

---

# Macheten für den Testdschungel

**Christian Heger**



# Agile Entwicklung – ohne Tests



**Mehr Tests  
– aber agiler?**





Fertig? Nee,  
fertig ist das  
nicht, da fehlen  
noch die Tests

# Der Code soll so sein wie der Code ist



```
1 [Fact]
2 public void App_should_crash_causing_maximum_confusion()
3 {
4     var subject = new Subject();
5
6     Asssert.Throws<NullReferenceException>(
7         () => subject.DoSomeWork());
8 }
```

```
1 [Fact]
2 public void Achieve_perfect_code_coverage()
3 {
4     var subject = CreateSubject();
5     subject.DoSomeWork();
6     Assert.NotNull(subject);
7 }
```

# Test First



AlmDays2012 - Microsoft Visual Studio

FILE EDIT VIEW NCRUNCH PROJECT BUILD DEBUG TEAM SQL TOOLS TEST ANALYZE WINDOW HELP

Test Explorer

- Streaming Video: Improving quality with unit tests
- Run All | Run...
- Slow > 1 sec (1)
- Medium > 100 ms (1)
- Fast < 100 ms (2)
  - ✓ Add\_1\_and\_1\_yields\_2 1 ms
  - ✓ Add\_1\_and\_2\_yields\_3 88 ms

Add\_1\_and\_1\_yields\_2

Source: Calculator.Tests.cs line 10

✓ Test Passed - Add\_1\_and\_1\_yields\_2

Elapsed time: 1 ms

```
1 using Xunit;
2 using FluentAssertions;
3
4 namespace Zuehlke.AlmDays2012.Demo
5 {
6     public class CalculatorTests
7     {
8         [Fact]
9         public void Add_1_and_1_yields_2()
10        {
11            var subject = new Calculator();
12
13            int result = subject.Add(1, 1);
14
15            result.Should().Be(2, "1 + 1 == 2");
16        }
17
18        [Fact]
19        public void Add_1_and_2_yields_3()
20        {
21            var subject = new Calculator();
22
23            int result = subject.Add(1, 2);
24
25            result.Should().Be(3, "1 + 2 == 3");
26        }
27    }
28 }
```

Solution Explorer

Solution 'AlmDays2012' (1 project)

- Demo
  - Properties
  - References
  - 010\_TestFirst
    - Calculator.cs
    - Calculator.Tests.cs
  - 020\_Isolation

Output Error List

Ready Ln 26 Col 11 Ch 11 INS



Halbe Stunde für  
das Feature...

STYLE & QUALITY  
MANUFACTURER EST. 1994  
68 FEAT. RED DINK BEIN, REN

READY

und dann muss ich  
die Integrations-  
Tests gerade ziehen.  
Also zwei Tage.



