BDD – Behaviour Driven Development

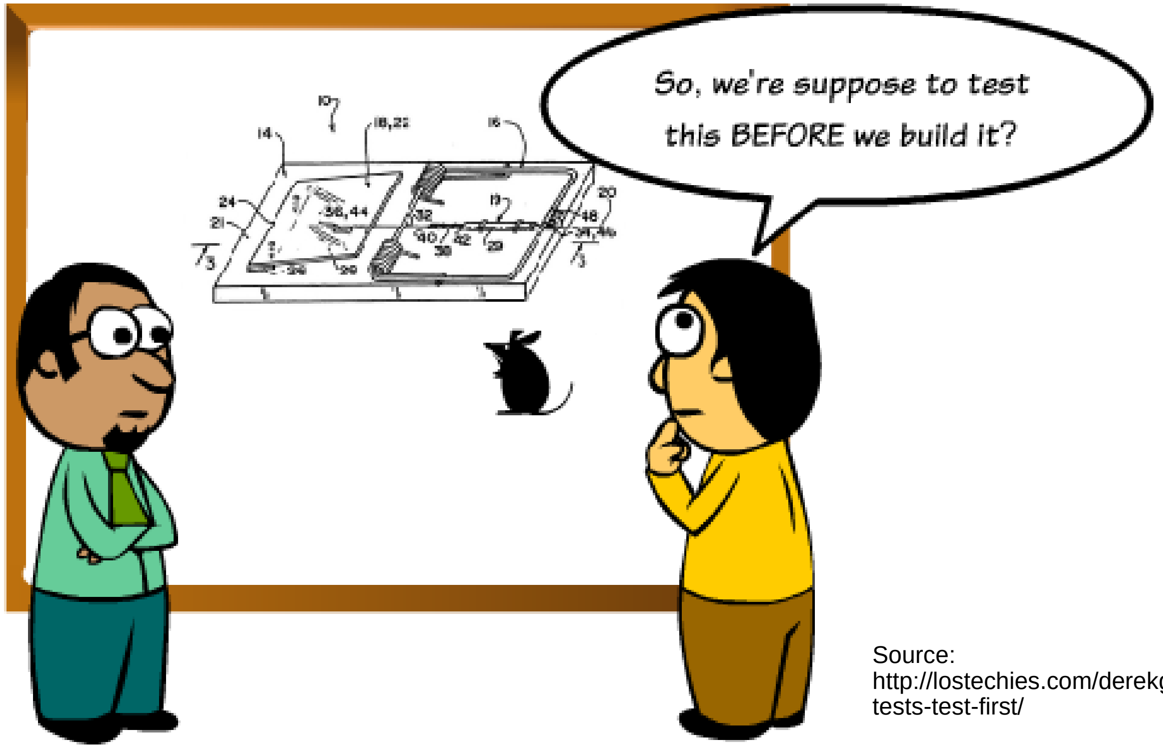
Marek Majchrzak, Andrzej Bednarz
Wrocław, 11.10.2011

BDD in a nutshell

It is an evolution in the thinking behind TDD (Test Driven Development) and Acceptance Tests Driven Planning

It brings together strands from TDD (Test Driven Development) and Domain Driven Design into an integrated whole

Source: http://behaviour-driven.org/



Source: http://lostechies.com/derekgreer/2011/03/21/effective-tests-test-first/

BDD origins – Dan North

I had a problem. While using and teaching agile practices like test-driven development (TDD) on projects in different environments, I kept coming across the same confusion and misunderstandings

Programmers wanted to know where to start, what to test and what not to test, how much to test in one go, what to call their tests, and how to understand why a test fails.

My response is behaviour-driven development (BDD).

Source: http://dannorth.net/introducing-bdd/

Template for user stories

As a *[X]*
I want *[Y]*
so that *[Z]*

in ubiquitous language!

As *a customer,*
I want *to withdraw cash from an ATM,*
so that *I don't have to wait in line at the bank.*

A story's behaviour is simply its acceptance criteria – if the system fulfills all the acceptance criteria, it's behaving correctly; if it doesn't, it isn't. So we created a template to capture a story's acceptance criteria.

