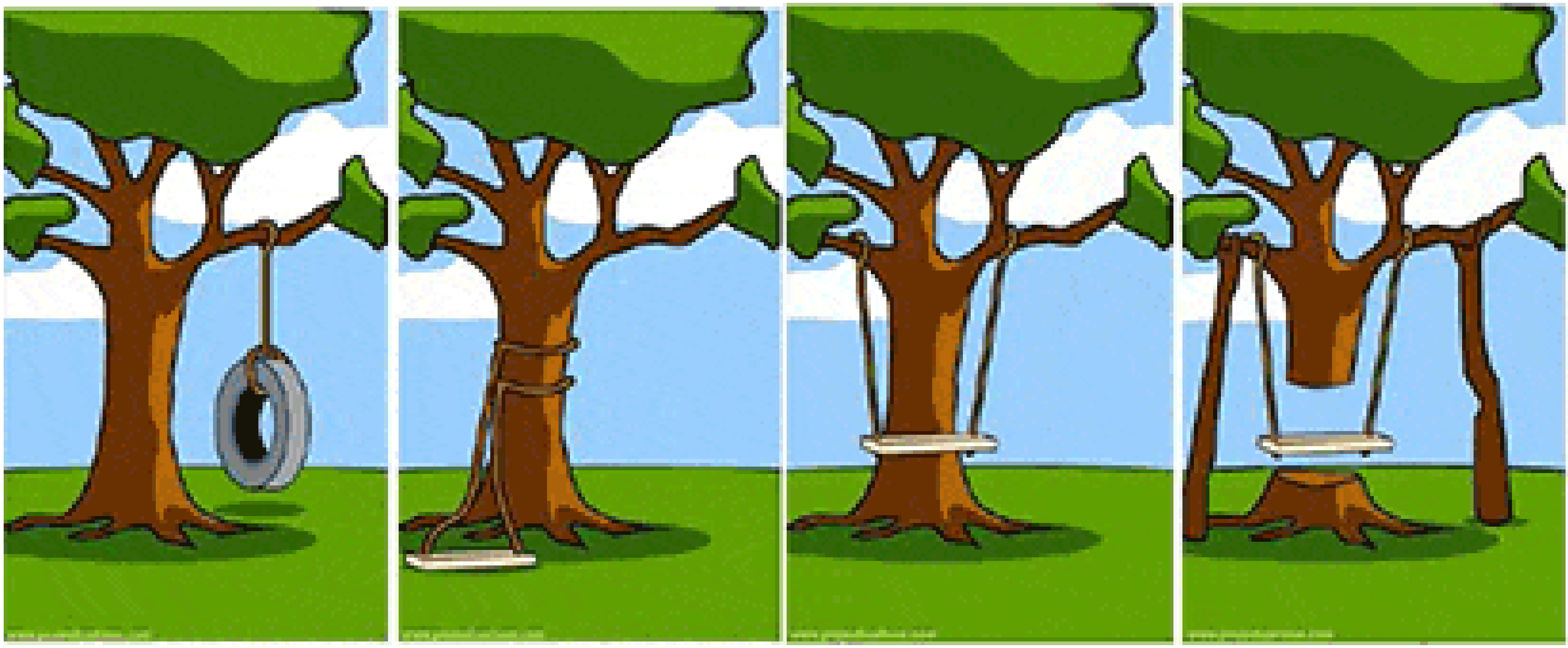
Code that matters

An introduction to Behaviour-Driven Development

Agenda

- A little bit of background
- The challenge of doing the right thing
- Why TDD alone won't help
- Alternatives
- EasyB

Bridging the gap ...



CALLISTA
Enterprise

©J.King ModernAnalyst.com
?

"The Line of Business wants a new system ASAP. They are very busy dealing with issues so they can't assign any stakeholders to the project. When can you have the requirements document ready?"

