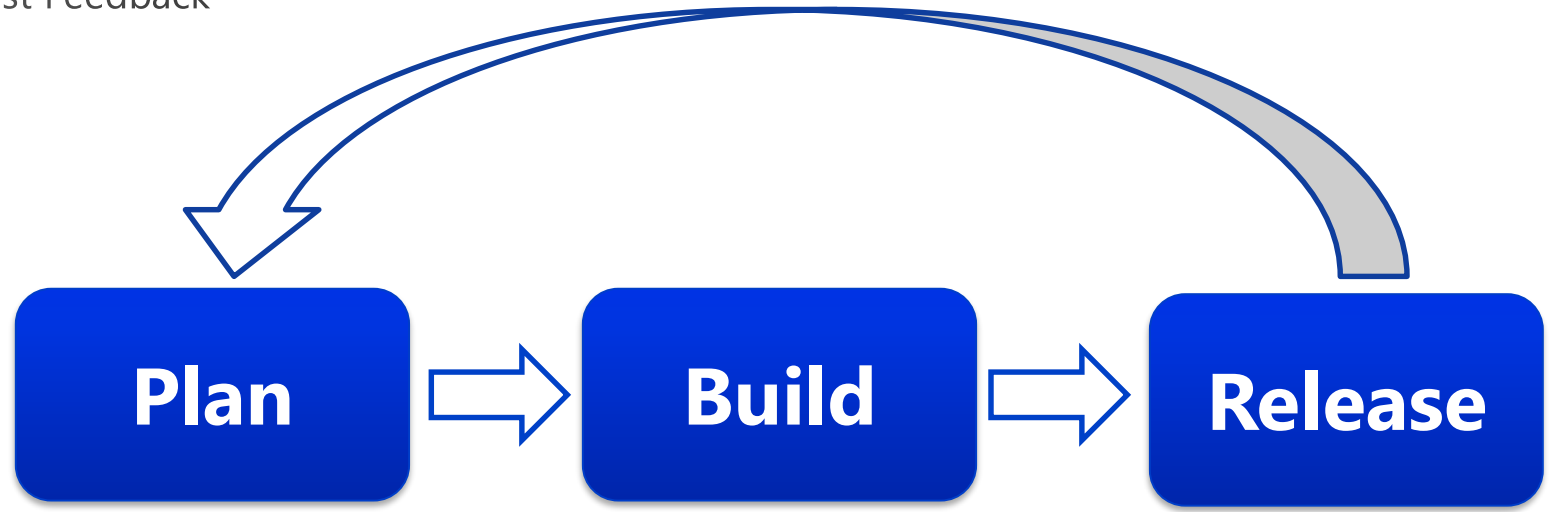
CodedUI – Gut vorbereitet ist halb getestet

