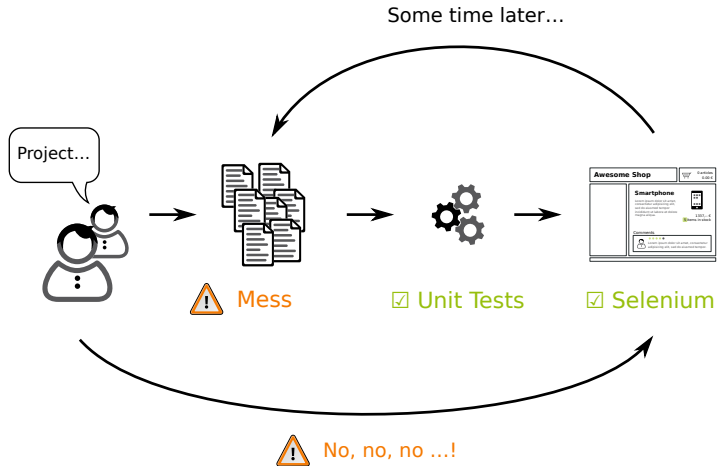


Behave! - BDD

DPC 2014

Tobias Schlitt (@tobySen)
2014-06-26

How a Project Typically Works



Behavior Driven Development (BDD)

- ▶ Executable specification
- ▶ Combines
 - ▶ Test Driven Development
 - ▶ Business communication
- ▶ Common language
 - ▶ Business people
 - ▶ Analysts
 - ▶ Developers
 - ▶ Testers

Outline

Behat

Mink

Conclusion

Behat

- ▶ BDD test framework for PHP
- ▶ Inspired by Ruby's Cucumber
- ▶ Work with Gherkin language framework
- ▶ <http://behat.org>

Gherkin

- ▶ Domain Specific Language (DSL) framework
- ▶ Goal: Business readable
- ▶ Create custom DSL on Gherkin
- ▶ Tie test code to DSL sentences

Gherkin in Action

1 **Feature:** DateTime proceeds to end of month.
2
3 **Scenario:** End of February is reached
4 **Given** it is date "2013-02-01"
5 **When** I proceed to the end of the month
6 **Then** the date is "2013-02-28"

Gherkin keywords

- ▶ **Given**
 - ▶ Precondition
 - ▶ Test setup
- ▶ **When**
 - ▶ Behavior trigger
 - ▶ Test stimulus
- ▶ **Then**
 - ▶ Expected behavior
 - ▶ Describe observable result
- ▶ **And / But**
 - ▶ Readability sugar

Binding Code to Sentences

```
1 Feature: DateTime proceeds to end of month.  
2  
3     Scenario: End of February is reached  
4         Given it is date "2013-02-01"  
5         When I proceed to the end of the month  
6         Then the date is "2013-02-28"  
  
1 class ExampleContext extends BehatContext  
2 {  
3 }
```

Binding Code to Sentences

1 **Feature**: DateTime proceeds to end of month.

2
3 **Scenario**: End of February is reached

4 **Given** it is date "2013-02-01"

5 **When** I proceed to the end of the month

6 **Then** the date is "2013-02-28"

```
1 class ExampleContext extends BehatContext
2 {
3     protected $date;
4
5     /**
6      * @Given /^ it is date "([^"]*)"$/
7      */
8     public function thereIsTheDate($date)
9     {
10         $this->date = new \DateTime($date);
11     }
12 }
```

Binding Code to Sentences

1 **Feature:** DateTime proceeds to end of month.

2

3 **Scenario:** End of February is reached

4 **Given** it is date "2013-02-01"

5 **When** I proceed to the end of the month

6 **Then** the date is "2013-02-28"

1 **class** ExampleContext **extends** BehatContext

2 {

3 /**

4 * @When /^ I proceed to the end of the month\$/

5 */

6 **public function** iProceedToTheEndOfTheMonth()

7 {

8 \$this->date->modify("last day of this month");

9 }

10 }

Binding Code to Sentences

1 **Feature**: DateTime proceeds to end of month.