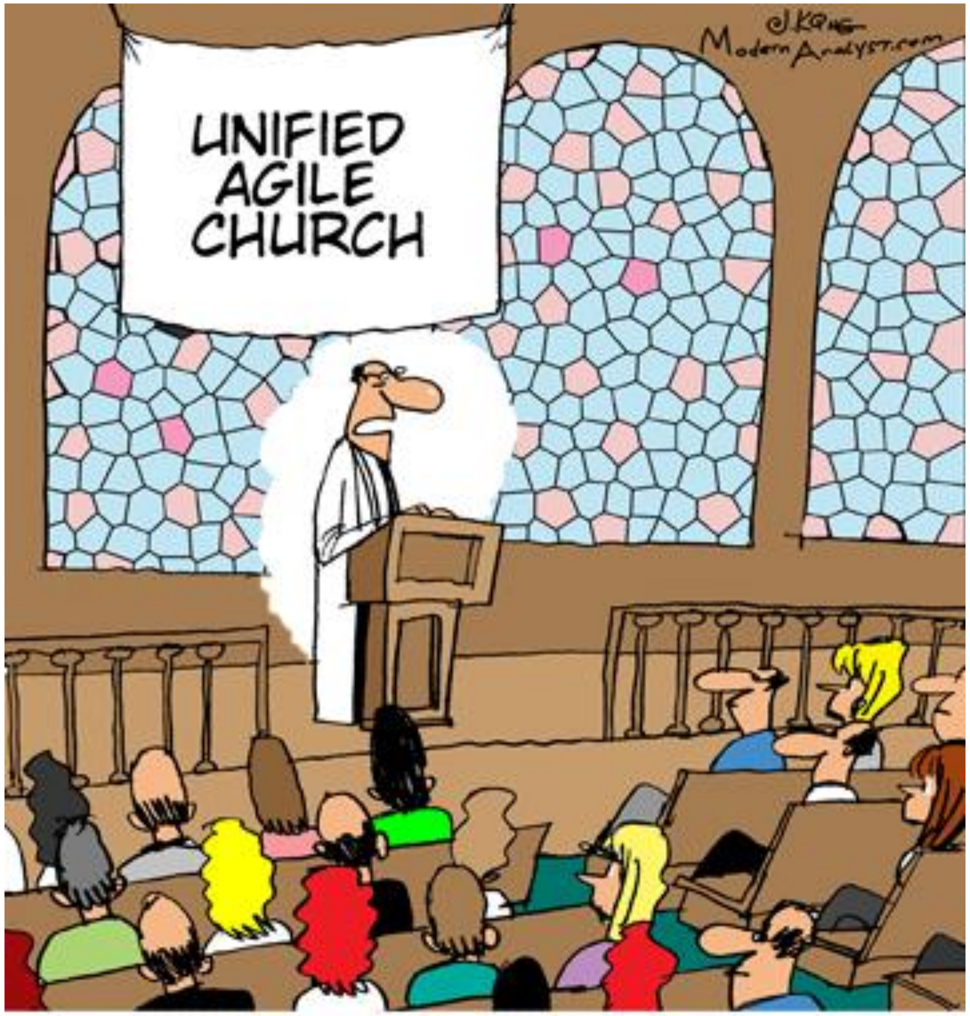
©KRG
ModernAnalyst
Business Field
Requirements Repository

TRADITIONAL REQUIREMENTS

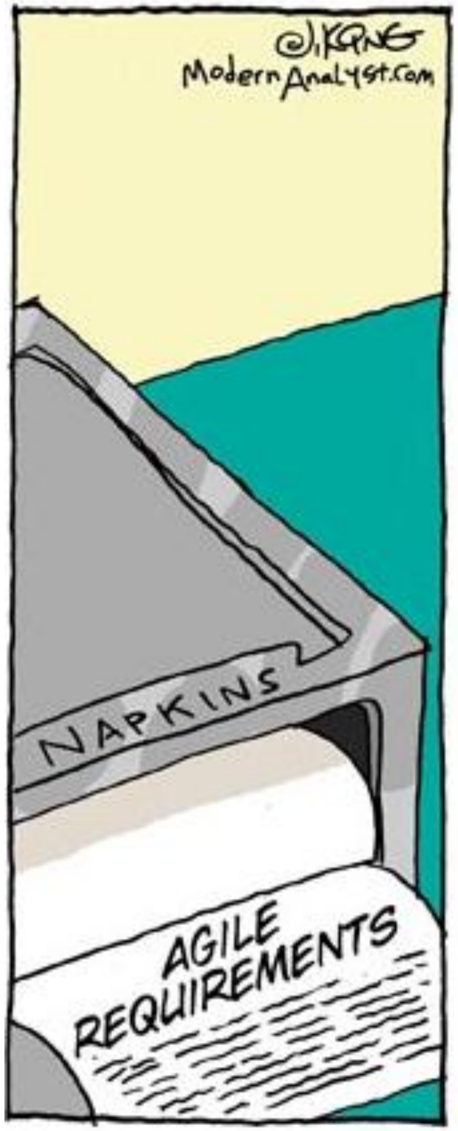
CALLISTA
Enterprise

©JiKING
ModernAnalyst.com

“These are not the requirements the business communicated to us!”