---

**Szenario: Kunde bekommt leckeren Kaffee**

**Wenn die „Kaffee“-Taste gedrückt wird**

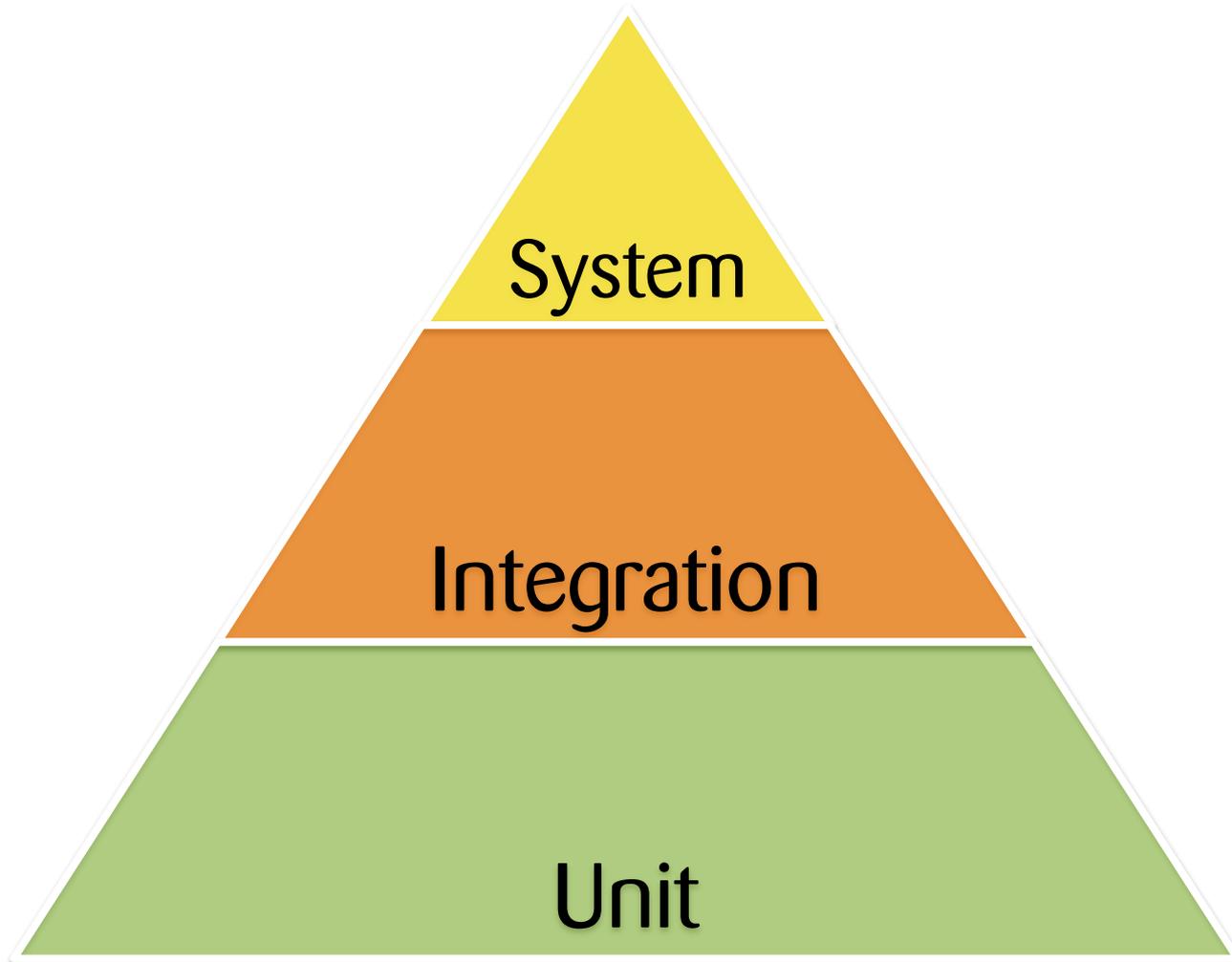
**Dann werden 15 Bohnen je nach gewähltem Mahlgrad gemahlen**

