



BASTA!
.NET, WINDOWS, JAVASCRIPT

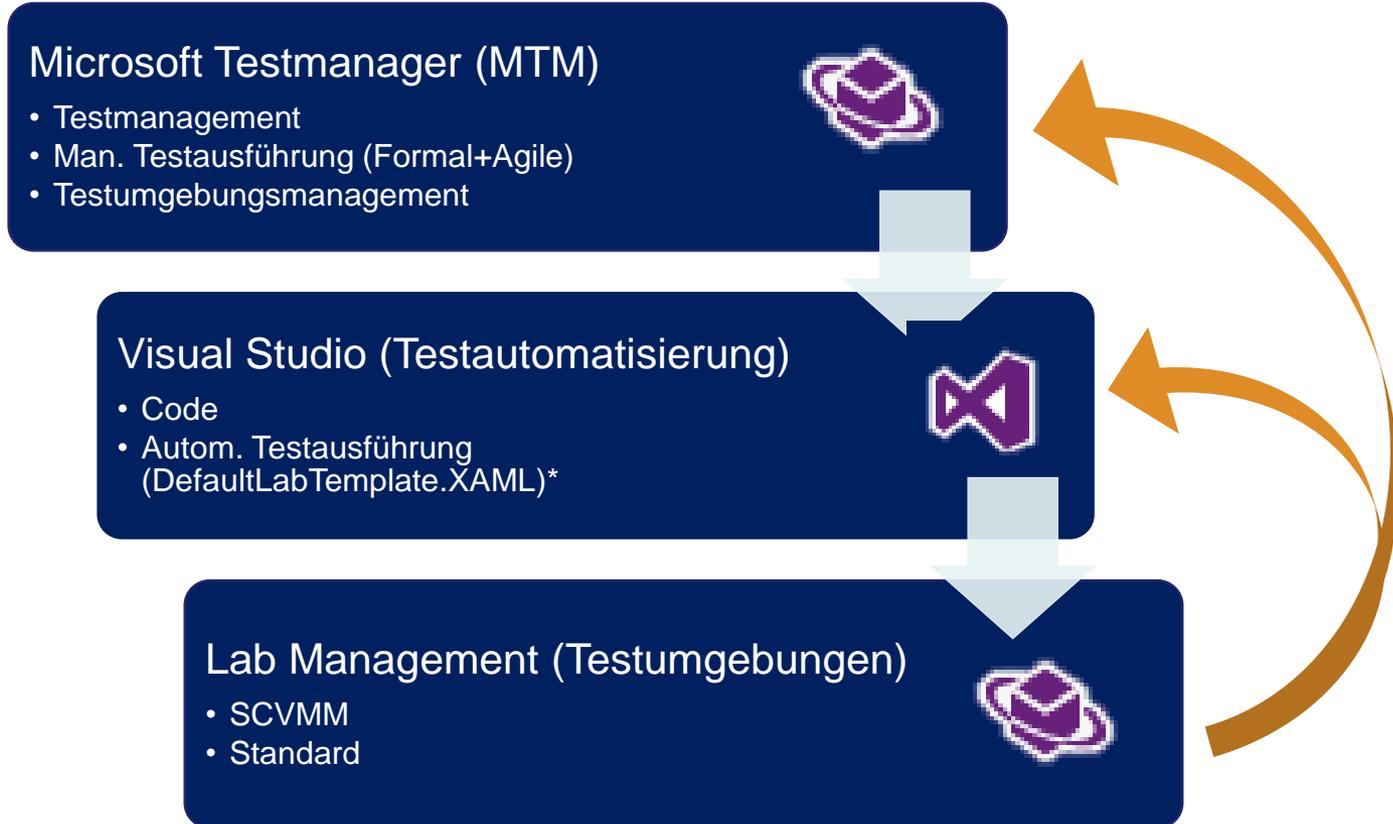
Nico Orschel | AIT
Marc Müller | 4tecture

