



IT WORKS – WE CARE

TECHTALK



Testing im Wandel

Behavior Driven Development & SpecFlow

**TechTalk QM Frühstück
18. Februar 2010, Wien**



IT WORKS - WE CARE

TECHTALK

Herzlich Willkommen!

Christoph Seufert

Software Architect

Competence Lead Architektur/Entwicklung

Christian Hassa

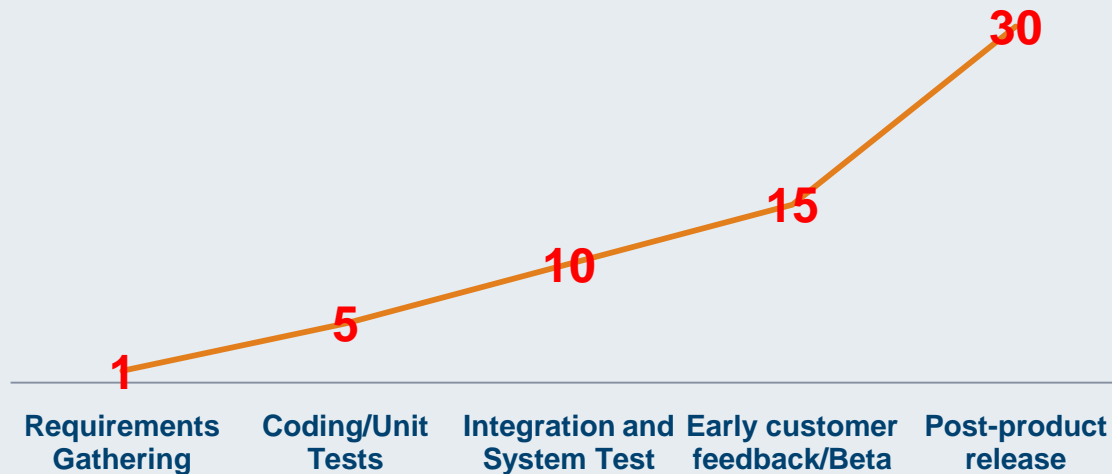
Geschäftsführender Gesellschafter

Methodik/Anforderungsanalyse



How much does a sw-defect cost?

Relative cost factor when found at different stages of software development



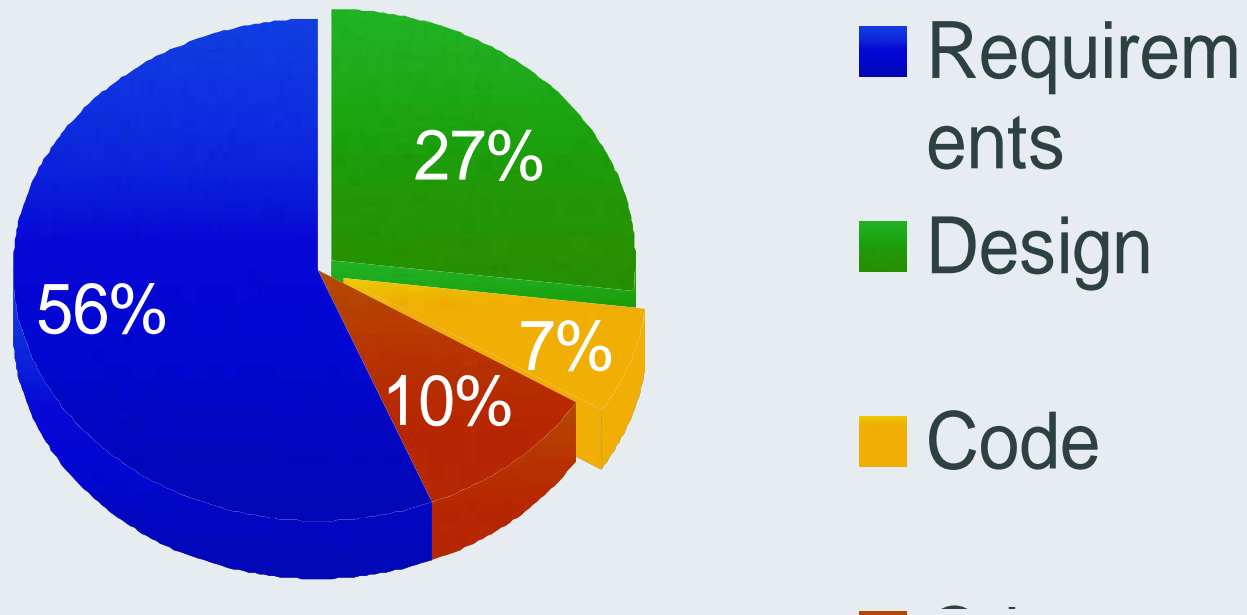
The Economic Impacts of Inadequate Infrastructure for Software Testing
National Institute of Standards and Technology, USA (2002)



IT WORKS - WE CARE

TECHTALK

Where are errors introduced?