Source: http://dannorth.net/introducing-bdd/

Template for scenarios

***Given** some initial context (the givens),*

***When** an event occurs,*

***Then** ensure some outcomes.*

Scenario 1:
Account is in credit

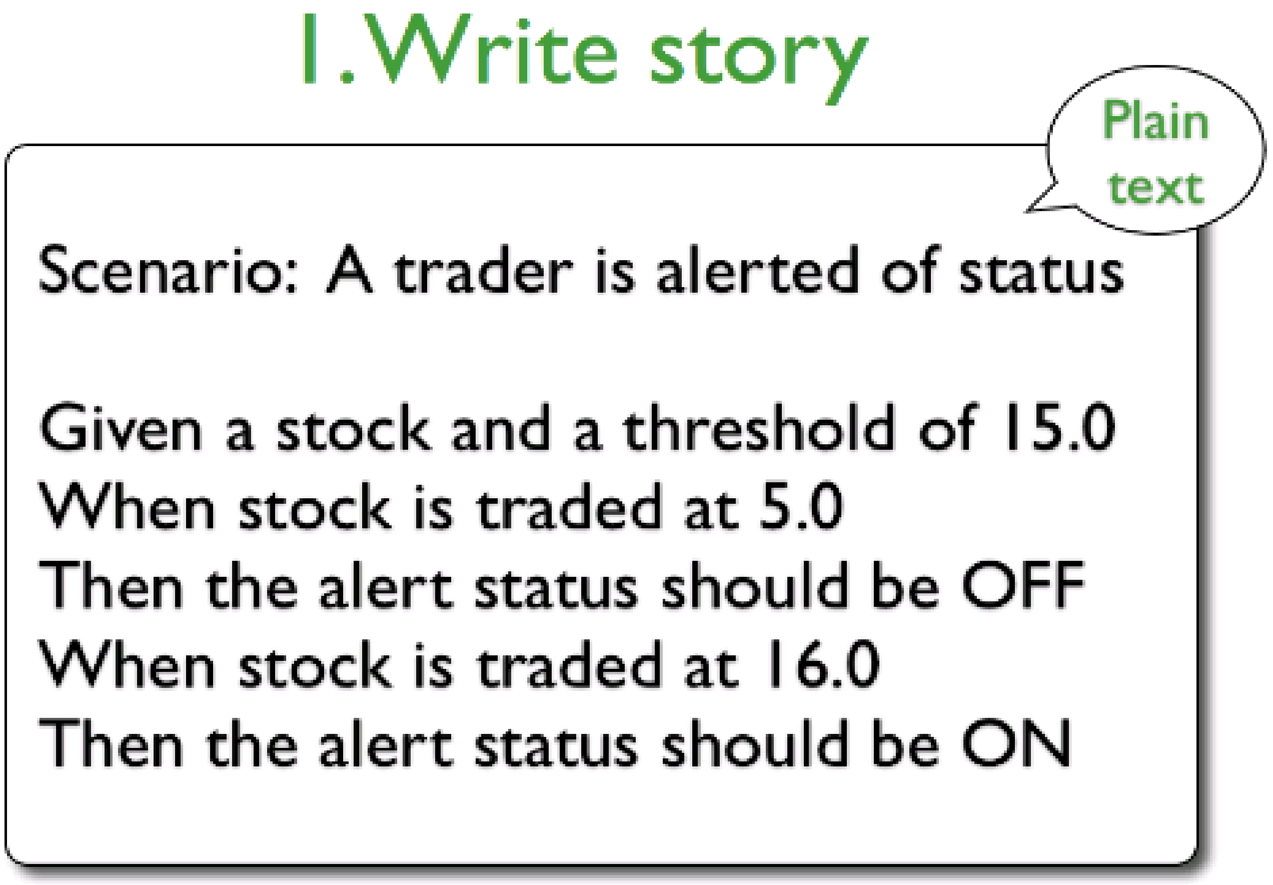
Given the account is in credit
And the card is valid
And the dispenser contains cash
When the customer requests cash
Then ensure the account is debited
And ensure cash is dispensed
And ensure the card is returned