2
3 **Scenario**: End of February is reached

4 **Given** it is date "2013-02-01"

5 **When** I proceed to the end of the month

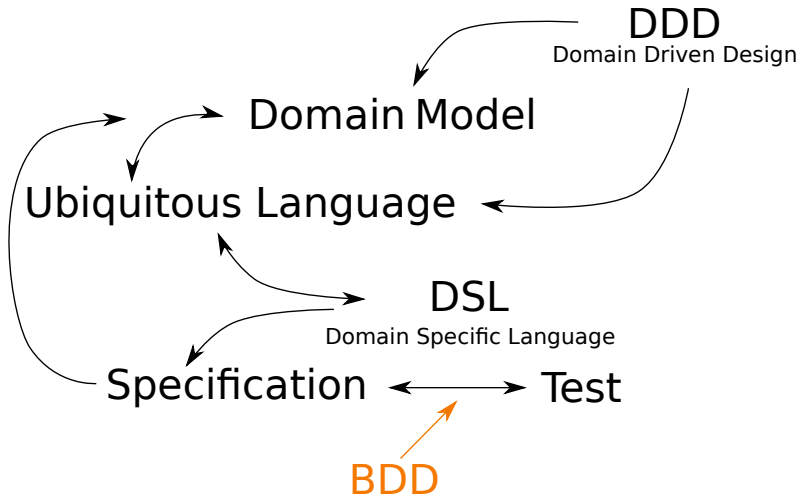
6 **Then** the date is "2013-02-28"

```
1 class ExampleContext extends BehatContext
2 {
3     /**
4      * @Then /^the date is "(?P<date>[^\"]*)"$/
5      */
6     public function theDateIsNow($date)
7     {
8         \PHPUnit_Framework_Assert::assertEquals(
9             $date,
10            $this->date->format("Y-m-d")
11        );
12    }
13 }
```

Using Behat

- ▶ `$ path/to/behat [.phar]`
Command for Behat usage
- ▶ `$ behat --init`
Initialize global test infrastructure
- ▶ `$ behat --init @SomeNiceBundle`
Initialize bundle test infrastructure
- ▶ `$ behat -dl`
Display known sentences
- ▶ `$ behat path/to/features`
Execute feature tests in directory
- ▶ `$ behat @SomeNiceBundle`
Execute feature tests of bundle

Behavior Driven Development



Outline

Behat

Mink

Conclusion

- ▶ Web acceptance test framework
- ▶ Abstracts browser emulations / controllers
 - ▶ Goutte
 - ▶ Zombie.js
 - ▶ Selenium / Selenium 2
 - ▶ Sahi
- ▶ Mink extension for Behat provides sentences
- ▶ <http://mink.behat.org>

Behat Mink Example

1 **Feature:** Browse Wikipedia

2
3 **Scenario:** Search front page

4 **Given** I am on "/"

5 **When** I fill in "searchInput" with "Kore"

6 **And** I press "searchButton"

7 **Then** I should see "Kore may refer to:"

8
9 **Scenario:** Follow redirect link

10 **Given** I am on "/"

11 **When** I fill in "searchInput" with "Kore"

12 **And** I press "searchButton"

13 **And** I follow "Kore (energy drink)"

14 **Then** the response status code should be 200

Mink Behat-extension

- ▶ Mink integration for Behat
- ▶ Pre-build sentences to browse pages
- ▶ Extensible with custom sentences
- ▶ `http://extensions.behat.org/mink`

Attention!

- ▶ This is no more BDD!
 - ▶ Too technical
 - ▶ Not bound to the model
 - ▶ Fragile for view changes
- ▶ However
 - ▶ Useful for system tests
 - ▶ Awesome for refactoring!

http://bit.ly/behav_cc

BDD through the Front-End

```
1  /**
2   * @Given /^ I am logged in as "([^"]*)"$/
3   */
4   public function iAmLoggedInAs($user)
5   {
6       $this->currentUsername = $user;
7
8       return array(
9           new Step\Given(' I am on "/"'),
10          new Step\When(' I fill in "_username" with "' .
11              $user . '"'),
12          new Step\When(' I fill in "_password" with "test"'),
13          new Step\When(' I press "Login"'),
14      );
15  }
```

