Introduction to Software Engineering: Tools and Environments

Session 11

Oded Lachish

Room: Mal 407
Visiting Hours: Wednesday 17:00 to 20:00
Email: oded@dcs.bbk.ac.uk
Last time

- Continuous integration - Jenkins
BDD – Behavioural Driven Development

• An agile software development technique (we’ll learn in OODP)

BDD goal – obtain a clear understanding of the behavioural requirements.

How?

Write tests by using the natural language as much as possible.

Why?

Shift the focus from the technical to the behavioural.
Integration Testing Cucumber
What is cucumber?

- An acceptance testing tool
- A BDD framework

Is it like Junit?

NO

Cucumber is used for converting plain text functional descriptions (Gherkin) into tests

How?

Cucumber parses a feature and generates a test template. The developer needs to insert the missing details.
Gherkin – feature (vending machine, feature file)

Feature: Dispensing
Check that dispenses the correct item

Scenario: buy item from vending machine
Given The vending machine is turned on
And the vending machine is fully loaded
When I have inserted <input_1> pence into vending machine
And I have pressed option <input_2>
And I press select
Then the result should be <output>

Examples:

<table>
<thead>
<tr>
<th>input_1</th>
<th>input_2</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>75</td>
<td>Chocolate Bar</td>
<td>Chocolate Bar</td>
</tr>
<tr>
<td>50</td>
<td>Chocolate Bar</td>
<td>insufficient money</td>
</tr>
<tr>
<td>50</td>
<td>Flapjack</td>
<td>Flapjack</td>
</tr>
</tbody>
</table>
Given, When & Then (And But)

- **Given** – is used in order to describe the state of the system
- **When** – describes the key action that the user performs
- **Then** – describes the expected outcomes
- **And, But** – in order to enable several Given, When, Then
Good features for feature files

• Should be from the user perspective and should not contain implementation issues

• Treat the feature files as part of the project documentation

• Feature files should be written as part of a collaboration

• Feature files should contain examples
In Java

```java
public class VendingSteps extends VMTestCase {

    @Given("The vending machine is turned on")
    public void turnOnVendingMachine() throws exception {
        this....
    }

    @When("I have inserted <input_1> pence into vending machine")
    public void insertMoneyIntoVM(int input_1) {
        this....
    }

    @Then("the result should be <output>")
    public void testVM(int output) {
        assertEquals(...
    }

```
Don’t forget

Installation and use is a bit more complicated than what we have seen
Deployment Tools
SmartFrog

• Java-based software framework for configuring, deploying and managing distributed software systems.

Why do we need something like this?

• Manage configuration

• Managing a large system with many components
SmartFrog (open source)

What does SmartFrog Do?

Why do we need something like this?

• Provide expressive notations for describing system configuration
• A secure Runtime system for deploying Software Components
• A library of dedicated components

SmartFrog has its own language.