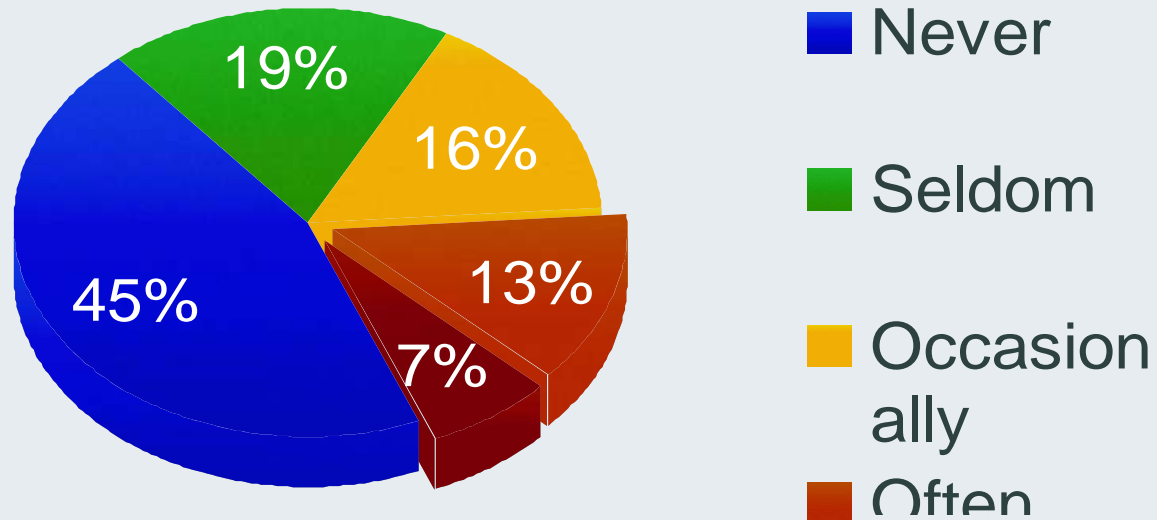


Source: James Martin, An Information Systems Manifesto



How to measure errors in Reqmts'?

How often are delivered features actually used?



CHAOS Report 2000, Standish Group



IT WORKS - WE CARE

TECHTALK

Stop Bugs early!



- Traditional developer testing focuses on code and maybe design
- Testing requirements is hardly ever a topic
- We can profit tremendously when we improve in requirements gathering.



IT WORKS - WE CARE

TECHTALK

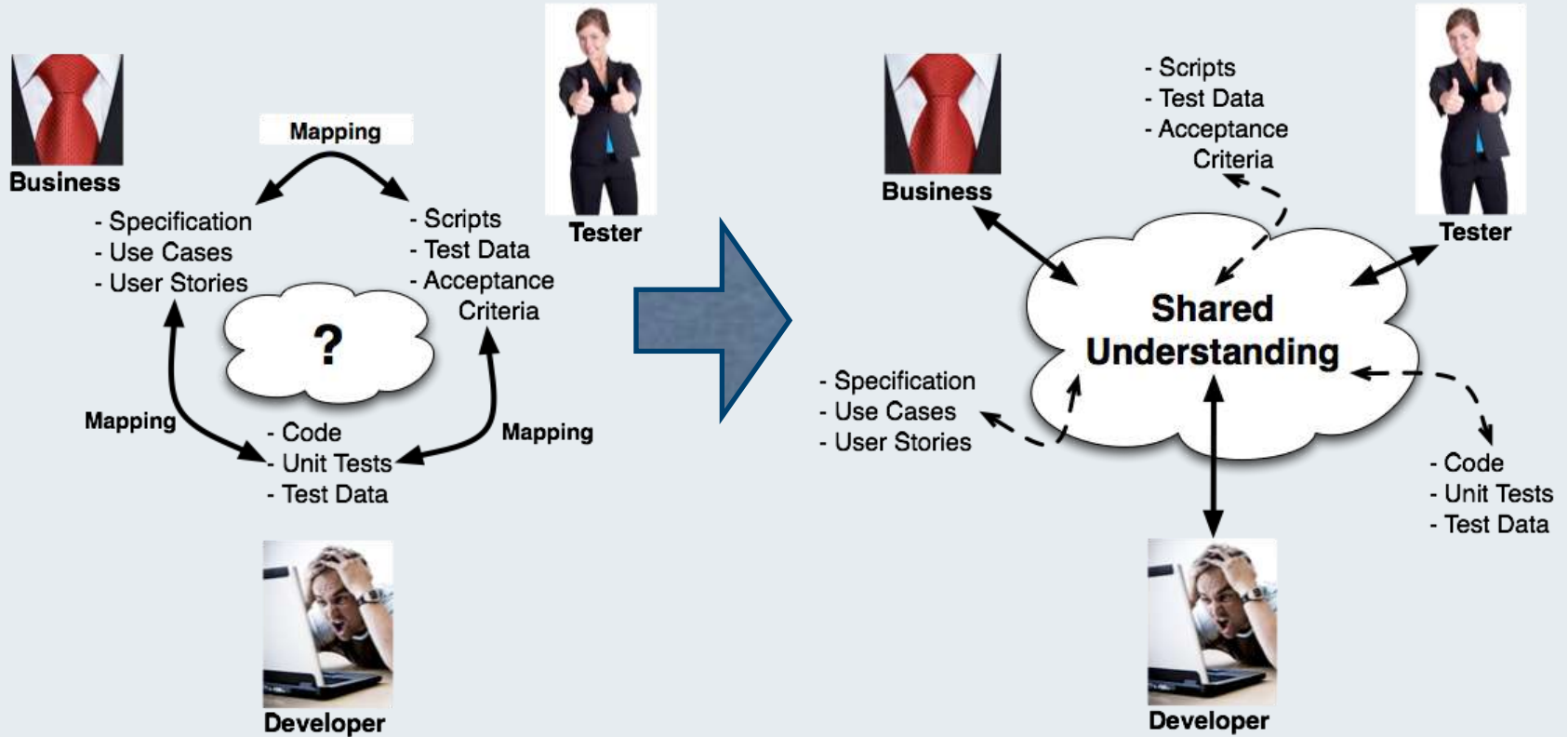
The Communication Gap

"I believe that the hardest part of software projects, the most common source of project failure, is communication with the customers and users of that software."