Testen von XAML-based Store Apps mit VS 2013

Agenda

- Einleitung
- Manuelles Testen
- Unit Testing
- UI Testing
- Fazit

Testen mit VS und TFS



Testpyramide

UI Test Frameworks

UI

Unit Testing +
Isolation
Framework

Service

Unit Testing
Framework +
Isolation
Framework

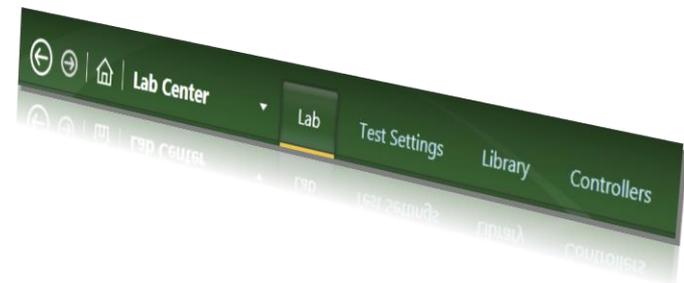
Unit / Component

Quelle: Mike Cohn

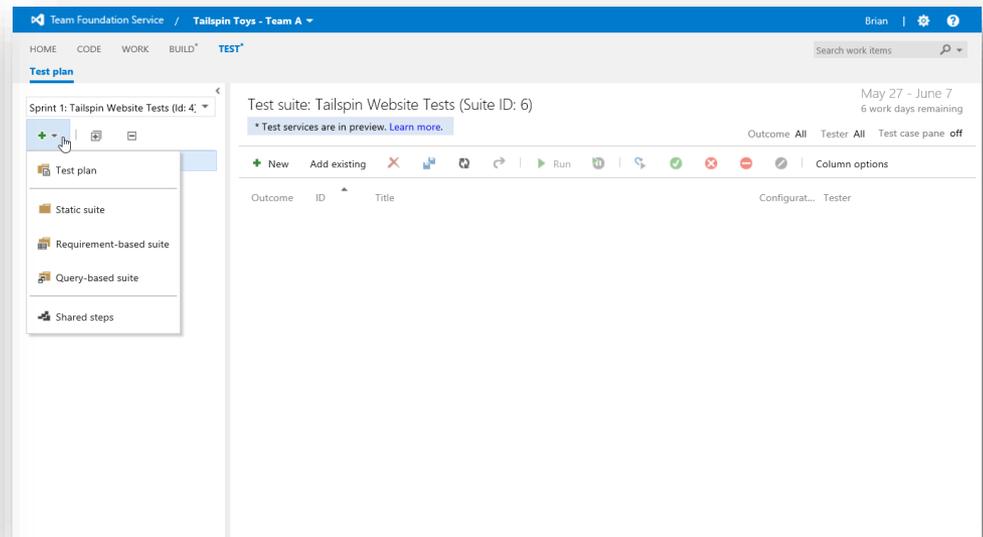
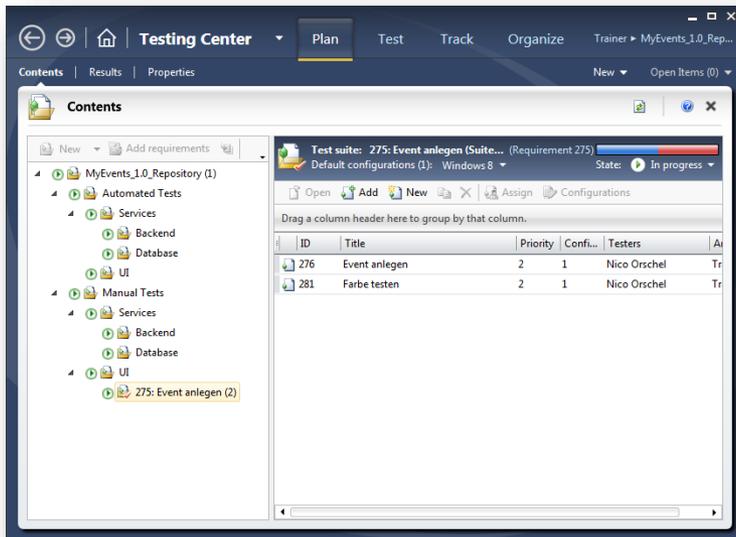
Manuelles Testen

