Behaviour-Driven Development

Short overview for DSS/ReSA Group
Premek, 10.12.2012
Behaviour-Driven Development

Short overview for DSS/ReSA Group
Premek, 10.12.2012
BDD in 5 minutes

Agile specifications
+ examples
+ automated tests
⇒ software as narrative

Dan North (thoughtworks.com)
2003, 2006
http://dannorth.net/introducing-bdd/
Testing

- we know we should do it
- we don’t like it that much
- code → tests doesn’t really help
Test-Driven Development

- helps reduce bias
- drives API and modular design
- but ...

How do I know what to test? Would my PO read the tests?
How do we write requirements

- IEEE Std 830
- DOD-STD-2167A
- Use cases + model
- User stories
- Talk
- Give examples

How do you create tests off that?
Tests x Requirements

Test::Unit to RSpec

class DogTest
def test_bark
  assert xxx
end

def setup
  s/test/should/
end

def teardown
  communication
end
describe Dog
  it “should bark”
  object.should be xxx
  before(:each) {}  
  after(:each) {}
Behaviour-Driven Development

Short overview for DSS/ReSA Group
Premek, 10.12.2012
Process
- describe features
- write stories
- decompose to scenarios
- write tests for them
- fail
- write/refactor code
- pass
- iterate

BDD basics

Tools
- Specification language
- code annotations
- testing framework
Describe features

= what users care about
= have business value
= can be prioritized, planned

Feature: Cucumber
In order to share the love
As a presenter
I will demonstrate behavior driven development
Write stories for features

User expectations
= how the system should behave
= “requirements”

A Story Structure

- Narrative
  - As a [some_role_here]
  - I want [some_feature_here]
  - So that [benefit/value_of_the_feature_here]

- Scenario(s) (this is acceptance criteria)
  - Given [some_initial_condition(s)]
  - When [event(s) occurs]
  - Then [ensure_some_outcome(s)]

Story 28 - View patient details

In order to choose the most suitable gas
an Anaesthetist
wants to view the Patient’s surgical history

http://dannorth.net/whats-in-a-story/
Decompose stories into scenarios

System specification = acceptance criteria

A Story Structure

- Narrative
  - As a [some_role_here]
  - I want [some_feature_here]
  - So that [benefit/value_of_the_feature_here]

- Scenario(s) (this is acceptance criteria)
  - Given [some_initial_condition(s)]
  - When [event(s)_occurs]
  - Then [ensure_some_outcome(s)]

Scenario - existing patient with history

Given we have a patient on file
And the patient has had previous surgery
When I request the Patient's surgical history
Then I see all the previous treatments
import org.jbehave.core.annotations.Alias;

public class CarparkSteps {

    Parking carpark;
    int slotsStart;
    int slotsEnd;
    int carsIn;
    RuntimeException ex = null;

    public static int CAPACITY_DEFAULT = 10;

    @Given("a carpark with \<before\> empty car slots")
    @Alias("a carpark with $before empty car slot")
    public void givenCarpark(@Named("before") int before) {
        carpark = new Parking(before);
        slotsStart = before;
    }

    @Given("a new carpark")
    @Aliases(values={"a carpark with free slots",
        "a carpark with full capacity free"})
    public void newCarPark() {
        carpark = new Parking(CAPACITY_DEFAULT);
        slotsStart = carpark.getCapacity();
    }

    @Given("a carpark with 1 free slot")
    public void carparkLastSlotFree() {
        carpark = new Parking(3);
        carpark.arrive();
        carpark.arrive();
    }
}
Run via xUnit framework

```
iampersistent@iampersistent-desktop:~/workspace/behat$ behat features
Feature: Addition
   In order to avoid silly mistakes
   As a math idiot
   I want to be told the sum of two numbers

Scenario: Add two numbers
    Given I have entered 50 into the calculator
    And I have entered 70 into the calculator
    When I press add
    Notice: Undefined index: file
    Then the result should be 120 on the screen

1 scenario (1 failed)
4 steps (2 passed, 1 skipped, 1 failed)
0.072s
```
**Feature:** User logins
In order to have extended abilities
As a site user
I need to be able to login

**Background:**
*Given* a site have users:

<table>
<thead>
<tr>
<th>username</th>
<th>password</th>
</tr>
</thead>
<tbody>
<tr>
<td>everzet</td>
<td>qwerty</td>
</tr>
</tbody>
</table>

**Scenario Outline:** Only existing users can login

*Given* I am on the “/login” page
*When* I fill in “username” with “<username>”
*And* I fill in “password” with “<password>”
*And* I press “login” in login form
*Then* I should see “<message>”

**Examples:**

<table>
<thead>
<tr>
<th>username</th>
<th>password</th>
<th>message</th>
</tr>
</thead>
<tbody>
<tr>
<td>everzet</td>
<td>qwerty</td>
<td>Welcome, everzet</td>
</tr>
<tr>
<td>someone</td>
<td>pa$$word</td>
<td>Login or password is incorrect</td>
</tr>
</tbody>
</table>
BDD Languages

One vocabulary for
... analysts
/ business
... designers
... testers
... programmers

http://martinfowler.com/bliki/BusinessReadableDSL.html
BDD Tools

http://wiki.kiv.zcu.cz/PremekBrada/ToolsForBDD

JBehave

live example

Selenium etc integration

=>

functional tests
Closing thoughts

The 2nd V of V&V
Close to Executable Specification

"Behaviour-driven development is about implementing an application by describing its behaviour from the perspective of its stakeholders."  [Dan North]

Widespread use? Research potential?
BDD Process

Customer  Domain Expert  QA  Developer

Stories → Behaviour → Application

Matches to

Exercise expectations against
Resources

Dan North (author)
“How to sell BDD to the business”
“Introducing BDD”: http://dannorth.net/introducing-bdd/

BDD with Behat
http://www.slideshare.net/IamPersistent/bdd-with-behat

Vasilev: BDD with Java
http://www.slideshare.net/shadrik/bdd-with-java-8323915

Kudryashov: BDD in Symphony
http://www.slideshare.net/everzet/bdd-in-symfony2

Mabey: The WHY behind TDD/BDD and the HOW with RSpec

JBehave website http://jbehave.org/