Scenario 2:
Account is overdrawn past the overdraft limit

Given the account is overdrawn
And the card is valid
When the customer requests cash
Then ensure a rejection message is displayed
And ensure cash is not dispensed
And ensure the card is returned

Source: http://dannorth.net/introducing-bdd/

Example

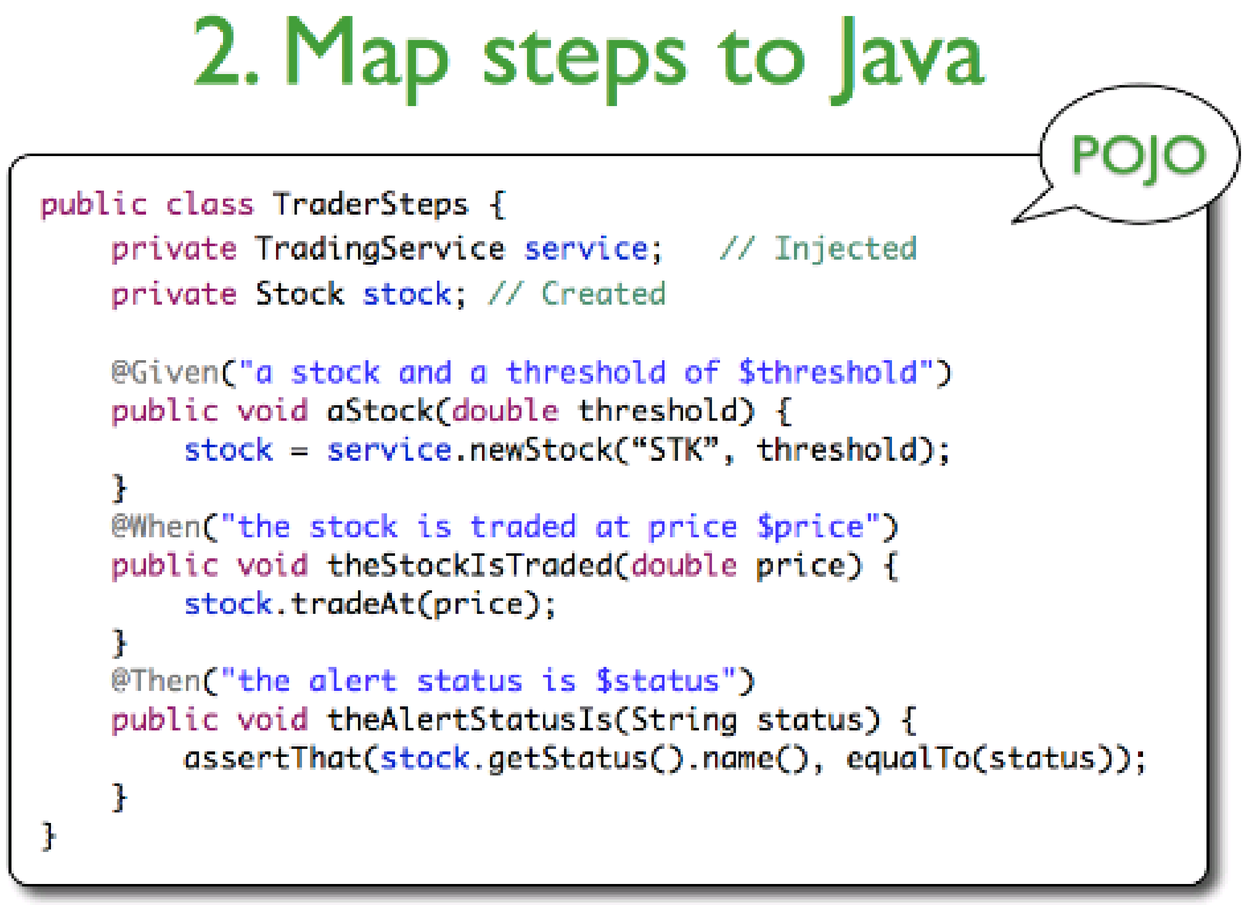


1. Write story

Plain text

```
Scenario: A trader is alerted of status

Given a stock and a threshold of 15.0
When stock is traded at 5.0
Then the alert status should be OFF
When stock is traded at 16.0
Then the alert status should be ON
```