Microsoft Test Manager

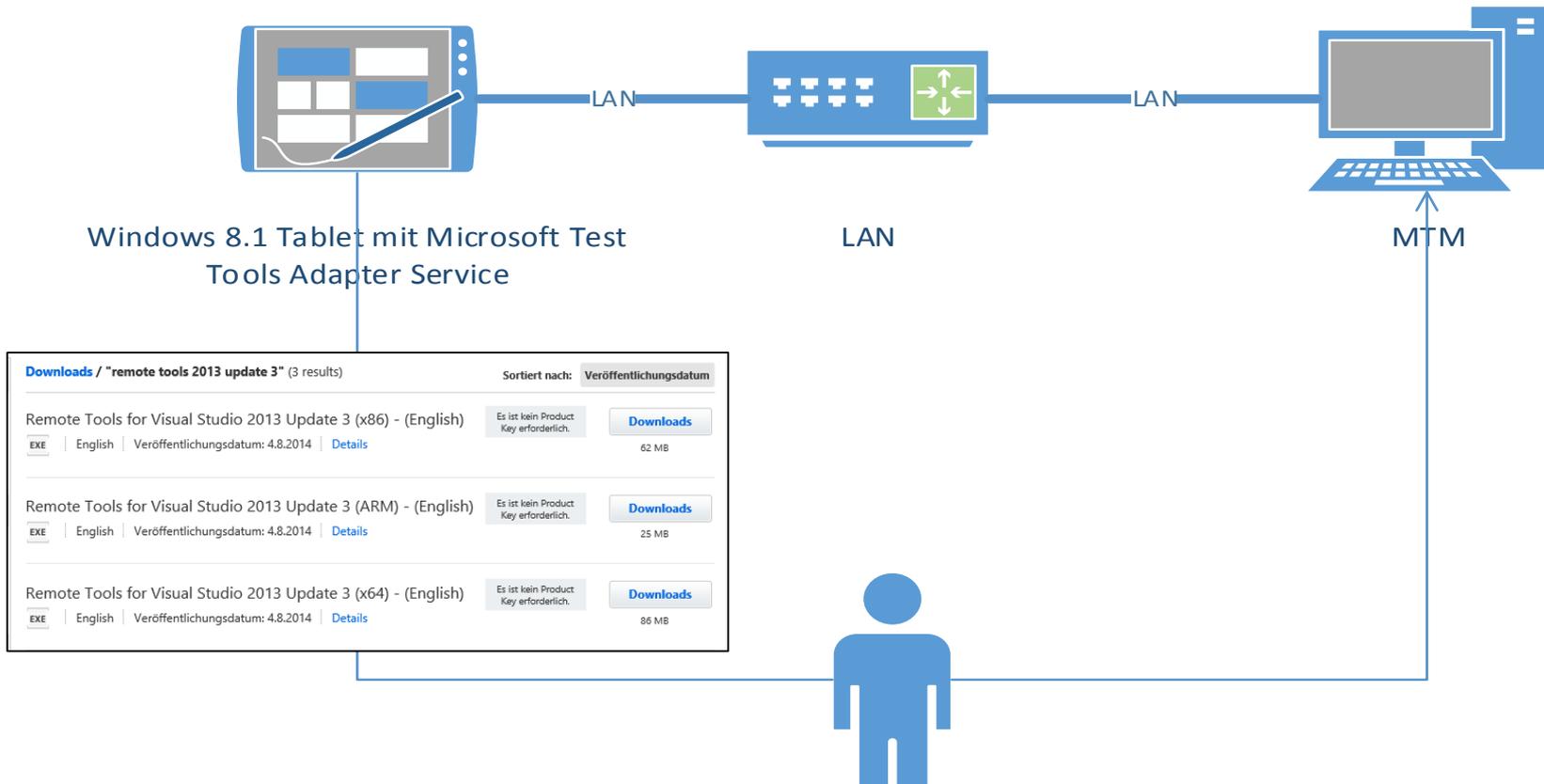
- Orientiert an Testeraufgaben
 - Testplanung
 - Testdurchführung
 - Testnachverfolgung
 - Umgebungsmanagement
- „Testautomatisierung“ für Oberflächen
- Test und Entwicklung enger zusammenbringen
 - „Rich Bugs“
 - Testvirtualisierung



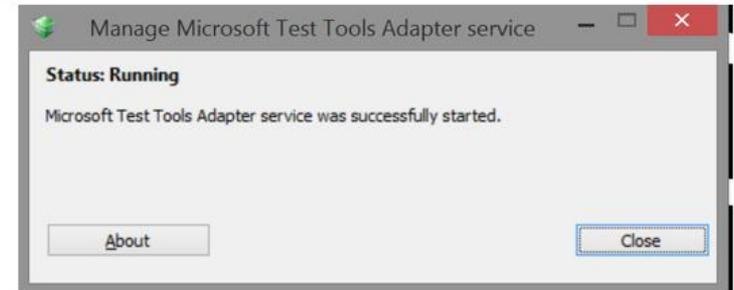
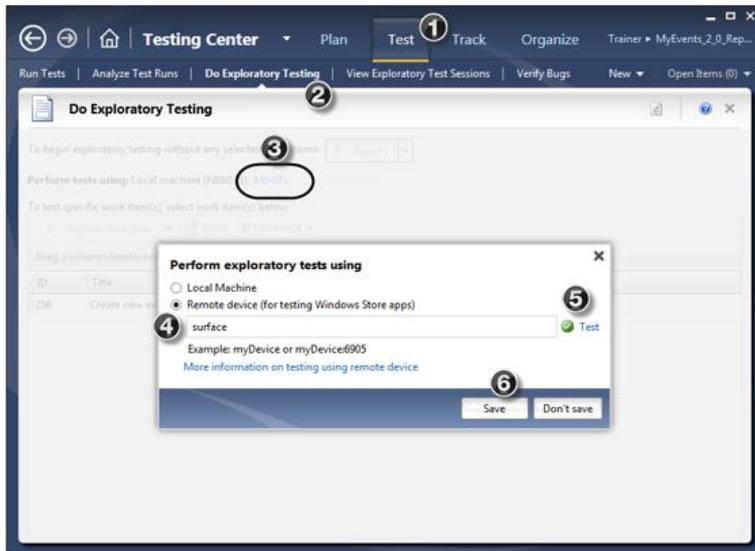
Die Qual der Wahl: MTM vs. Test Hub