Outline

Behat

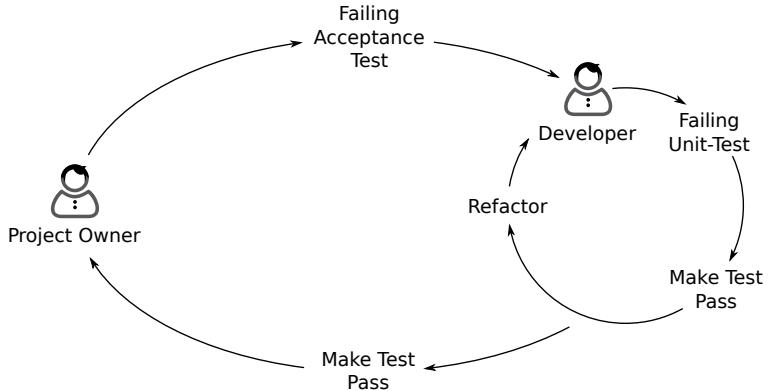
Mink

Conclusion

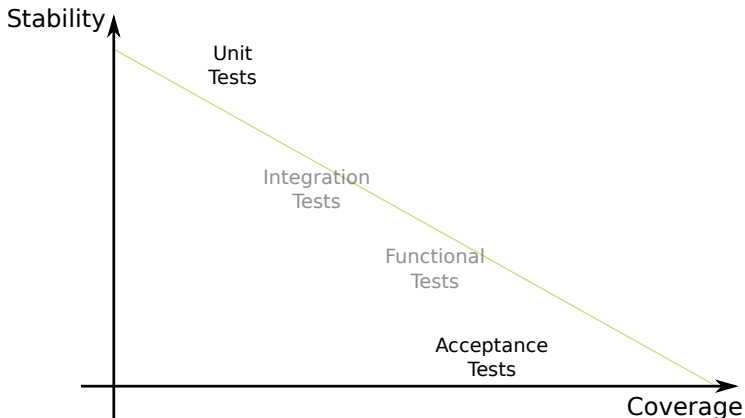
Unit Testing?

- ▶ The hype is over
- ▶ Still important for
 - ▶ Complexity
 - ▶ High impact code
 - ▶ Libraries

Double Feedback Test Cycle



Test Mix Aspects



Conclusion

- ▶ BDD makes specification executable
 - ▶ Awesome for communication
 - ▶ Awesome to abuse for front-end testing (Mink)
 - ▶ No low hanging fruit
 - ▶ Easy to fail
- ▶ Establish individual test mix for your team

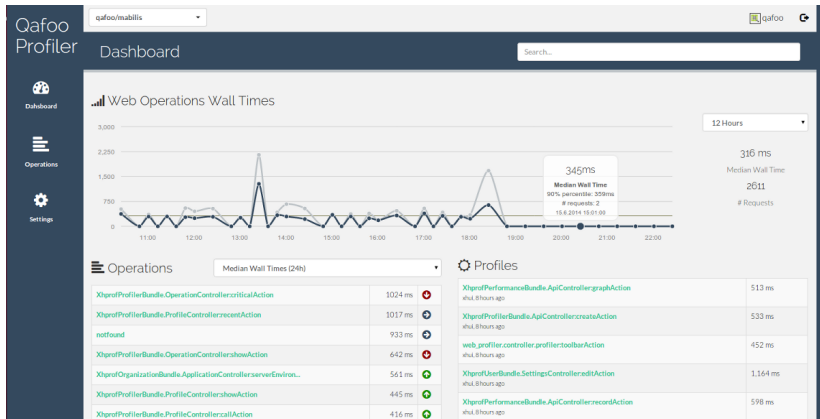
We are



Helping people to create high quality web applications.

<http://qafoo.com>

Qafoo Profiler



<http://qa.foo/profiler>

<https://joind.in/10870>



THANK YOU

Rent a quality expert
qafoo.com