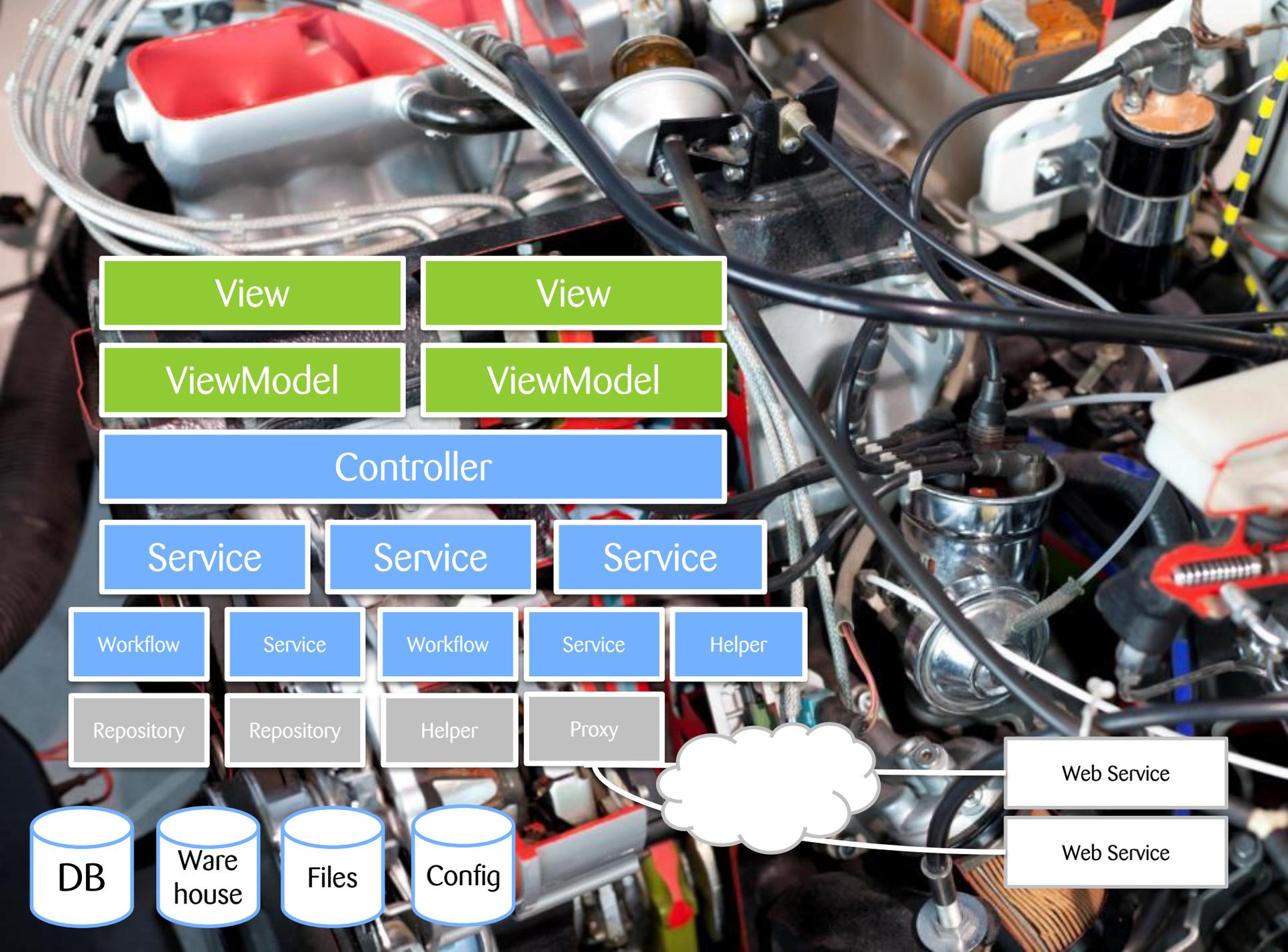
**Und werden 3.5g Kaffeepulver mit dem Kaffeepulverförderer in den Brühbehälter befördert**