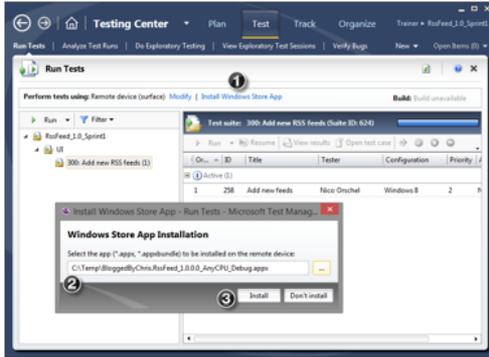


Infrastruktur



Infrastruktur





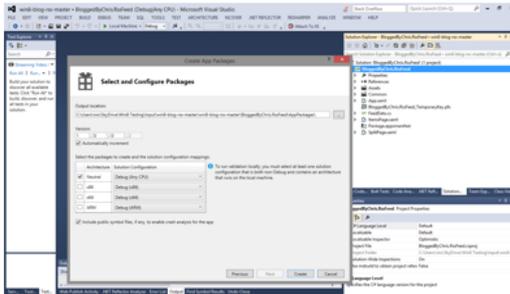
MTM

Win 8.1
Tablet



Visual
Studio /
TFS

MTM



Demo

Unit Testing

Unit Testing als Fundament

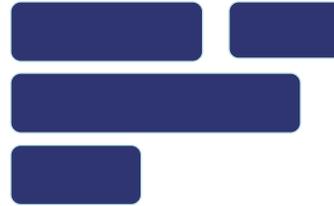
- Store Apps sind schnelllebig
- Kurze Releasezyklen
- Reaktion auf Kundenfeedback
- Refactoring

Red, Green, Refactor

Tests



Code



Refactor!

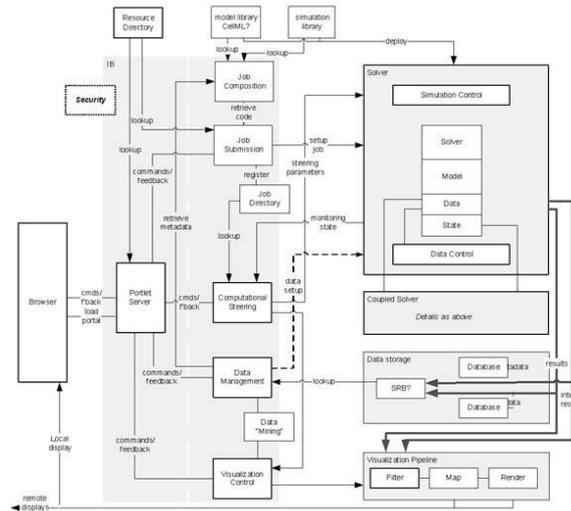


«Old fashioned» Big Design Up Front

All requirements known

- Requirement

Architecture Diagram



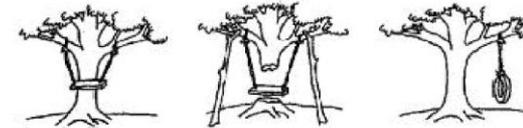
QS / Rollout



As proposed by the project sponsor.

As specified in the project request.

As designed by the senior architect.



As produced by the engineers.

As installed at the user's site.

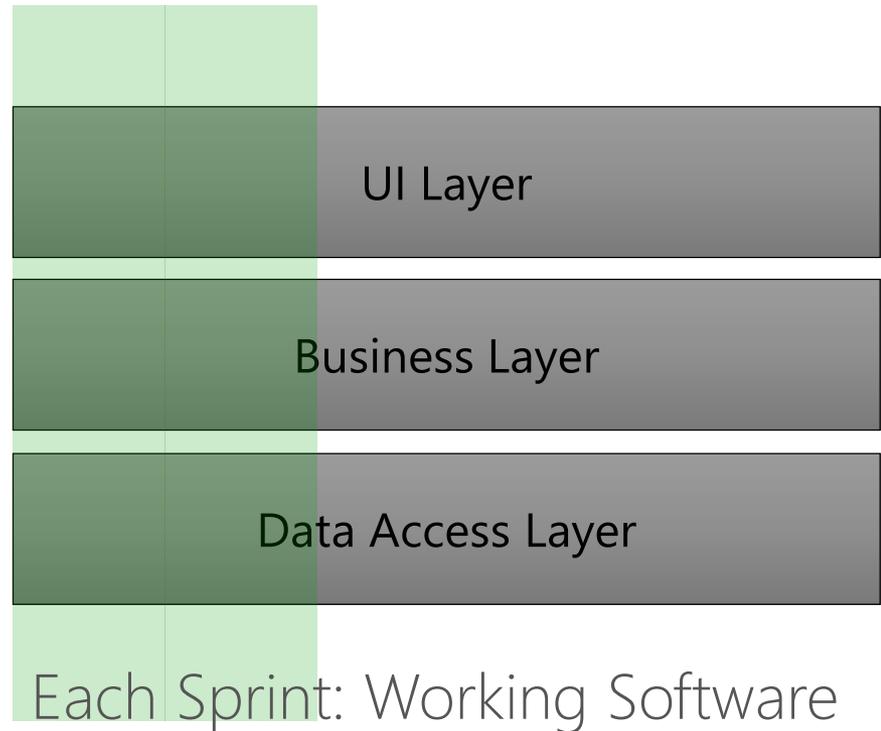
What the customer really wanted.