"Repent of your Traditional sins! Only through Agile you can save your projects!"


©JKRNG
ModernAnalyst.com
NAPKINS
AGILE REQUIREMENTS

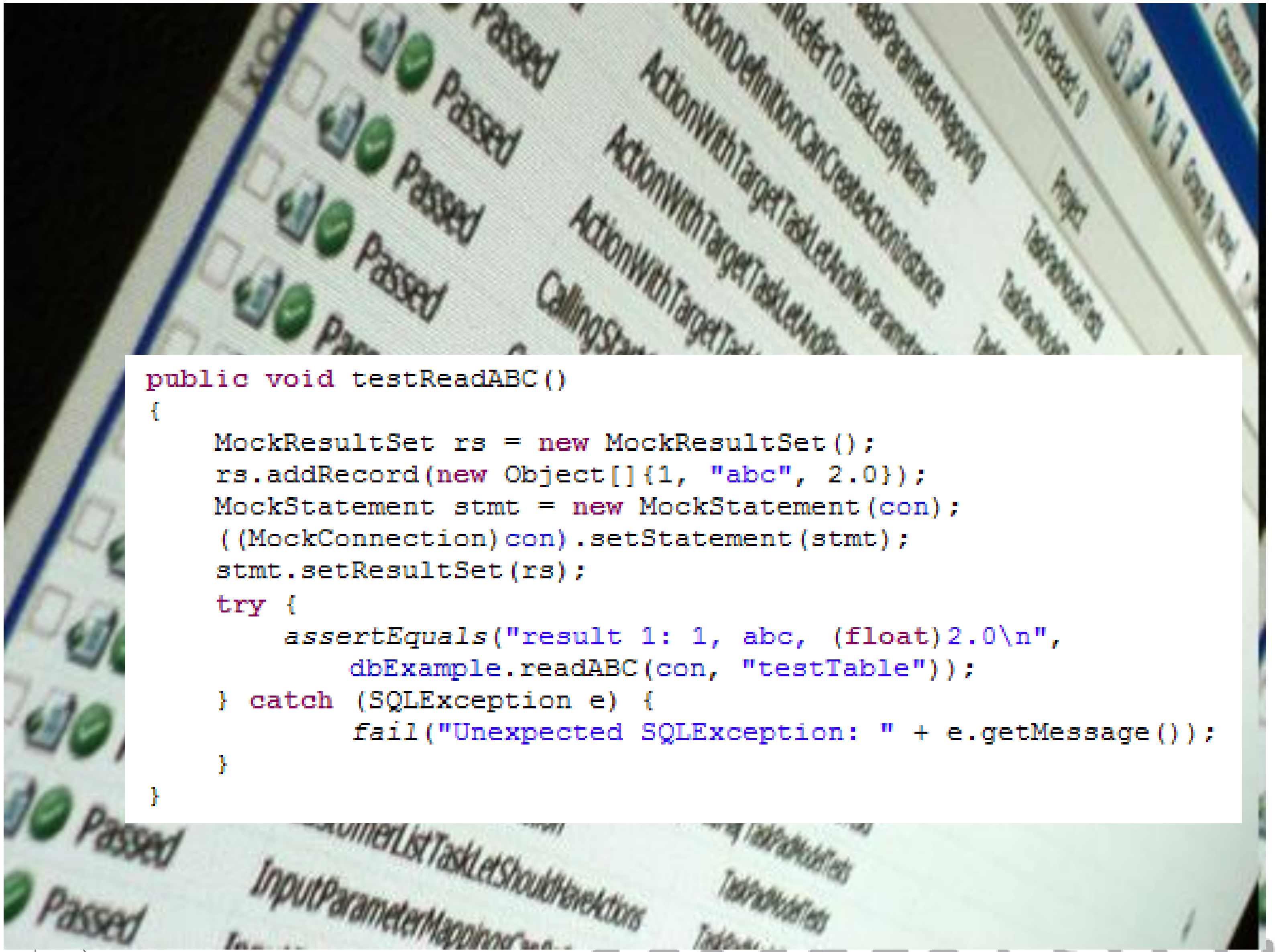
Card, Conversation, and Confirmation

"... a user story is a reminder to have a conversation with your customer ..."