Nico Orschel, MVP @ AIT, DE
Marc Müller, MVP @ 4tecture, CH

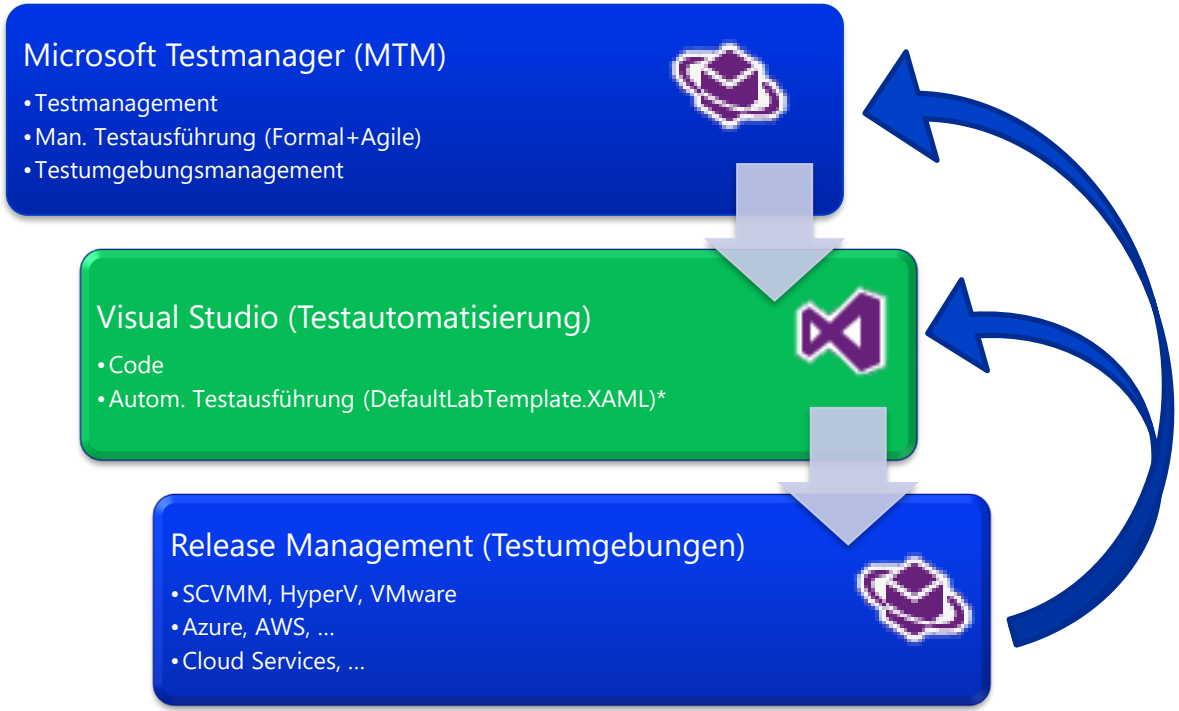
Warum Testautomatisierung

Ziele der modernen Software-Entwicklung

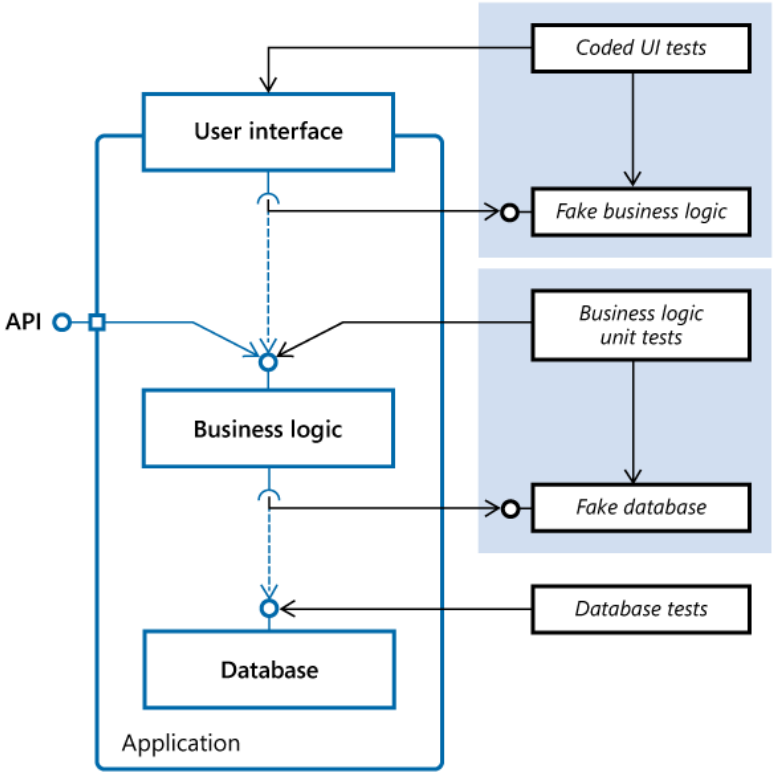
- Cycle-Times reduzieren
- Fast Feedback



