

Behave! - Behavior Driven Development

IPC 2013 SE

Tobias Schlitt (@tobySen)
June 5th 2013

About me

- ▶ Tobias Schlitt (**Toby**)
- ▶ Degree in computer science
- ▶ Professional PHP since 2000
- ▶ Open source enthusiast
- ▶ Passion for
 - ▶ Software Design
 - ▶ Automated Testing

Co-founder of



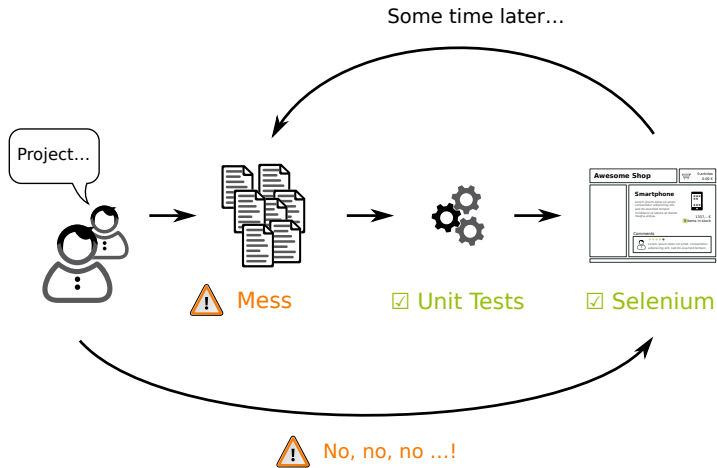
Helping teams to create high quality web applications.

<http://qafoo.com>

- ▶ Expert consulting
- ▶ Individual training

Get your team bootstrapped with BDD effectively!

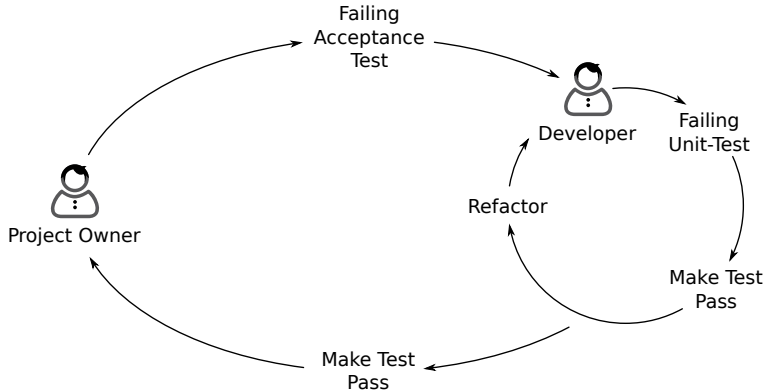
How to do a project



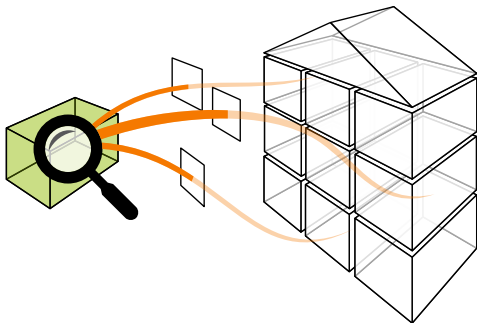
Behavior Driven Development (BDD)

- ▶ Make specification executable
- ▶ Combines:
 - ▶ Test Driven Development
 - ▶ Business focus
- ▶ Uses common language for
 - ▶ Business people
 - ▶ Analysts
 - ▶ Developers
 - ▶ Testers