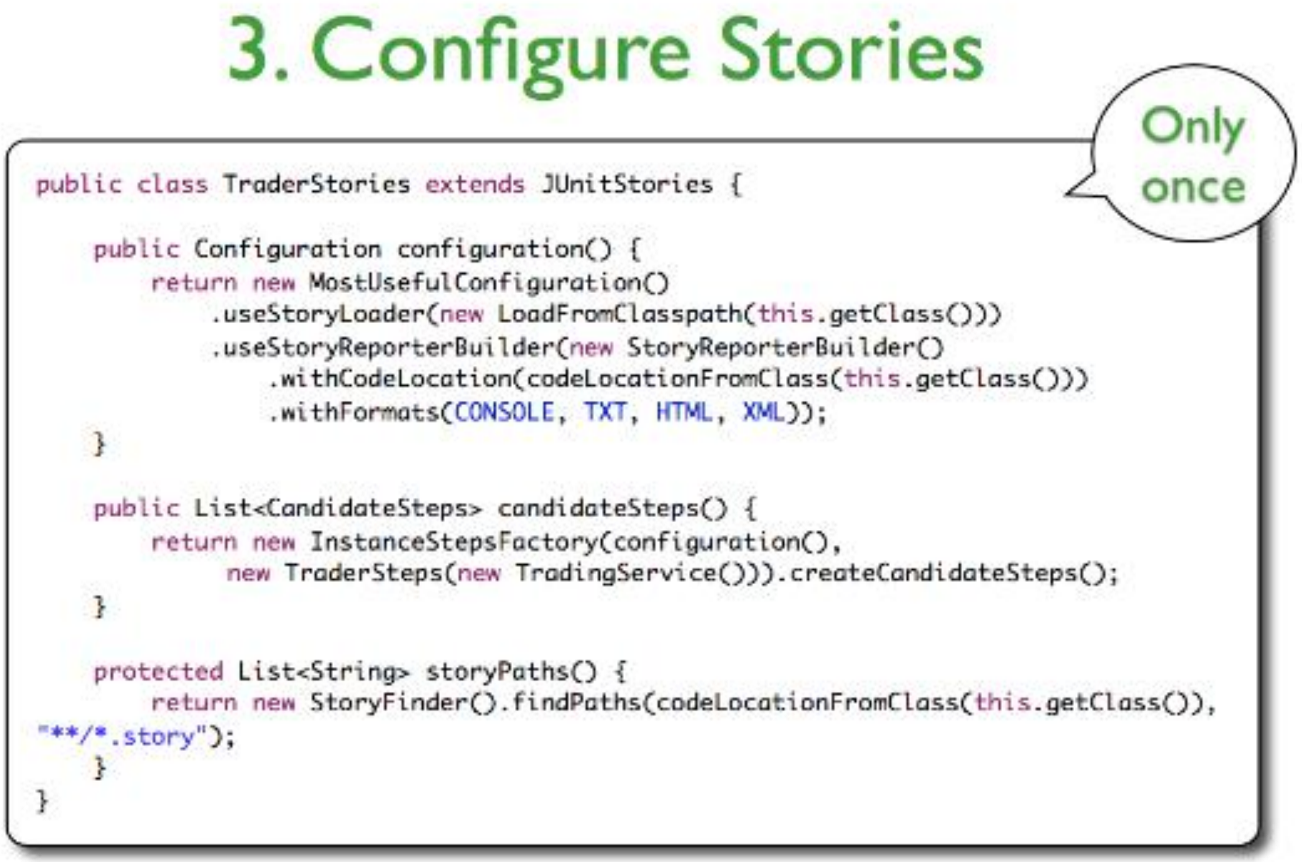
Example



2. Map steps to Java

```java
public class TraderSteps {
    private TradingService service;   // Injected
    private Stock stock; // Created

    @Given("a stock and a threshold of $threshold")
    public void aStock(double threshold) {
        stock = service.newStock("STK", threshold);
    }
    @When("the stock is traded at price $price")
    public void theStockIsTraded(double price) {
        stock.tradeAt(price);
    }
    @Then("the alert status is $status")
    public void theAlertStatusIs(String status) {
        assertThat(stock.getStatus().name(), equalTo(status));
    }
}
```