- Martin Fowler, DSL Book



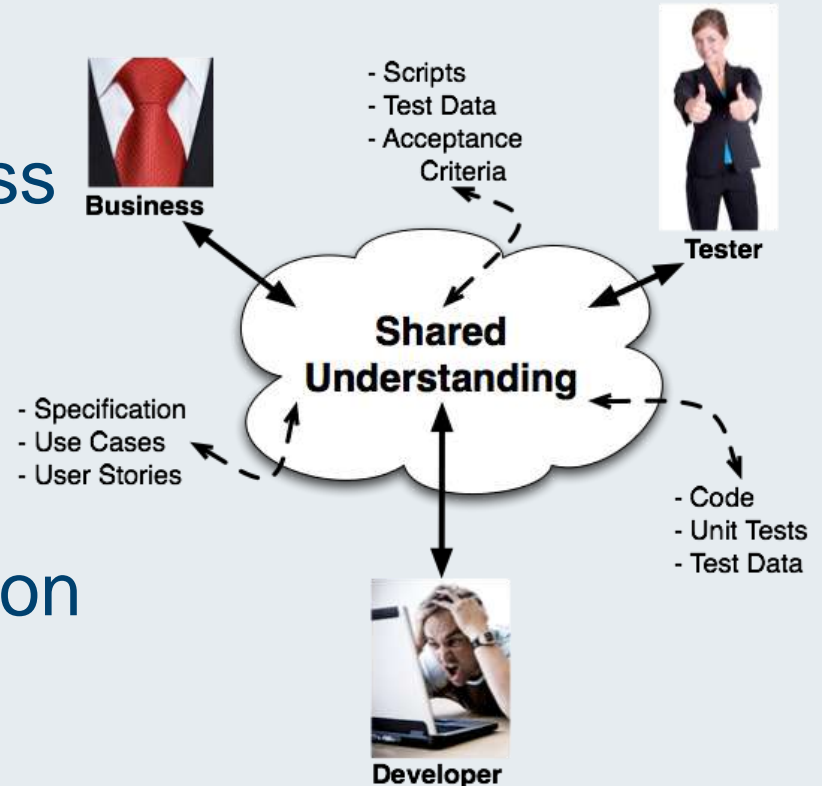
Building a shared understanding





Shared understanding

- Terminology and abstractions of business
- Business readable
- Concrete enough for developers
- Bound to implementation
- Consistent





IT WORKS - WE CARE

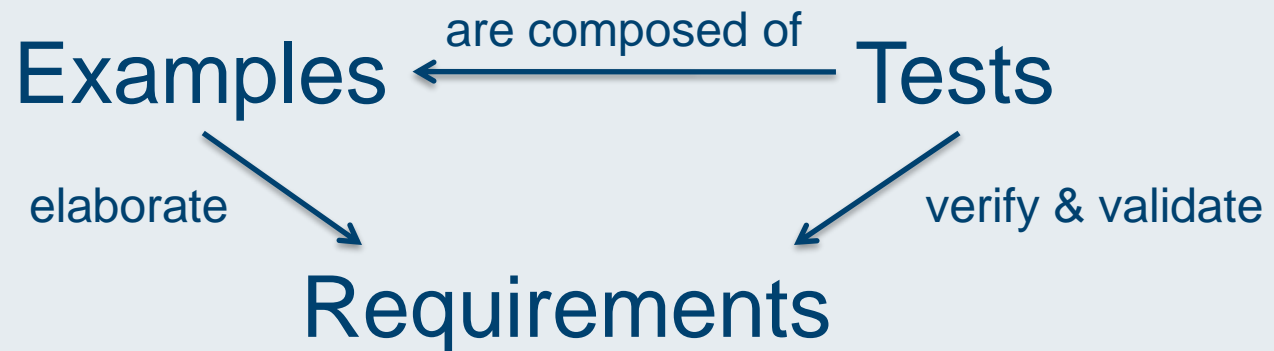
TECHTALK

How?



Specification by Example

- Abstract requirements and specifications are not a good tool for communication. Real-life examples are much better.
- We mostly communicate by examples
- Usually examples are not formalized and not shared





Examples

- Enable developers and testers to **communicate efficiently** with business people
- Focus on the **actual business value**
- Provide **enough information** for developers to implement and for testers to verify
- Realistic examples contain **precise information** and ask for precise answers
- Are a way to **test specifications!**



Requirements vs. Tests

As formality increases, tests and requirements become indistinguishable. At the limit, tests and requirements are equivalent.

Equivalence Hypothesis (Martin, Melnik)

→ Executable test-based specifications



IT WORKS - WE CARE

TECHTALK

Test based Development Methodology

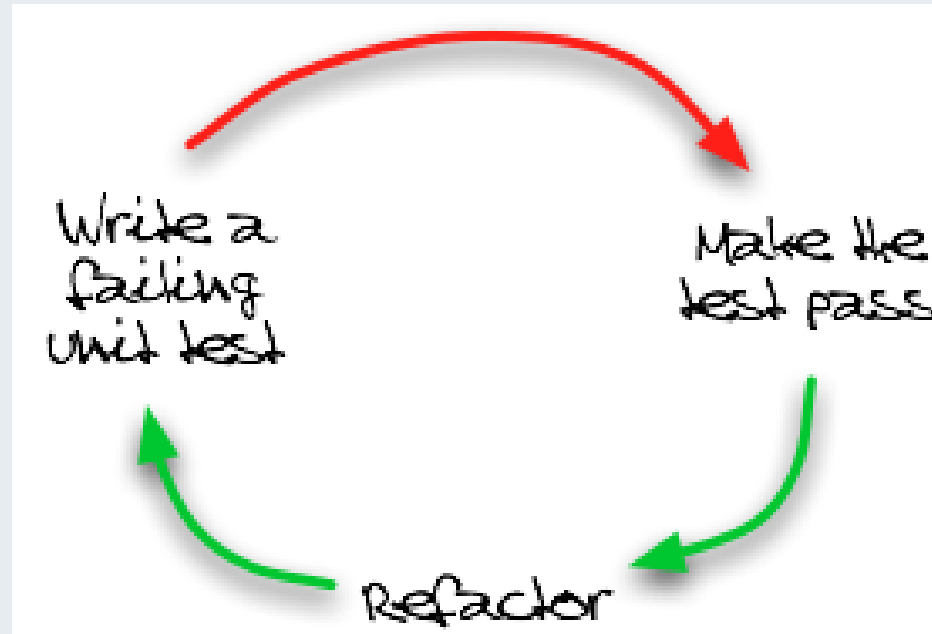
- ATDD: Acceptance Test Driven Development
 - User Tests: Part of XP
- BDD: Behavior Driven Development
 - 2001 by Dan North