**Und wird Wasserbehälter gepumpt, Spannung angelegt, und das Wasser auf 92°C erhitzt**

**Und die Tasse enthält leckeren Kaffee**

# Was teste ich da eigentlich?





View

View

ViewModel

ViewModel

Controller

Service

Service

Service

Workflow

Service

Workflow

Service

Helper

Repository

Repository

Helper

Proxy

DB

Ware house

Files

Config



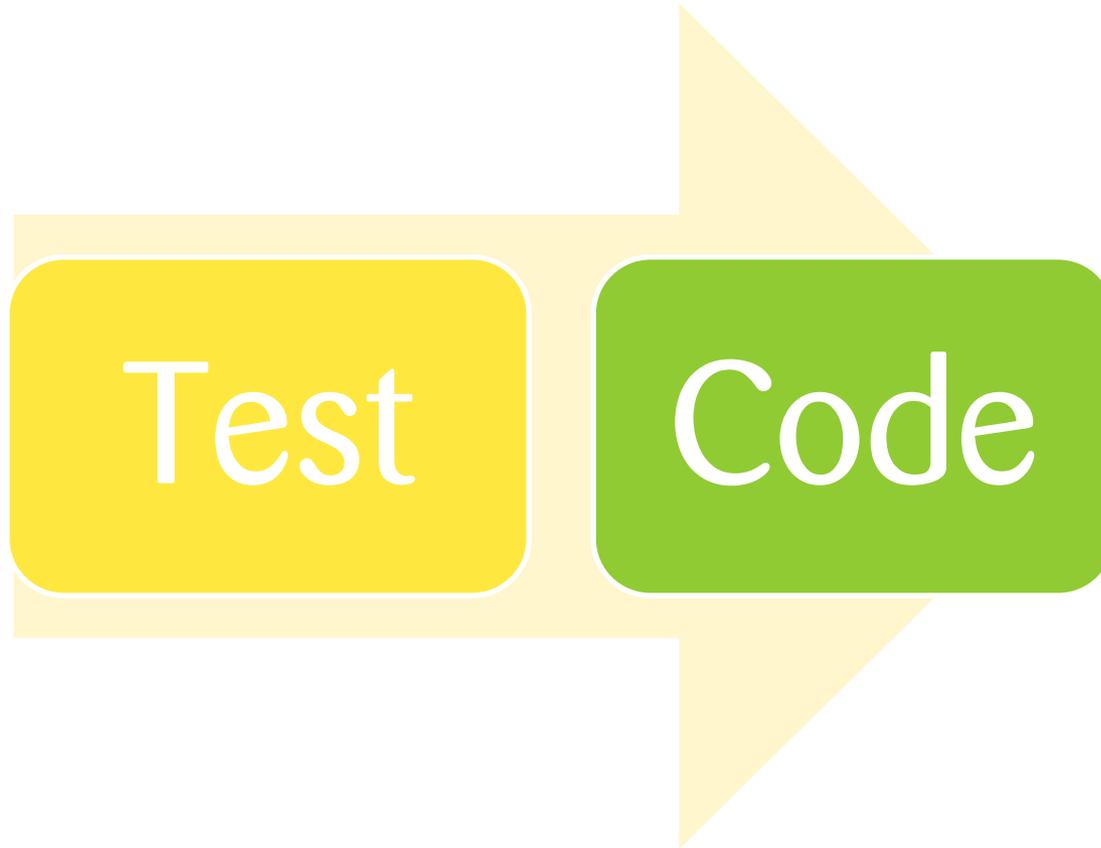
Web Service

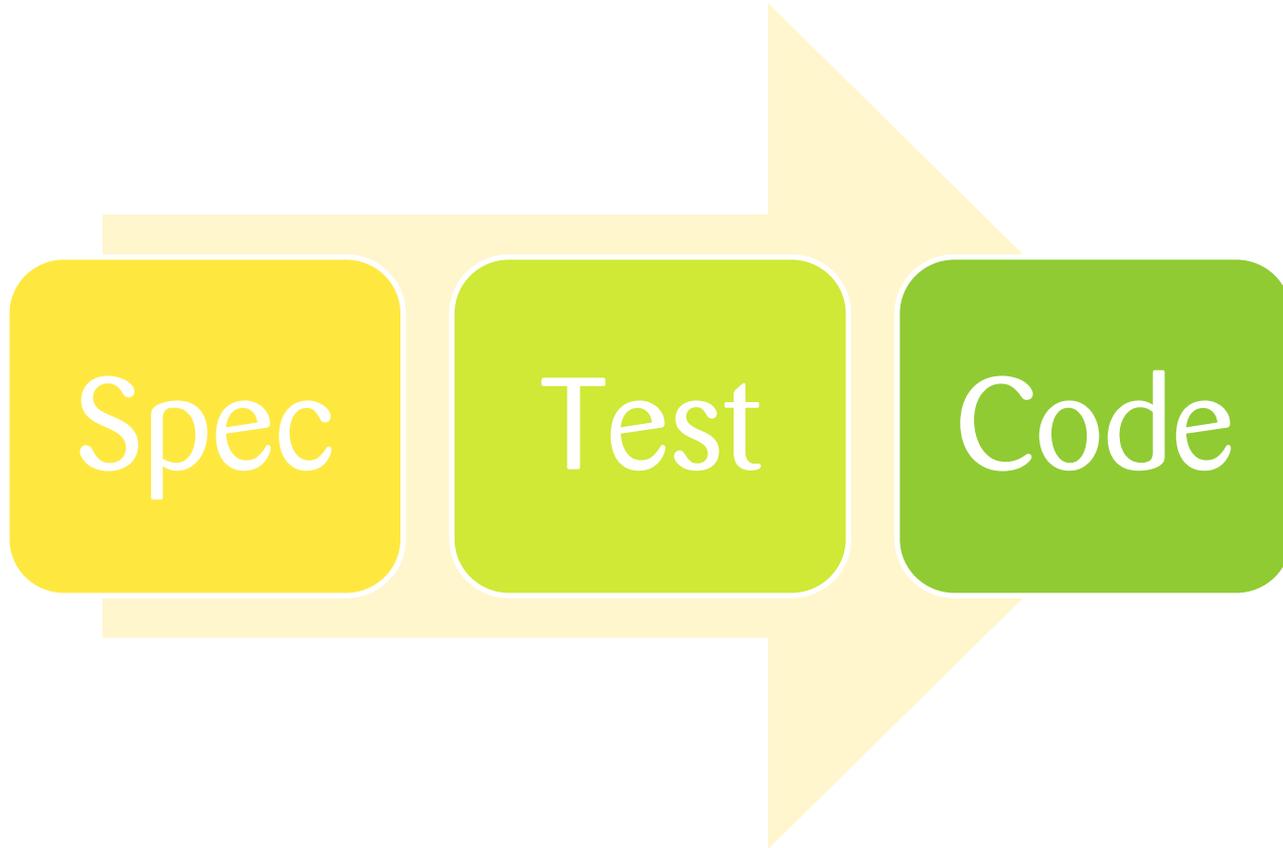
Web Service

# Segmentierung



	Module1	Module2	Module3
View	Green	Green	Orange
ViewModel	Green	Green	Orange
Controller			Orange
Services			Orange
Repository			Orange
Database			Orange





```
1 Funktionalität:  
2   Als Nutzer  
3   Möchte ich die Liste nach Datum sortieren  
4   Damit ich die letzten Bestellungen finden kann  
5  
6 Szenario: Die Liste enthält eine hohe Anzahl Einträge  
7   Gegeben sei: Die Beispieldaten "5000 Artikel" werden geladen  
8   Und die Liste ist nach Artikel sortiert  
9   Wenn der Button "Sortieren nach Datum" aktiviert wird  
10  Dann haben die Listeneinträge ein aufsteigendes Datum  
11  Und der Vorgang dauert nicht länger als 0.8 Sekunde
```

A man in a blue and white striped shirt is looking back over his shoulder towards two other men who are playing foosball. The man in the foreground has a speech bubble above him. The two men in the background are smiling and looking towards the camera. The foosball table is in the foreground, and the background shows a plain wall and a doorway.

