



DigitasLBi

BDD, Behat and Mink
02/20/2014

Behaviour-Driven Development is a method
that ***establishes a list of user behaviour*** for a
given product

Behaviour-Driven Development is a method that ***establishes a list of user behaviour*** for a given product

How can we ***easily verify*** that these ***behaviors*** are ***valid*** and ***are not affected by new*** behaviors?

AGENDA



- ✦ Behaviour-Driven Development
- ✦ Behat
- ✦ Mink
- ✦ Outcome



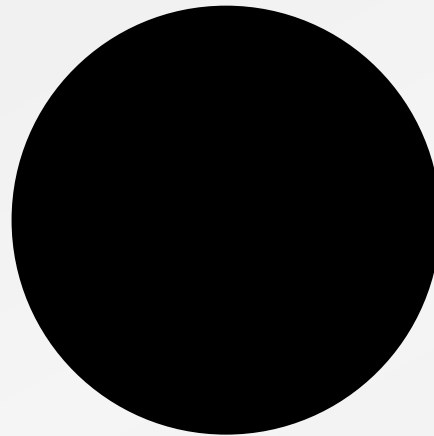
BDD.

Behaviour-Driven Development

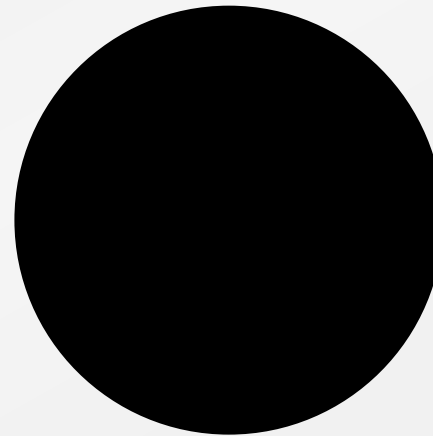
PROJECT - PHILOSOPHY



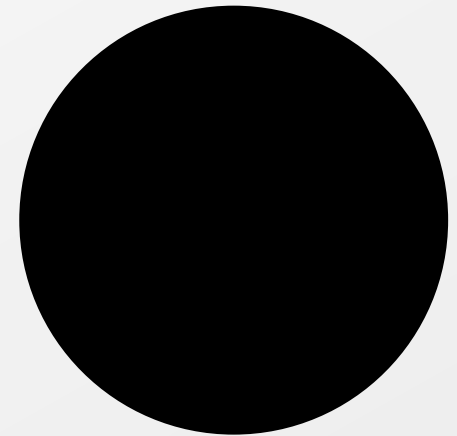
Product Owner



Func



Tech



BDD - PHILOSOPHY

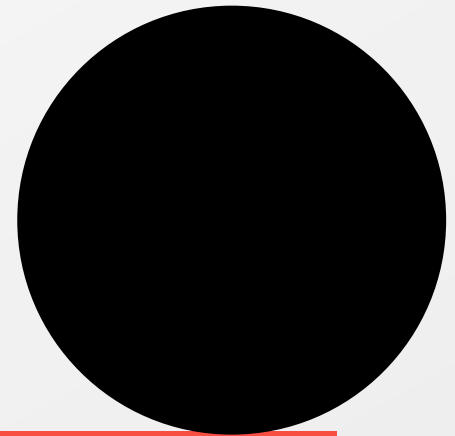
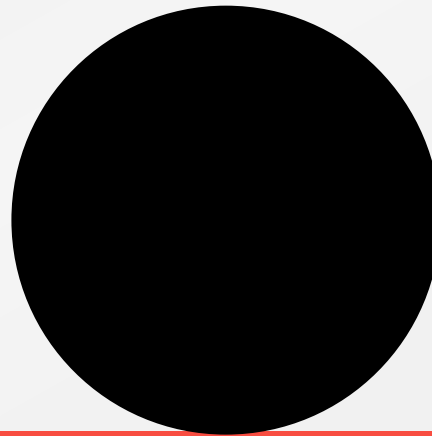
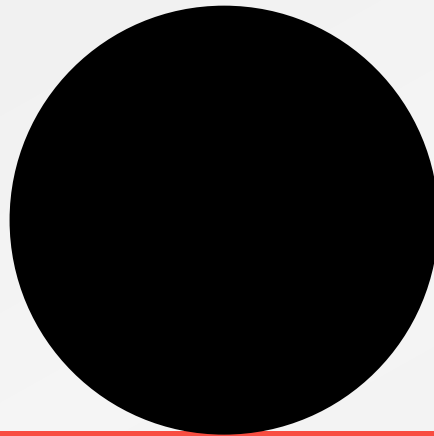
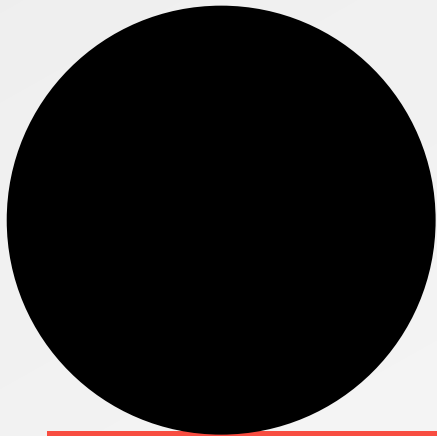


User

Product Owner

Func

Tech



Speak Same Language



BDD - WORKFLOW



4 Steps

- ❖ **Define** the business value of the feature => Product Owner
- ❖ **Prioritize** features by their business value => Product Owner
- ❖ **Describe** them with readable scenarios => Func
- ❖ And then, only then – **implement** them => Thec

BDD – GHERKIN



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

...