IT WORKS - WE CARE

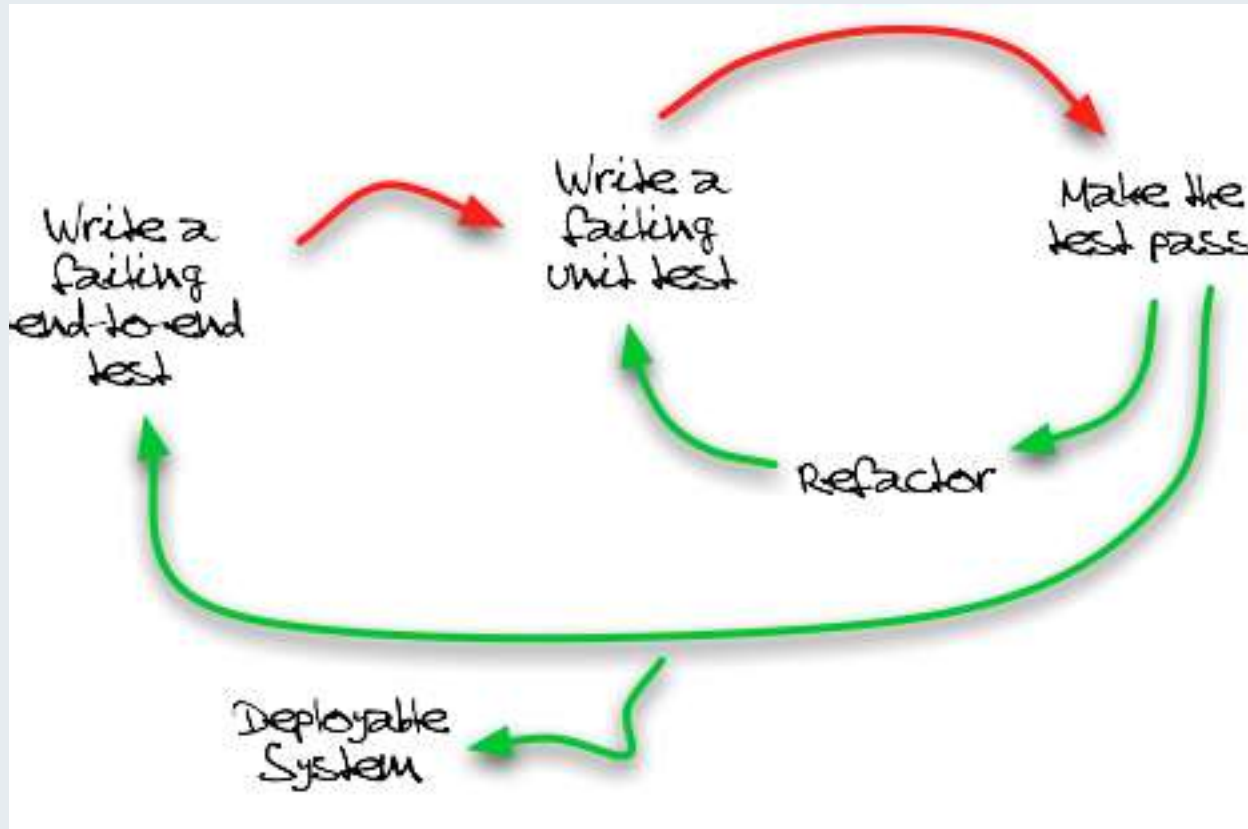
TECHTALK

TDD





Rising the Level of Abstraction





The real Goal

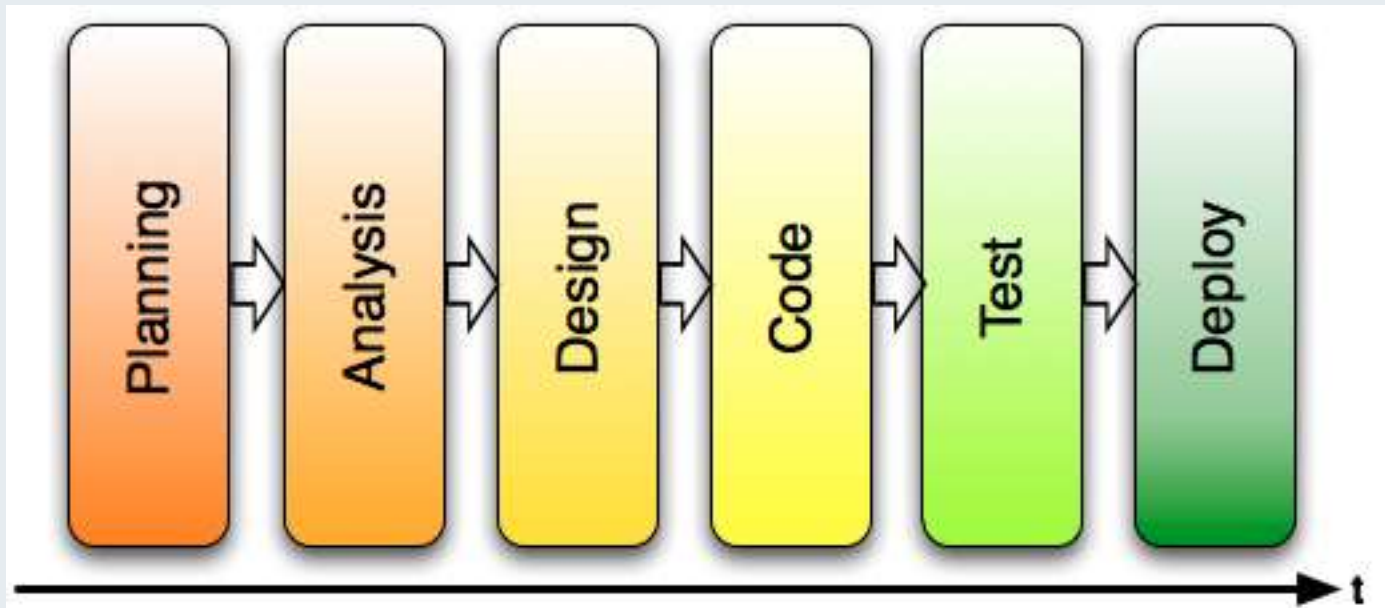
- Test Driven Development is about **Design**
- Acceptance Test Driven Development is about **Specification and Communication**
- Writing acceptance tests early
 - is a **requirements engineering technique**
 - can save time and money
 - helps businesses better respond to change.