Scott Ambler, Agile Modeling

corbis

TDD to the rescue?



```
public void testReadABC()
{
    MockResultSet rs = new MockResultSet();
    rs.addRecord(new Object[]{1, "abc", 2.0});
    MockStatement stmt = new MockStatement(con);
    ((MockConnection)con).setStatement(stmt);
    stmt.setResultSet(rs);
    try {
        assertEquals("result 1: 1, abc, (float)2.0\n",
            dbExample.readABC(con, "testTable"));
    } catch (SQLException e) {
            fail("Unexpected SQLException: " + e.getMessage());
    }
}
```

"These are not the requirements the business communicated to us!"

CALLISTA
Enterprise

TDD can help us build the system right, but it doesn't really help us **build the right system**.

TDD means ...

Low level, geeky format

TDD means ...

All or nothing, red or green, ...

TDD means ...

Working from the inside out

starting with the tiny details,
instead of the big picture

Business people wants to express the required **behaviour**, not necessarily write tests

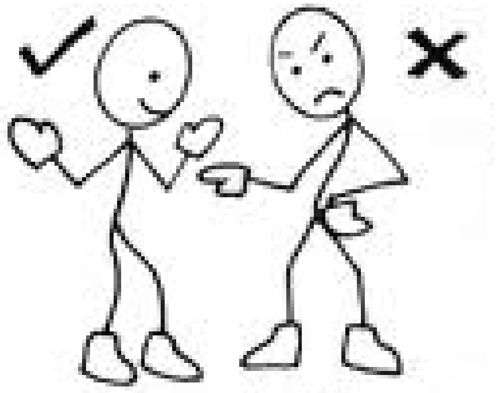
We want to ...

State required behaviour in a precise enough way, using the vocabulary of the business

Yes,
I want to STOP my dog from,
being aggressive, chewing,
digging, barking, jumping,
biting, whining
And All other behavior problems!

We want to …

State required behaviour using **examples**



We want to ...

Gradually **refine** and **evolve** the examples, as the shared understanding grows

We want to ...

Work **outside-in**, starting with the overall scenarios that gives business value

CALLISTA Enterprise

Same, same, but different

- Close to the original notion of "Customer Tests"
- Known under different names:
  - Test-driven Requirements
  - Agile Acceptance Tests
  - Behaviour Driven Development

CALLISTA
Enterprise

Lots of available tools and frameworks:

- FitNesse
- Text tool
- RSpec
- Cucumber
- Cuke4Duke
- EasyB
- …

easyb

- A BDD framework for the Java platform
- Based on Groovy

easyb in a nutshell: *Story*

- Use a natural language, narrative approach
- Describe precise requirements
- Usually made up of a set of scenarios
- Use an easy-to-understand structure
  given ”some context”
  when ”something happens”
  then ”something else should happen”

Example User Story

White-collar time reporting **5**

As a White-collar Employee

I want to record my working time

So that I get correctly calculated salary

Example Task

Task: Regular days

Given a white-collar employee

When he arrives at 8:00 and leaves at 17:00

Then his result for that day is 0.

CALLISTA
Enterprise

Example Task

Task: Flex

Given a white-collar employee with flex
When he arrives at 8:00 and leaves at 18:00
Then his result for that day is +60.

Given a white-collar employee with flex
When he arrives at 8:00 and leaves at 16:00
Then his result for that day is -60.

Example Task

Task: Overtime

Given a white-collar employee with flex and overtime
When he arrives at 8:00 and leaves at 19:00
Then his flex result for that day is +120
and his overtime is 0.

Given a white-collar employee with flex and overtime
When he arrives at 8:00 and leaves at 20:10
Then his flex result for that day is +120
and this overtime is +1,5.

Demo

To summarize:

- TDD can help us build the system right, but that is not enough
- TDR/ATDD/BDD can help us **build the right system**, by providing a slightly different perspective:
  - State required **behaviour** using **examples**
  - Gradually **refine** and evolve the examples
  - Work from the **outside-in**
- easyb is a bdd framework for the Java platform, that leverages the power and beauty of Groovy

Time for Questions!