BDD – Feature Language



Gherkin: 3 levels of description

- ❖ 1 - Narrative
- ❖ 2 - Possible scenarios
- ❖ 3 - Scenarios details

BDD – Feature Language



Gherkin: 3 levels of description



1 -

```
2   Feature: Authorization
3       In order to get an exclusive access to informations
4       As an administrator
5       I need to be able to authenticate
6
```



2 - Possible scenarios



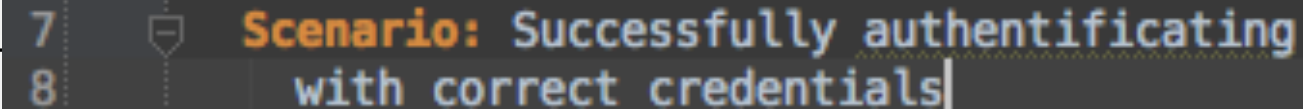
3 - Scenarios details

BDD – Feature Language



Gherkin: 3 levels of description

✦ 1 - Narrative

✦ 2 -  **Scenario:** Successfully authenticating
with correct credentials

✦ 3 - Scenarios details

BDD – Feature Language



Gherkin: 3 levels of description

- ✦ 1 - Narrative
- ✦ 2 - Possible scenarios

✦ 3

```
9  Given I am on "login" page
10 When I fill form with "stephane" login
11 Then I should see "stephane" informations
12
```



Features

that

Validate

our app



*Let's
Automate
Them*


Behat.

“A php Framework for testing your business expctations”

Behat - Installation




Getting started with Behat is as easy as

 vim composer.json

```
{
  "require": {
    "behat/behat": "2.4.*@stable"
  },
  "config": {
    "bin-dir": "bin/"
  }
}
```

```
curl http://getcomposer.org/installer | php
php composer.phar install --prefer-source
```

 bin/behat --init
vim features/your_first.feature

```
Feature: Your first feature
  In order to start using Behat
  As a manager or developer
  I need to try
```

```
Scenario: Successfully describing scenario
  Given there is something
  When I do something
  Then I should see something
```

```
bin/behat
```

Source: <http://behat.org/>

Behat - exercices



Case 1 : php script that calculate age

Source: <http://behat.org/>

Mink.

“Web Acceptance Testing”

Mink - Installation



```
{  
  "require": {  
    "behat/mink": "*",  
  
    "behat/mink-goutte-driver": "*",  
    "behat/mink-selenium-driver": "*",  
    "behat/mink-selenium2-driver": "*",  
    "behat/mink-sahi-driver": "*",  
    "behat/mink-zombie-driver": "*"   
  }  
}
```

Source: <http://behat.org/>

Mink - Installation



Need A Browser Emulator

Configuration File

Add new basic sentence

Source: <http://behat.org/>

Mink - exercices



Case 2 : Access secured area on Digex

Source: <http://behat.org/>

Mink - exercices



Case 3 : Facebook moderation

Source: <http://behat.org/>

So?

Best Practices.

Best Practices



1 – Use Behat with BDD

Best Practices



- ❖ 1 – Use Behat with BDD
- ❖ 2 – Sentences refactoring rather than code refactoring

Best Practices



- ❖ 1 – Use Behat with BDD
- ❖ 2 – Sentences refactoring rather than code refactoring
- ❖ 3 – Scenario abstraction Level

Best Practices



- ❖ 1 – Use Behat with BDD
- ❖ 2 – Sentences refactoring rather than code refactoring
- ❖ 3 – Scenario abstraction Level
- ❖ 4 – Write simple Scenario

Best Practices



- ❖ 1 – Use Behat with BDD
- ❖ 2 – Sentences refactoring rather than code refactoring
- ❖ 3 – Scenario abstraction Level
- ❖ 4 – Write simple Scenario
- ❖ 5 - Use subcontext, hook...

USEFUL LINKS

<http://blog.lepine.pro/wp-content/uploads/2012/04/behat-cheat-sheet1.pdf>

<https://knpuniversity.com/screencast/behat>

<http://blog.lepine.pro/?s=behat>

<http://behat.org/>

<http://www.purepixel.co.uk/2013/02/using-selenium-server-2-and-phpunit-to-automate-browser-acceptance-testing/>