IT WORKS - WE CARE

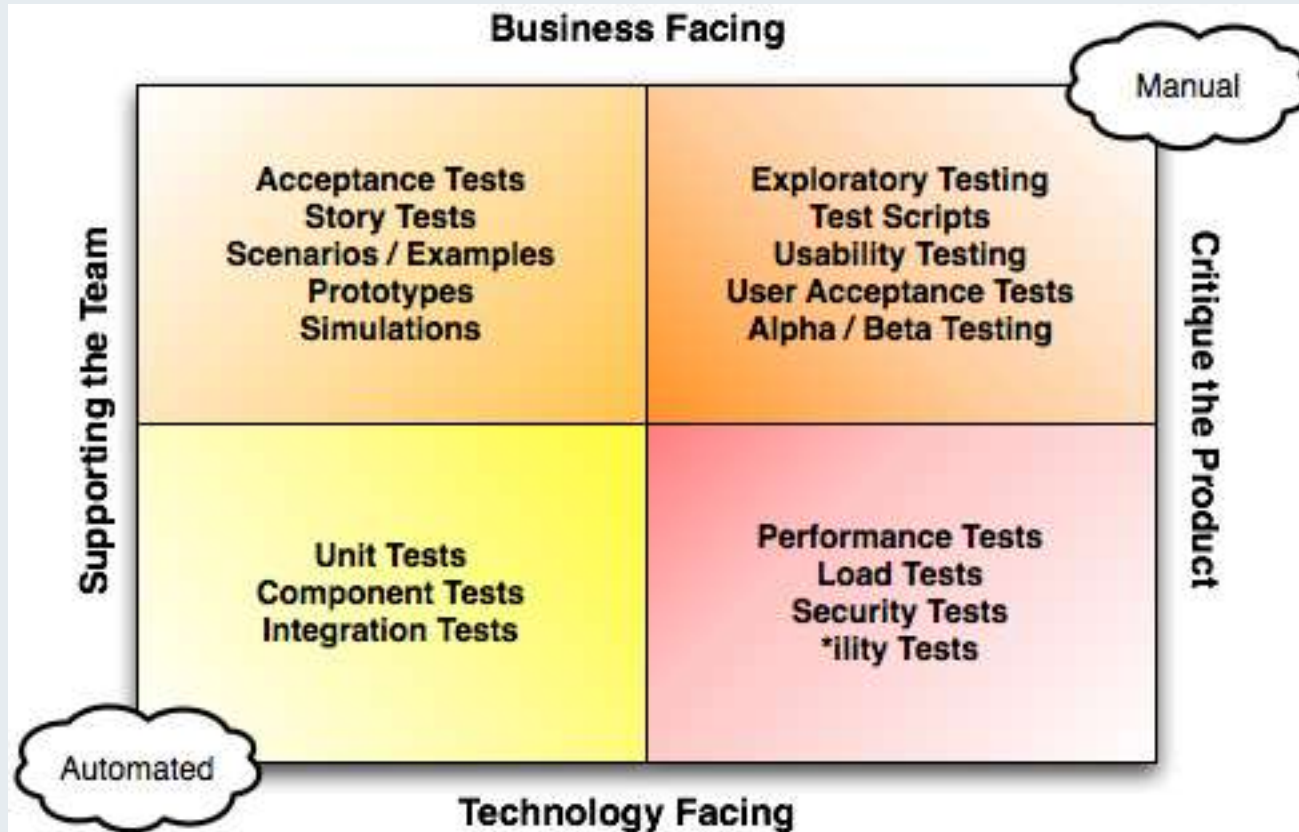
TECHTALK

What about testing?





Testing can be more ...