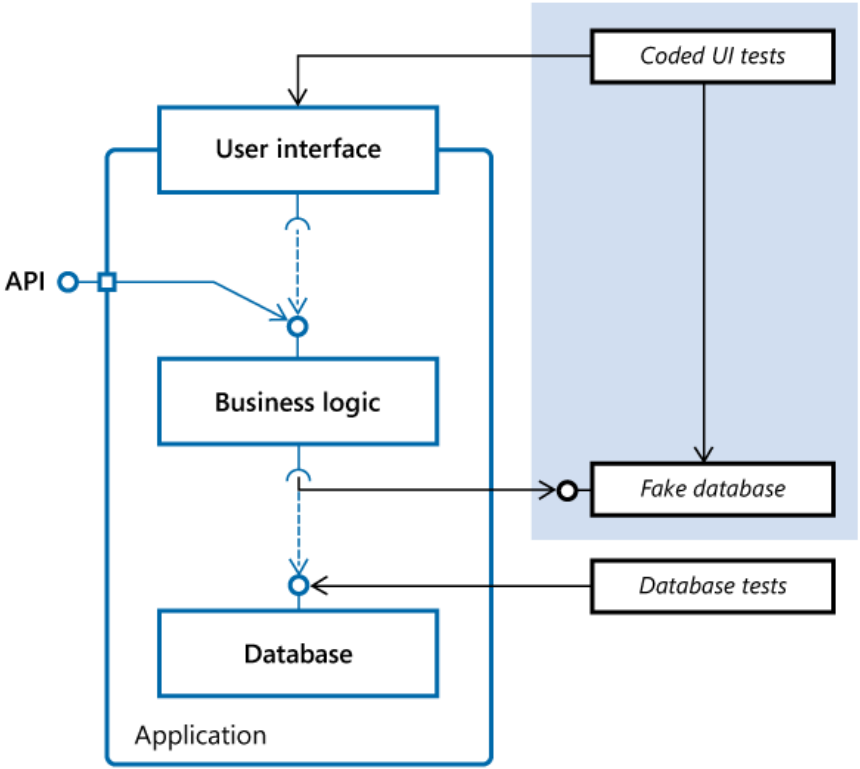
Wo sind wir in der TFS/VS Welt?



Testing auf unterschiedlichen Ebenen



CodedUI in kleinen Projekten



Warum CodedUI

CodedUI ist eine Testautomatisierungsplattform

CodedUI unterstützt die meisten Windows UI Technologien

Gleiche API für Rich Client und Web Applikationen

Running Coded UI Tests

Ausführung möglich über:

- Visual Studio Test Explorer
- Lab Management (Standard + SCVMM Env.) / Release Management
- MTM
- Build

Weitere Informationen

- http://blog.aitgmbh.de/2010/09/13/codedui-tests-ohne-lab-management-ausfhren/

Unterstützte Plattformen

Technology	Support	Comment
IE 8-11	Yes	Older IEs are not supported anymore
WinForms 2.0+, WPF	Yes	3rd party controls could be a problem
Windows Store Apps	Yes	XAML based Store Apps
Chrome, FireFox	Yes (> VS 2012 Upd. 4)	Latest version via Selenium Adapter
Silverlight	No	Unofficial support for Silverlight v4/v5 (VS 2010 / 2012 only)
Flash, Java	No	3rd Party-Vendor like Ranorex
Windows Win32 / MFC	Partially	May work with known issues
SharePoint	Yes (> VS 2012 Upd. 2)	2007+
Windows 8.1 Store Apps	Yes	>= VS 2013
Windows Phone 8.1. Apps	Yes	>= VS 2013 Update 2

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

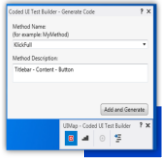
UI-Ansteuerungstechnologien

Technology	UI Test Implementation Model
Windows Forms	Microsoft Active Accessibility (MSAA)
Windows Presentation Foundation	UI Automation (UIA)
Internet Explorer	MSHTML
Firefox (VS 2010)	JavaScript and Firefox DOM
Firefox / Chrome (VS 2012+)	Selenium
Silverlight	Code Injection and reflection