Example



3. Configure Stories

Only once

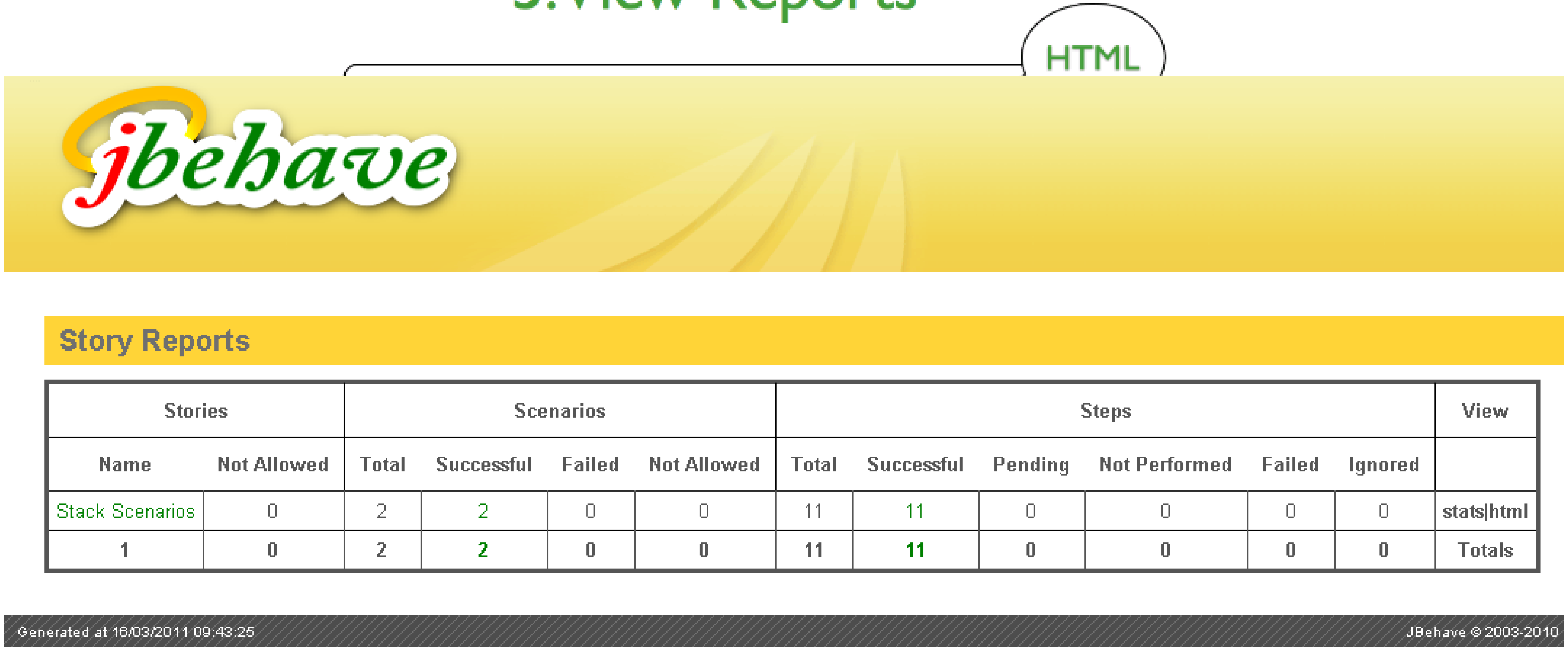
```
public class TraderStories extends JUnitStories {

    public Configuration configuration() {
        return new MostUsefulConfiguration()
            .useStoryLoader(new LoadFromClasspath(this.getClass()))
            .useStoryReporterBuilder(new StoryReporterBuilder()
                .withCodeLocation(codeLocationFromClass(this.getClass()))
                .withFormats(CONSOLE, TXT, HTML, XML));
    }

    public List<CandidateSteps> candidateSteps() {
        return new InstanceStepsFactory(configuration(),
             new TraderSteps(new TradingService())).createCandidateSteps();
    }

    protected List<String> storyPaths() {
        return new StoryFinder().findPaths(codeLocationFromClass(this.getClass()),
"**/*.story");
    }
}
```