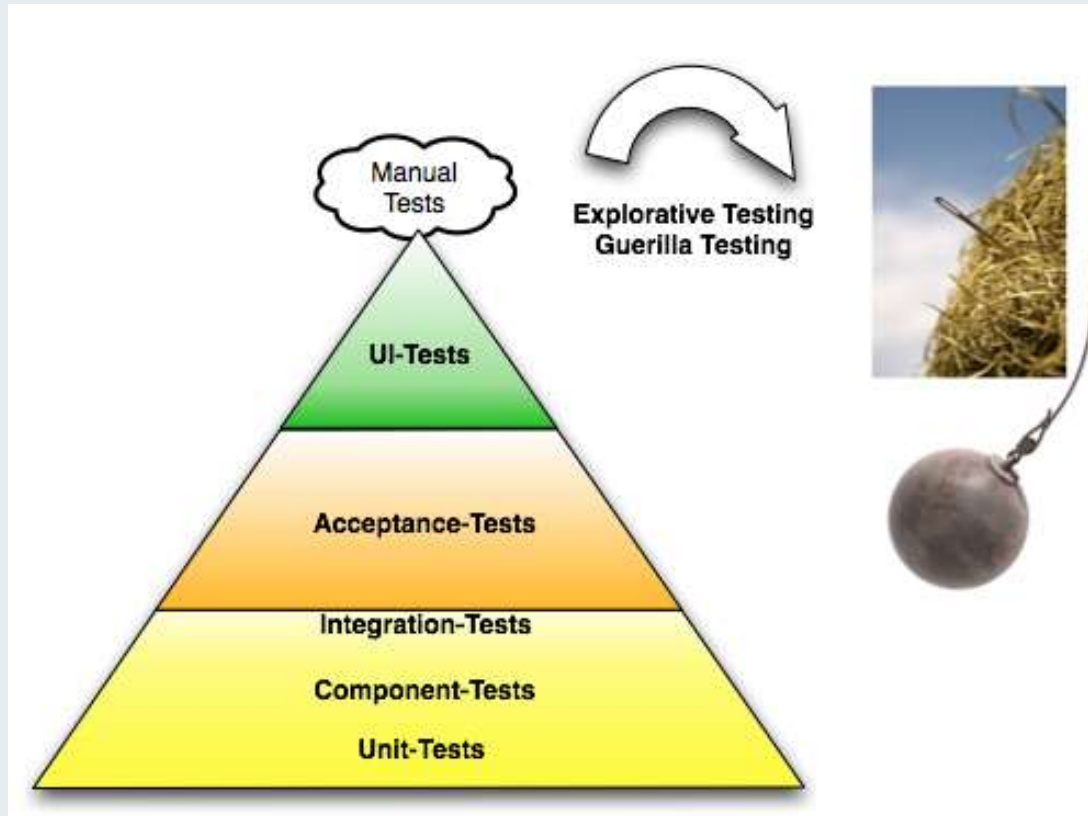


Agile Testing Quadrants from "Agile Testing" by Crispin/Gregory

- Not only detect bugs
- Improve quality by preventing bugs ...



Testing Pyramid



Test Automation Pyramid by Mike Cohn

- Automation allows more resources for „valuable“ testing activities



IT WORKS - WE CARE

TECHTALK

DEMO



Bowling

1	4	4	5	6	▲	5	▲	■	0	1	7	▲	6	▲	■	2	▲	6
5	14	29	49	60	61	77	97	117	133									

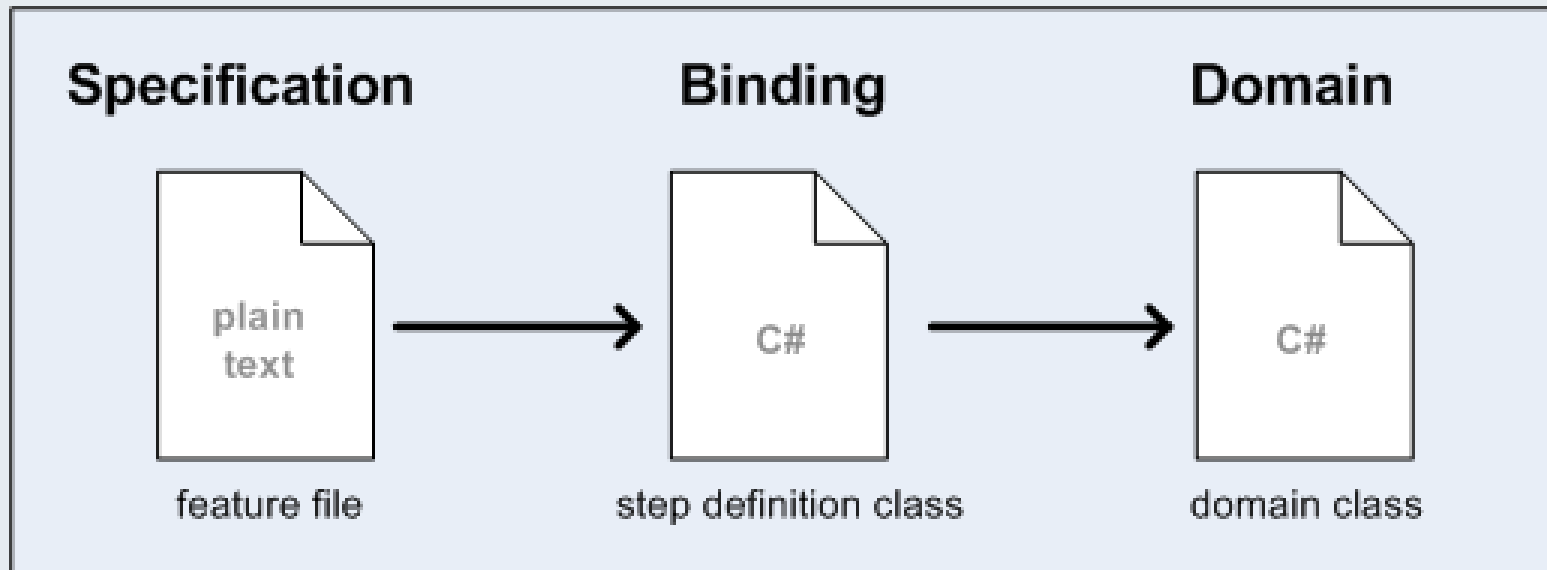
- 10 Durchgänge
- 10 Kegel pro Durchgang
- 2 Kugeln pro Durchgang
- Strike: Alle 10 Kegel mit der ersten Kugel
-> Die nächsten zwei Würfe werden zum aktuellen Durchgang addiert
- Spare: Alle 10 Kegel mit zwei Kugeln
-> Der nächste Wurf wird zum aktuellen Durchgang addiert
- Der letzte Durchgang kann 3 Kugeln umfassen