Source: <http://www.integrativebiology.ac.uk/softarch.html>

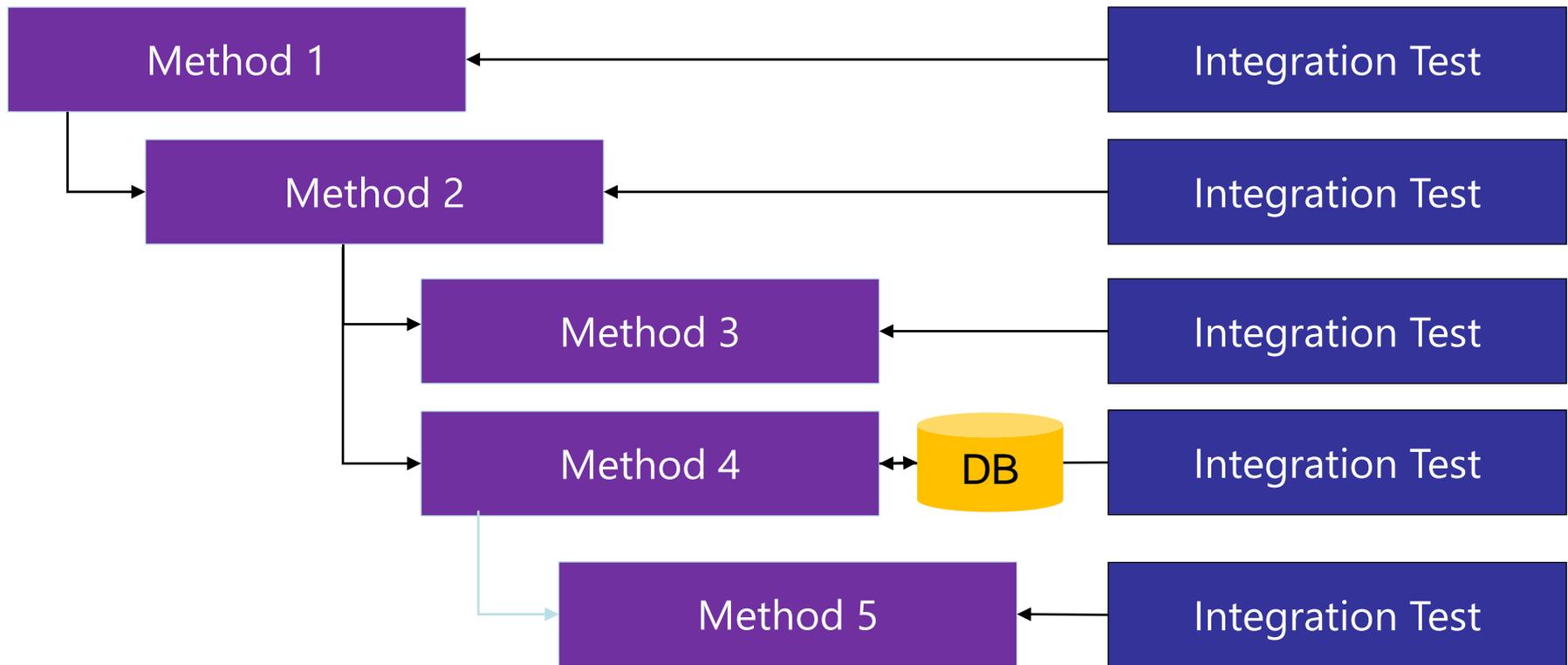
Source: <http://www.effectiveeng.com/en-030619.htm>