Multiple ways to create UI tests

Record and Playback

- Default Approach
- Poor Code Generation
- Poor Maintainability



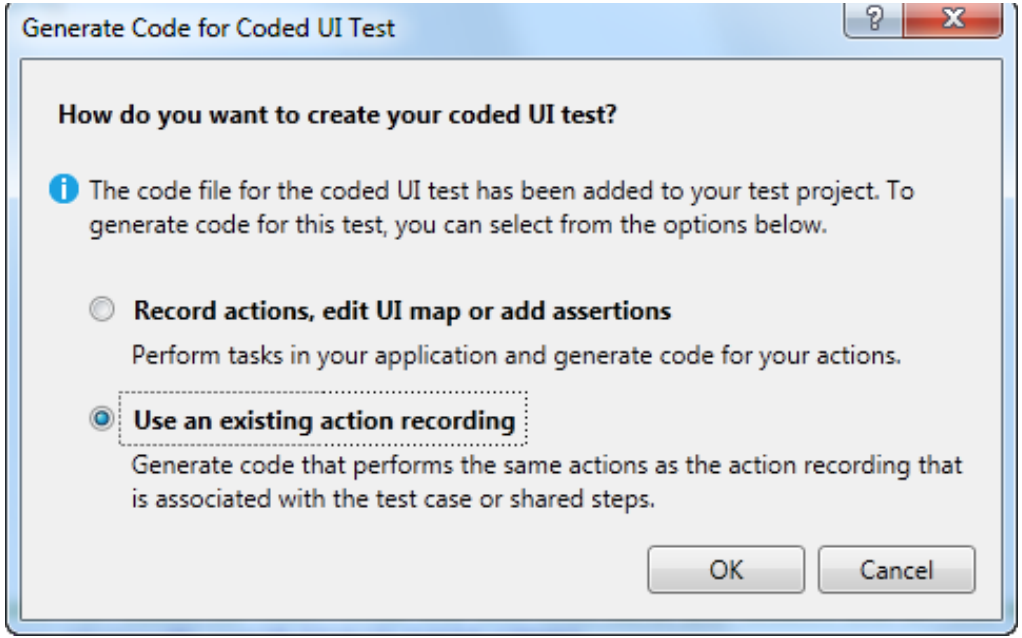
Record UI Maps

- Better Maintainability
- Supports large test repositories

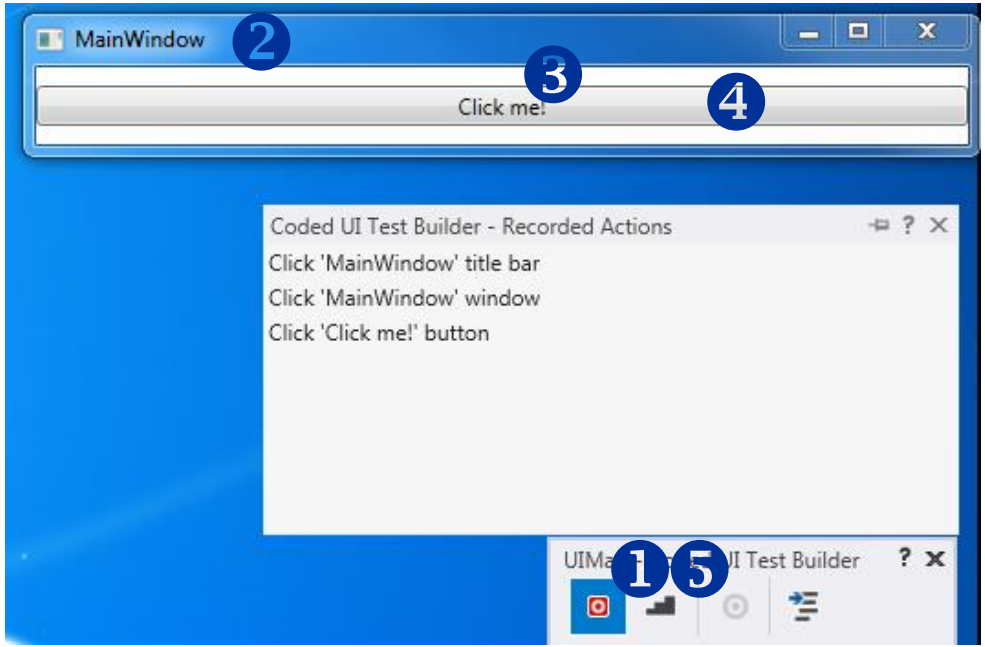
Hand Code Tests

- Expert Mode
- Cleanest Automation Model

How-to generate CodedUI Tests?