IT WORKS - WE CARE

TECHTALK

Das Konzept





Step 1: Write a feature

Bowling.feature

Feature: Bowling

In order to have fun

As a bowling enthusiast

I want to play a bowling game

Scenario: Gutter Game

Given a new bowling game

When all of my balls are landing in the gutter

Then my total score should be 0

Scenario: All Strikes

Given a new bowling game

When all of my rolls are strikes

Then my total score should be 300



Step 2: Watch it fail

Unit Test Sessions - Session #1

Session #1

Tests failed: 2, passed: 0, ignored: 0

- <Bowling.SpecFlow> (2 tests) 2 tests failed
 - Bowling.SpecFlow (2 tests) 2 tests failed
 - BowlingFixture (2 tests) 2 tests failed
 - GutterGame Failed: MissingMethodException: Method
 - AllStrikes Failed: MissingMethodException: Method

BowlingFixture.GutterGame : Failed

```
Given a new bowling game
-> No matching step definition found for the step. Use the following code to create one:
[Binding]
public static class StepDefinitons
{
    [Given(@"a new bowling game")]
    public static void GivenANewBowlingGame ()
    {
        ScenarioContext.Current.Pending ();
    }
}
```



Step 3: Implement step definitions

```
BowlingSteps.cs | Bowling.feature
Bowling.SpecFlow.BowlingSteps
namespace Bowling.SpecFlow
{
    [Binding]
    public static class BowlingSteps
    {
        [Given(@"a new bowling game")]
        public static void GivenANewBowlingGame ()
        {
            Game = new Game ();
        }

        [When(@"all of my balls are landing in the gutter")]
        public static void WhenAllOfMyBallsAreLandingInTheGutter ()
        {
            for (int i = 0; i < 20; i++)
            {
                Game.Roll (0);
            }
        }

        [When(@"all of my rolls are strikes")]
        public static void WhenAllOfMyRollsAreStrikes ()
        {
            for (int i = 0; i < 20; i++)
```



Step 4: Create a skeleton

The screenshot shows an IDE window with three tabs: Game.cs, BowlingSteps.cs, and Bowling.feature. The active tab is Game.cs, which displays the following C# code skeleton:

```
namespace Bowling
{
    public class Game
    {
        public void Roll(int i){}

        public int Score { get { return -1; } }
    }
}
```



IT WORKS - WE CARE

TECHTALK

Step 5: Watch it fail

Unit Test Sessions - Session #1

Session #1 X

Group by: Projects and Namespaces

Tests failed: 2, passed: 0, ignored: 0

<Bowling.SpecFlow> (2 tests)	2 tests failed
{ } Bowling.SpecFlow (2 tests)	2 tests failed
BowlingFixture (2 tests)	2 tests failed
GutterGame	Failed: AssertionException: Expected:
AllStrikes	Failed: MissingMethodException: Meth

BowlingFixture.GutterGame : Failed

```
Given a new bowling game
When all of my balls are landing in the gutter
Then my total score should be 0
-> error:   Expected: 0
          But was:  -1
```



Step 6: Implement the domain

```
Game.cs | BowlingSteps.cs | Bowling.feature
Bowling.Game | isSpare(int frameIndex)
namespace Bowling
{
    public class Game
    {
        private int[] rolls = new int[21];
        private int currentRoll = 0;

        public void Roll(int pins)
        {
            rolls[currentRoll++] = pins;
        }

        public int Score
        {
            get
            {
                int score = 0;
                int frameIndex = 0;
                for (int frame = 0; frame < 10; frame++)
                {
                    if (isStrike(frameIndex))
                    {
                        score += 10 + strikeBonus(frameIndex);
                        frameIndex++;
                    }
                }
            }
        }
    }
}
```



Step 7: Watch it pass

Unit Test Sessions - Session #1

✔ Session #1 ✕

Tests failed: 0, passed: 2, ignored: 0

- ✔ <Bowling.SpecFlow> (2 tests) Success
- ✔ Bowling.SpecFlow (2 tests) Success
 - ✔ BowlingFixture (2 tests) Success
 - ✔ GutterGame Success
 - ✔ AllStrikes Success

BowlingFixture.GutterGame : Passed

```
Given a new bowling game
When all of my balls are landing in the gutter
Then my total score should be 0
```

BowlingFixture.AllStrikes : Passed

```
Given a new bowling game
When all of my rolls are strikes
Then my total score should be 300
```



IT WORKS - WE CARE

TECHTALK

Repeat!
Next Feature Please!



Welcome to the „real“ world ...

TC:110

@acceptance @TC110

Scenario: TC110 - Message with a rate for a currency that is not in the database (unknown currency)

Given the current time is '2009/09/15 13:52'

And there are currencies in the database with the following data:

code	
EUR	

And there are exchange rates in the database with the following data:

code	rate	date	error code	
EUR	1.2	2009/09/14	OK	

And the maximum exchange rate deviation is 2%

When there are exchange rates received for '2009/09/15' on EMB with the following data:

code	rate	
EUR	1.2	
SKK	1.3	

And wait until the exchange rate is processed

Then there is an exchange rate in the database with:

code	rate	date	error code	
EUR	1.2	2009/09/14	OK	

And there is no exchange rate in the database with:

code	date	
EUR	2009/09/15	

And an e-mail is sent by the CurrencyConverter service to 'error@import.info' (CC: 'error@import.info') with subject 'Unknown Currency'

And the e-mail should contain 'Error during import!'



What have we gained?