Was wir hier  
machen?

Das Build läuft  
noch ne halbe  
Stunde.

# Keine Isolation langsamer Komponenten



```
1 [Fact]
2 public void ViewModel_displays_correct_data()
3 {
4     var dataSource = new DataSource();
5     var viewModel = new ViewModel(dataSource);
6     viewModel.Initialize();
7
8     viewModel.Name.Should().Be("Bob");
9 }
```

```
1 [Fact]
2 public void Save_to_database_works()
3 {
4     subject.SaveSquirrel(new Squirrel { Name="Bob" });
5 }
6
7 [Fact]
8 public void Read_from_database_works()
9 {
10    subject.GetSquirrel(x => x.Name == "Bob");
11 }
12
13 [Fact]
14 public void Delete_from_database_works()
15 {
16    subject.DeleteSquirrel(x => x.Name == "Bob");
17 }
```

# Quiz

Och, das ist bestimmt nur wieder so ein Test.

Das Build ist rot!

Ich starte das einfach noch mal, dann wird das schon.





---

## Visual Studio 2012

- Verbesserter Test Runner, auch für xUnit und NUnit
- Code Coverage
- Duplikatensuche (Code Clones)
- Tests nach Build ausführen



---

## NCrunch

- Lässt Tests sofort beim Schreiben laufen
- Visuelle, sofortige Test Coverage
- [www.ncrunch.net](http://www.ncrunch.net)

## SpecFlow

- BDD für .NET
- Erzeugt Unit Test Code aus Spezifikation + Schritt-Definition
- [www.specflow.org](http://www.specflow.org)

Fragen?  
Meinungen?  
Kommentare?



Christian Heger  
[about.me/christian.heger](https://about.me/christian.heger)