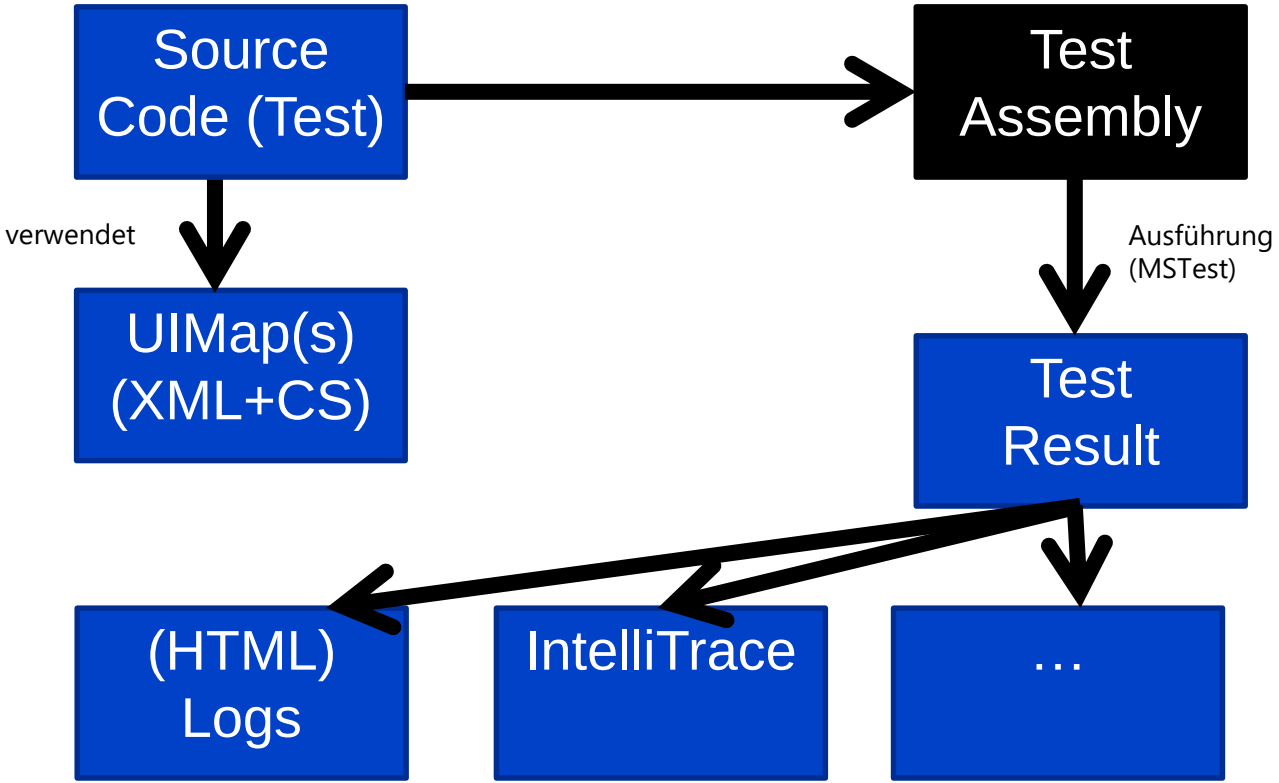


Recording Test Steps



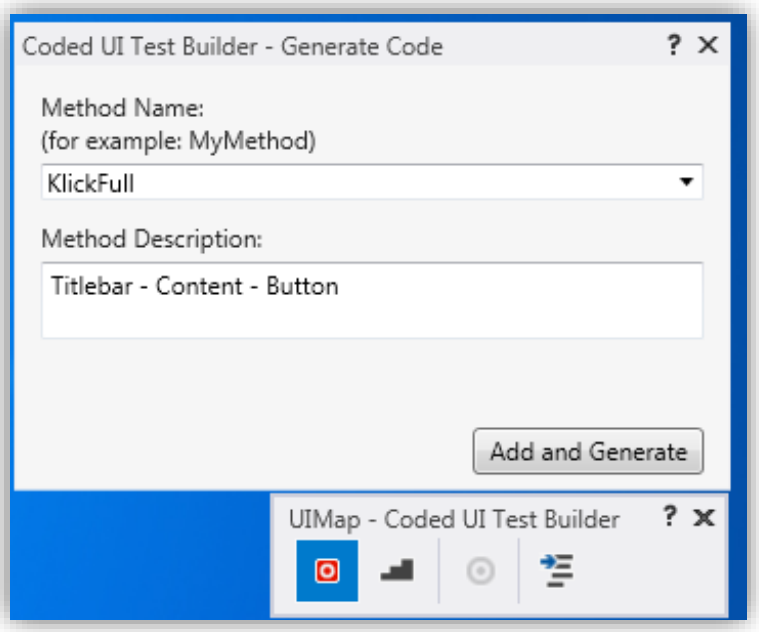
Klick

Understanding File Relationships

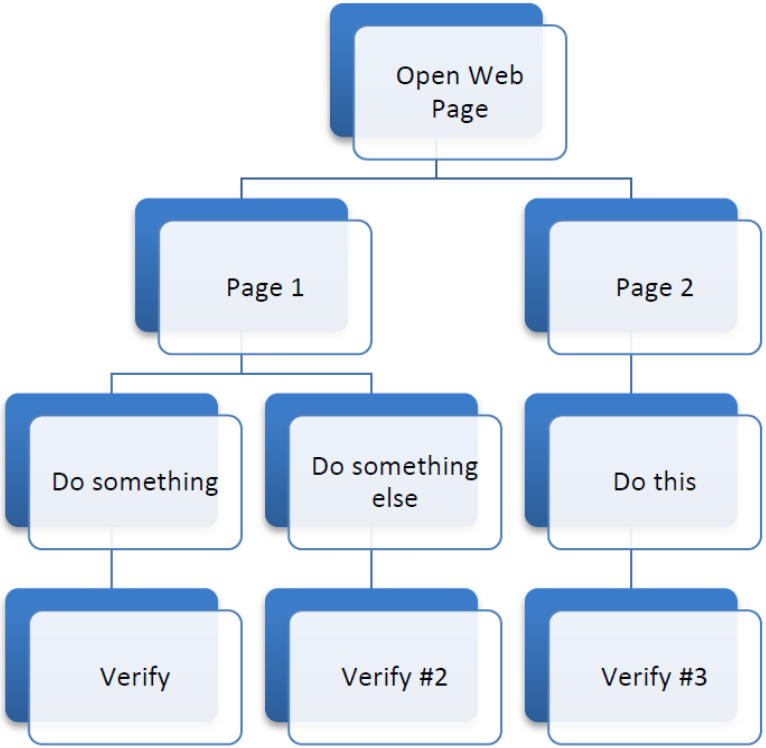


Einsatz der Recorder Controls

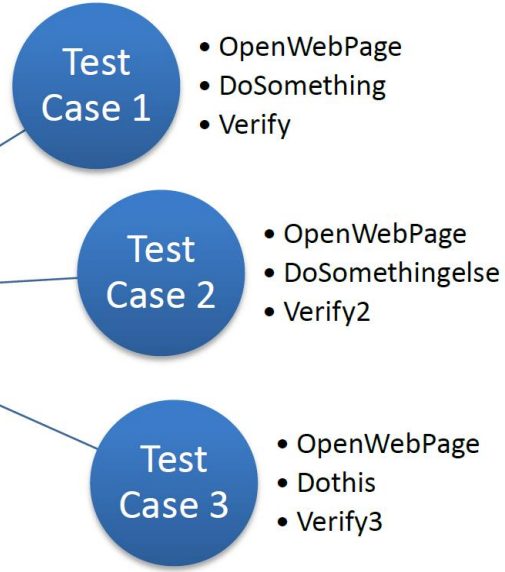
- Aufnahme starten/ pausieren
- Aufgenommene Schritte anzeigen
- Aussagen hinzufügen
- Code generieren



Mehrere UIMaps – Modulares Design




```
[CodedUITest]
public class MyCodedUITest {

    [TestMethod]
    public void TestCaseOne()
    {
        var util = new TestUtility();

        util.OpenWebPage();
        util.NavigateToPageOne();
        util.PageOne.DoSomething();
        util.PageOne.Verify();
    }

    [TestMethod]
    public void TestsCaseTwo()
    {
        var util = new TestUtility();

        util.OpenWebPage();
        util.NavigateToPageOne();
        util.PageOne.DoSomethingelse();
        util.PageOne.VerifyTwo();
    }

    [TestMethod]
    public void TestCaseThree()
    {
        var util = new TestUtility();

        util.OpenWebPage();
        util.NavigateToPageTwo();
        util.PageTwo.Dothis();
        util.PageTwo.VerifyThree();
    }
}
```



Quelle: Coded UI Testing Guide

Vorteile Page Object Pattern

Anwendung der SE Prinzipien

- Wartbare Tests
- SOLID, DRY, …

Einfache Lesbarkeit der Szenarios

Test ist fokussiert auf Interaktion, kein «Plumbing»