- Unified artifacts
 - Specification
 - Development aid
 - Automated test
 - Documentation
- Business readable and executable
 - Enable communication
 - Bound to implementation



What have we gained?

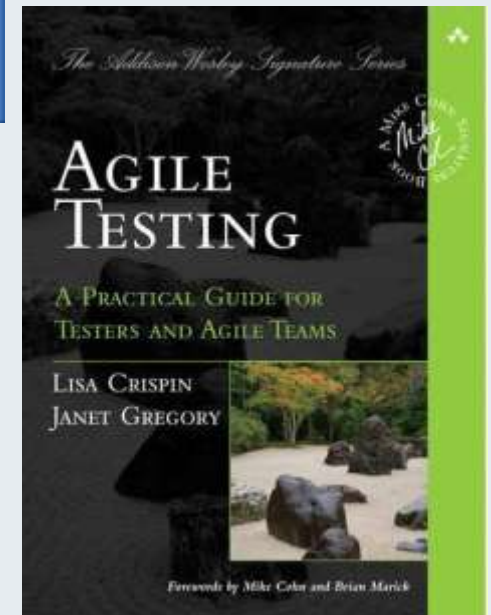
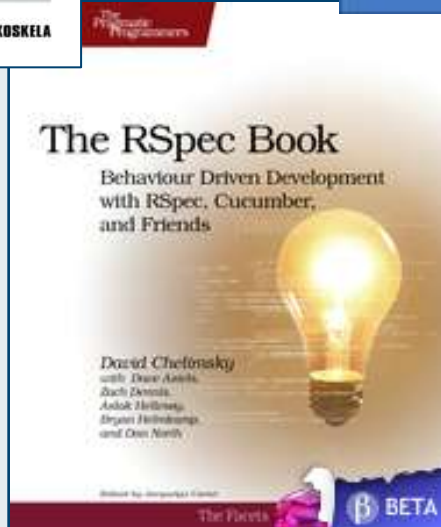
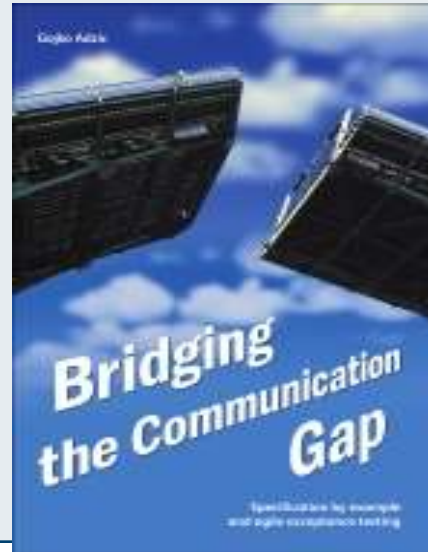
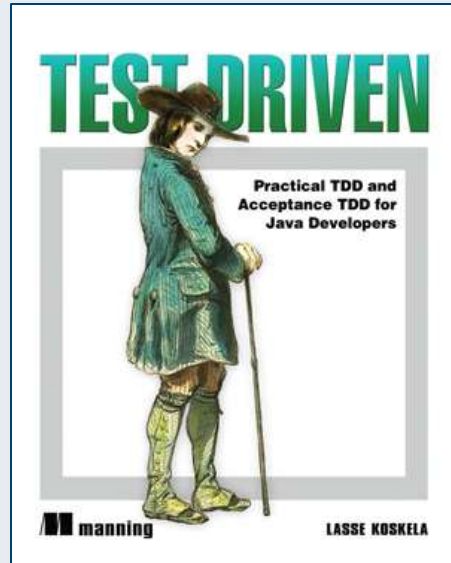
- Focus
 - Development is focused on business value
 - Implementing features is the main goal
 - The process is driving the development
- Traceability of requirements
 - Development activities are directly related to a feature
- Perfect match for User Stories



IT WORKS - WE CARE

TECHTALK

Books





IT WORKS - WE CARE

TECHTALK

Tools



www.cukes.info

Concordion

www.concordion.net

cuke4vs



www.fitnesse.org



www.specflow.org



<http://studios.thoughtworks.com/agile-test-automation>

JBehave

JBehave / NBehave



IT WORKS - WE CARE

TECHTALK

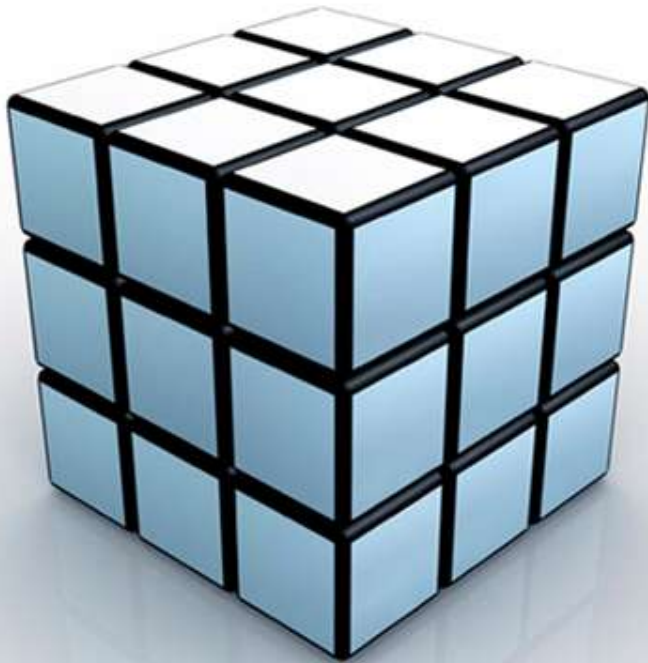
Remember

BDD is a Mindset not a Tool!



IT WORKS – WE CARE

TECHTALK



Fragen?

Christian Hassa
Christoph Seufert