Double Feedback Test Cycle



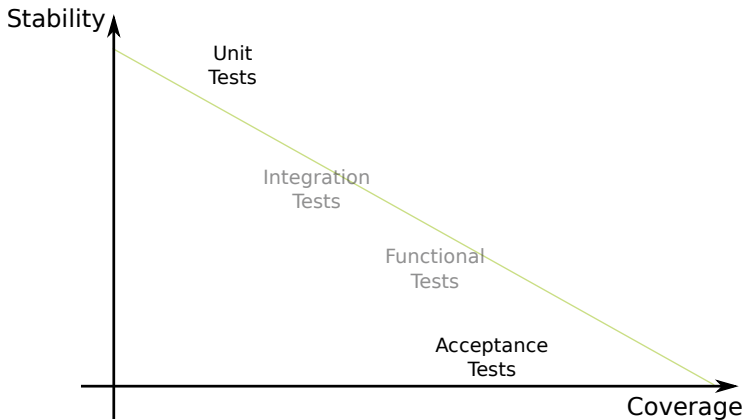
Unit Testing



Unit tests?

- ▶ Losing importance
- ▶ Still valid for
 - ▶ Libraries
 - ▶ Complex business logic
 - ▶ High impact code

Instability vs. Coverage



Outline

Behat

Mink

Tool Stack Overview

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,

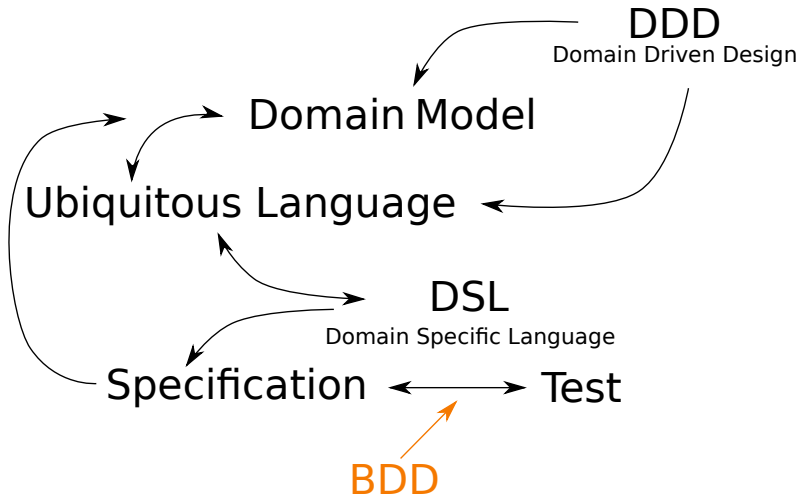
10 \$this->date->format("Y-m-d")

11);

12 }

13 }

Behavior Driven Development



Outline

Behat

Mink

Tool Stack Overview

- ▶ 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  }
```

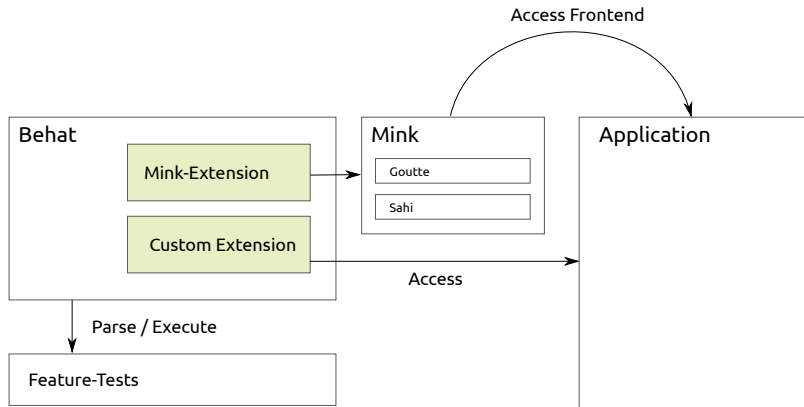
Outline

Behat

Mink

Tool Stack Overview

Behat tool stack



Conclusion

- ▶ BDD tries to make specification executable
 - ▶ DDD context
 - ▶ No low hanging fruit
 - ▶ Easy to fail
- ▶ Behat is a tool for BDD in PHP
- ▶ Behat + Mink = Awesome for refactoring



THANK YOU

Rent a quality expert
qafoo.com