Example

4. Run Stories

Example

5.View Reports



Story Reports

Stories		Scenarios				Steps						View
Name	Not Allowed	Total	Successful	Failed	Not Allowed	Total	Successful	Pending	Not Performed	Failed	Ignored	
Stack Scenarios	0	2	2	0	0	11	11	0	0	0	0	stats|html
1	**0**	**2**	**2**	**0**	**0**	**11**	**11**	**0**	**0**	**0**	**0**	Totals

Generated at 16/03/2011 09:43:25 JBehave © 2003-2010

Another example

```
Given a stock of <symbol> and a <threshold>
When the stock is traded at <price>
Then the alert status should be <status>

Examples:
|symbol|threshold|price|status|
|STK1|10.0|5.0|OFF|
|STK1|10.0|11.0|ON|
```

The Core Principles

Business and Technology should refer to the same system in the same way – *Its all behaviour*

Any system should have an identified, verifiable value to the business – *Wheres the business value*

Up-front analysis, design and planning all have a diminishing return – *Enough is enough*

BDD vs TDD

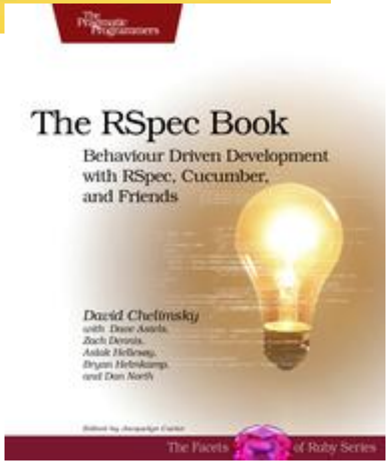
- ***It is important to keep BDD distinct from TDD. These two practices are equally important but address different concerns and should be complementary in best development practices.***

- ***BDD is concerned primarily with the specification of the behavior of the system under test as a whole, thus is particularly suited for acceptance and regression testing. TDD is concerned primarily with the testing of a component as a unit, in isolation from other dependencies, which are typically mocked or stubbed.***

- ***BDD should talk the language of the business domain and not the language of the development technology, which on the other hand is "spoken" by TDD.***

Sources, References and more…

1. *Dan North, Introducing BDD, http://dannorth.net/introducing-bdd/*
2. *BDD Wiki, http://behaviour-driven.org/*
3. *JBehave, http://jbehave.org/*
4. *RSpec, http://rspec.info/*
5. *Cucumber, http://cukes.info/*
6. *Dan North, Deliberate Discovery, http://dannorth.net/2010/08/30/introducing-deliberate-discovery/*
7. *Dan North, Programming is not a craft, http://dannorth.net/2011/01/11/programming-is-not-a-craft/ & Martin Fowler http://martinfowler.com/bliki/CraftmanshipAndTheCrevasse.html*
8. *Liz Keogh's blog, http://lizkeogh.com/*


The RSpec Book
Behaviour Driven Development with RSpec, Cucumber, and Friends
David Chelimsky
The Facets of Ruby Series

behaviour driven development with elegance and joy

Vielen Dank für Ihre Aufmerksamkeit!

www.de.capgemini.com

Programming is not a craft - Opposition to Software Craftmanship

The software shouldn't be at the center of a programmer's world, instead a programmer should focus on the benefit that the software is supposed to deliver.

Martin Fowler

I would love to see someone rewrite the Software Craftsmanship Manifesto in terms of getting results and delighting customers.

Dan North

I don't want "steadily adding value," I want "amazing their customers every day!" Software craftsmen should be egoless, humble, with a focus on the outcome rather than the code or the process.

Dan North

… refactoring to oblivion while failing to deliver, focusing on test coverage while the company continues to leak money, worrying about the latest testing tool while failing to talk to stakeholders, or insisting that no one can interrupt their two-week sprint once they've made a commitment …

Liz Keogh