«Modern» Emergent Architecture

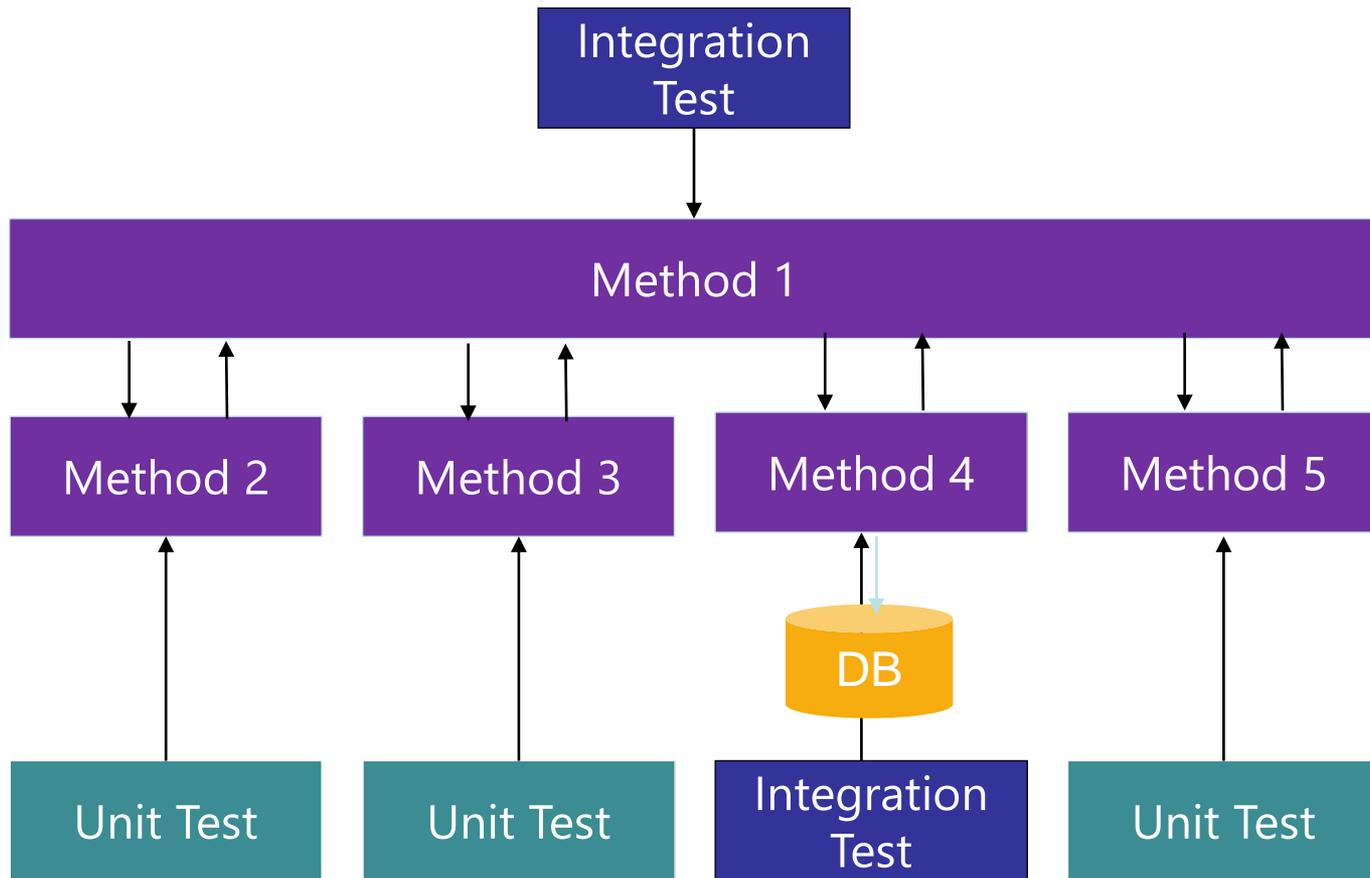
- Product Backlog Item



Classic Code Structure



Decoupled Code Structure



Unit Testing Checklist

- Dependency Injection
 - Dependencies auf Interfaces
 - Modularisierung
- MVVM
 - Binding
 - Commands
 - Behaviors
- Libraries: Prism, Caliburn Micro, etc.

Besonderheiten WinStore Apps

- Zugriff auf Windows Apps Sandbox über speziellen Test-Projekttyp
- Kein spezieller Test-Projekttyp für WinJS-basierende Apps
 - Alternative: Chuzpath JavaScript Testrunner + Referenzen auf WinJS Libraries

Asynchrone Methoden

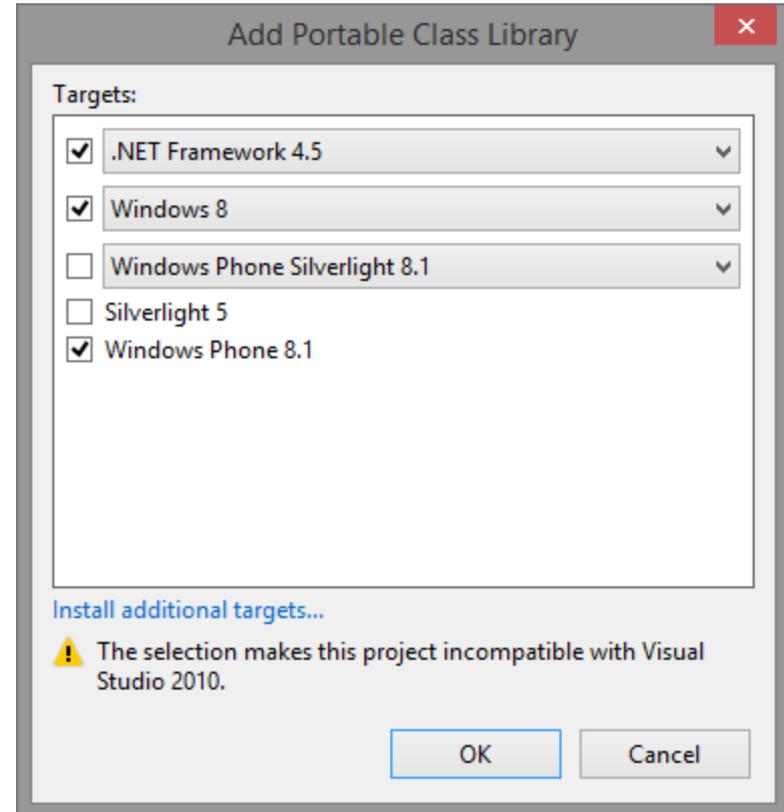
```
ComplexCalculator.cs  ComplexCalculatorTests.cs
C# MyBusinessLogic  MyBusinessLogic.Comple  MyBusinessLogic.Tests.Unit  MyBusinessLogic.Tests.Unit.Compl  ComplexCalculator_L

1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Linq;
5 using System.Text;
6 using System.Threading;
7 using System.Threading.Tasks;
8
9 namespace MyBusinessLogic
10 {
11     3 references
12     public class ComplexCalculator
13     {
14         1 reference | 1/1 passing
15         public async Task<int> Add(int number1, int number2)
16         {
17             return await new TaskFactory<int>().StartNew(() =>
18             {
19                 // simulating some long running stuff
20                 Thread.Sleep(50);
21                 return number1 + number2;
22             });
23         }
24         2 references | 2/2 passing
25         public async Task<int> Divide(int number1, int number2)
26         {
27             return await new TaskFactory<int>().StartNew(() =>
28             {
29                 // simulating some long running stuff
30                 Thread.Sleep(50);
31                 return number1 / number2;
32             });
33         }
34     }
35 }

26
27 [TestMethod]
28 | 0 references
29 public async Task ComplexCalculator_Devide_Ok()
30 {
31     // Arrange
32     var cc = new ComplexCalculator();
33     var expectedValue = 2;
34     var operand1 = 8;
35     var operand2 = 4;
36
37     // Act
38     var result = await cc.Divide(operand1, operand2);
39
40     // Assert
41     Assert.AreEqual(expectedValue, result);
42 }
43
44 [TestMethod]
45 | 0 references
46 [ExpectedException(typeof(DivideByZeroException))]
47 public async Task ComplexCalculator_DevideByZero_ThrownException()
48 {
49     // Arrange
50     var cc = new ComplexCalculator();
51     var operand1 = 8;
52     var operand2 = 0;
53
54     // Act
55     var result = await cc.Divide(operand1, operand2);
56
57     // Assert
58 }
59
60 }
```

