have deposited 1$

When I press the coffee button

Then I should be served a coffee

5. Differences between BDD and BDT

BDT is the testing perspective on BDD. BDT are Given-when-then style tests written in natural language which are easily understandable to non-technical individuals. Hence these tests allow business analysts and management people to actively participate in test creation and review process.