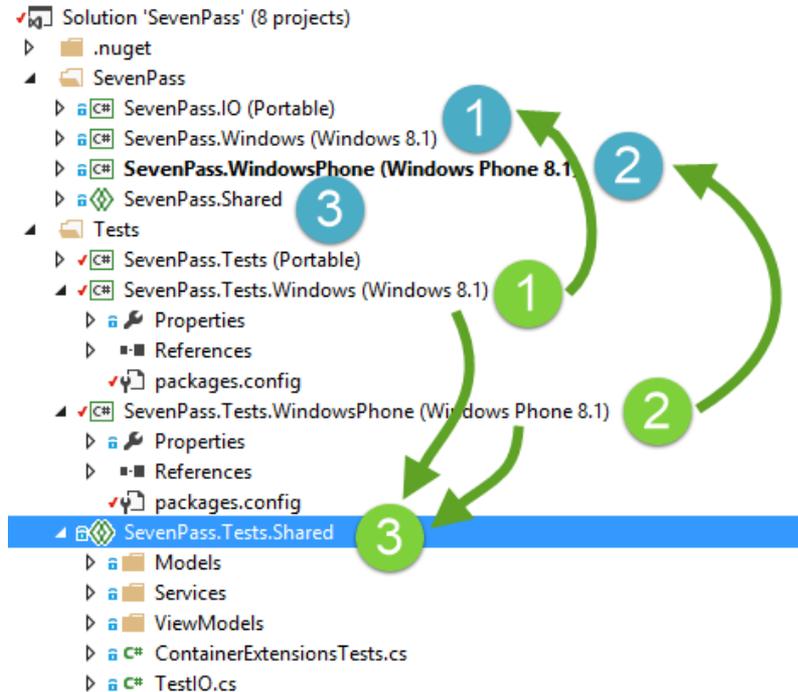
Portable Libs

- Shared DLL zwischen Store App, Phone App, etc.
- Auslagern der Logik, einfaches Unit Testen
- Reduzierter Befehlssatz



Code Sharing in Universal Apps

- Shared Code = Kopie zur Compile Zeit
- Unit Test Projekte pro Zielsystem



Demo

UI Testing

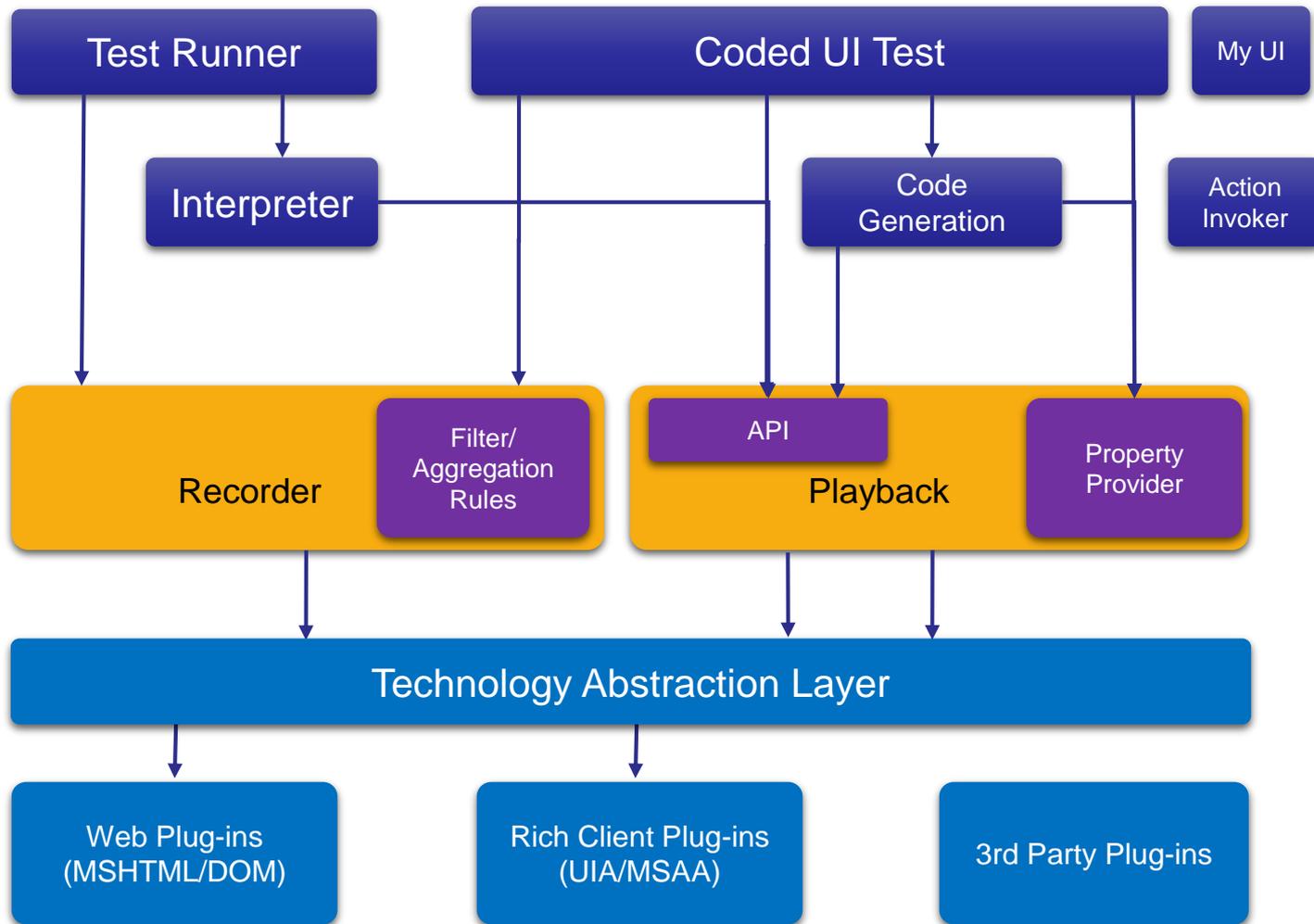
CodedUI

- CodedUI ist eine Testautomatisierungsplattform
- CodedUI unterstützt die meisten Windows UI Technologien
- Gleiche API für Rich Client und Web Applikationen

Technology	Support	Comment
IE 8-11	Yes	Older IEs are not supported anymore
WinForms 2.0, WPF 3	Yes	3rd party controls could be a problem
Win 8.1 Store Apps, Windows Phone 8.1 Apps	Yes	VS 2013, VS 2013 Update 2 (Phone)
Chrome, Firefox	Yes	Latest version via Selenium Adapter, (> VS 2012 Update 1, VS 2013)
Silverlight	No	Unofficial support for Silverlight v4/v5
Flash, Java, Mobile	No	e.g. Ranorex, Squish
Windows Win32 / MFC	Partially	May work with known issues
SharePoint	Yes	> SP 2007, > VS 2012 Update 1, VS 2013

Vollständige Liste
siehe
<http://msdn.microsoft.com/en-us/library/vstudio/dd380742.aspx>

CodedUI Architektur



3 CodedUI Arten

Record and Playback



- Default Approach
- Poor Code Generation
- Poor Maintainability

Record UI Maps



- Better Maintainability
- Supports large test repositories

Hand Code Tests



- Expert Mode
- Cleanest Automation Model

Demo

Zusammenfassung

- Zugriff auf Tablet (x86 + ARM) mit MTM
- Unit Testing als Fundament
 - Unit Testing Support für alle App-Typen verfügbar
- UI Testing Support für XAML Apps
 - Unterstützung Emulator und Hardware
 - Umfangreicher Gesture-Support

Wir unterstützen Sie

KONTAKT

Nico.Orschel@aitgmbh.de
+49 151 55052624



AIT GmbH & Co. KG

Leitzstr. 45, 70469 Stuttgart
www.aitgmbh.de

BERATUNG

Agile ALM und TFS
.NET und Architektur



ENTWICKLUNG

Dienstleister für individuelle
Lösungen mit .NET und Azure



© AIT GmbH & Co. KG – Alle genannten und gezeigten Marken oder Warenzeichen sind eingetragene Marken oder eingetragene Warenzeichen ihrer jeweiligen Eigentümer und ggf. nicht gesondert gekennzeichnet. Aus dem Fehlen der Kennzeichnung kann nicht geschlossen werden, daß es sich bei einem Begriff oder einem Bild nicht um eine eingetragene Marke oder ein eingetragenes Warenzeichen handelt.

4tecture[©]

empower your software solutions

4tecture GmbH
Thurgauerstrasse 40
CH-8050 Zürich
www.4tecture.ch

Marc Müller
Principal Consultant
mmueller@4tecture.ch
[@muellermarc](https://twitter.com/muellermarc)