- Separation of Concerns
- UI Interaktion ist abstrahiert
- Kann mittels UIMaps oder «Code First» implementiert werden

Code First – ohne UIMaps

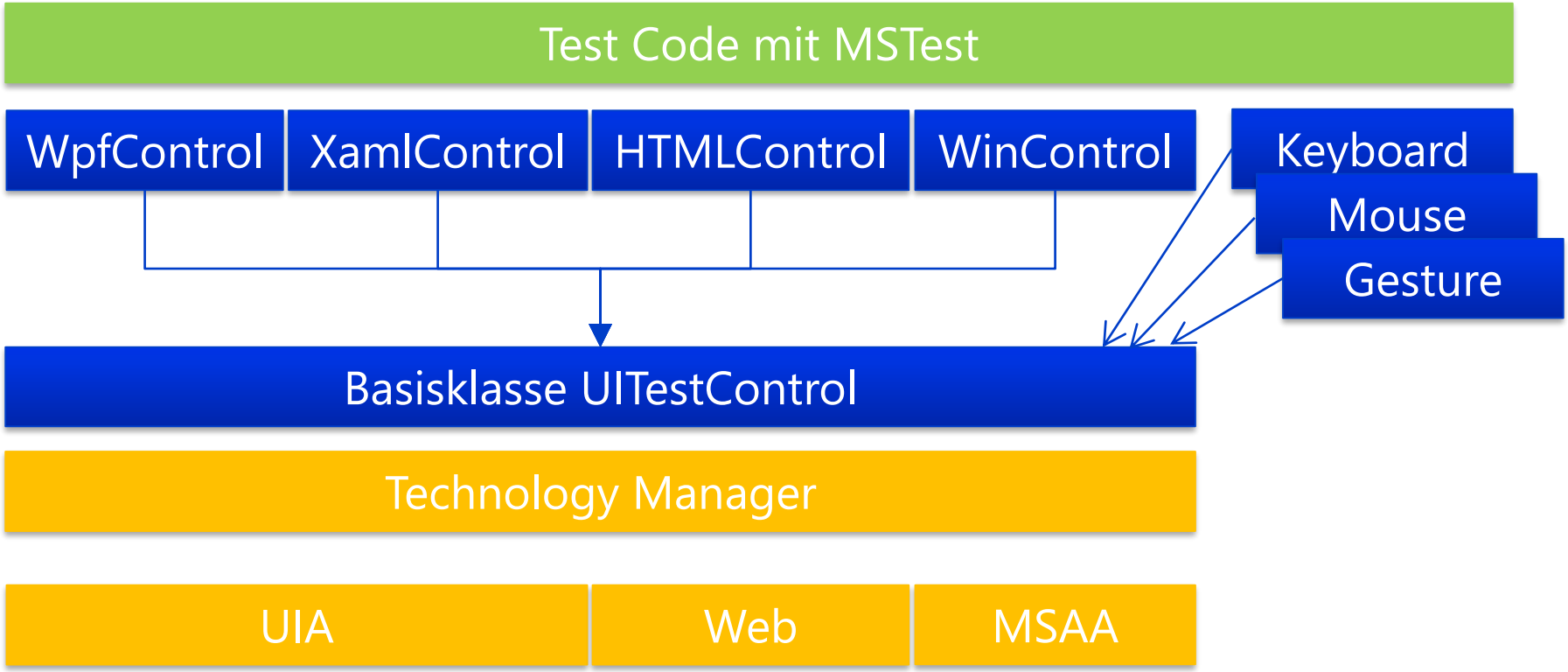
Code First Vorteile:

- Volle Kontrolle über den Code
- Einfaches Merging und Refactoring
- Bessere Team Collaboration Möglichkeiten

Code First Nachteile:

- Aufwändiger, es muss mehr Code geschrieben werden
- Detaillierte Kenntnisse über CodedUI Object Model notwendig

Technologiespezifische Umsetzung



UI Automation Controls

CodedUI abstrahiert die darunterliegende UI Technologie

- Alle Automation Controls sind von UITestControl abgeleitet.
- Ein Technology Manager lädt die richtige Implementation um mit der effektivsten Technology zu interagieren.
- Die UITestControl Klasse enthält ein TechnologyName Property

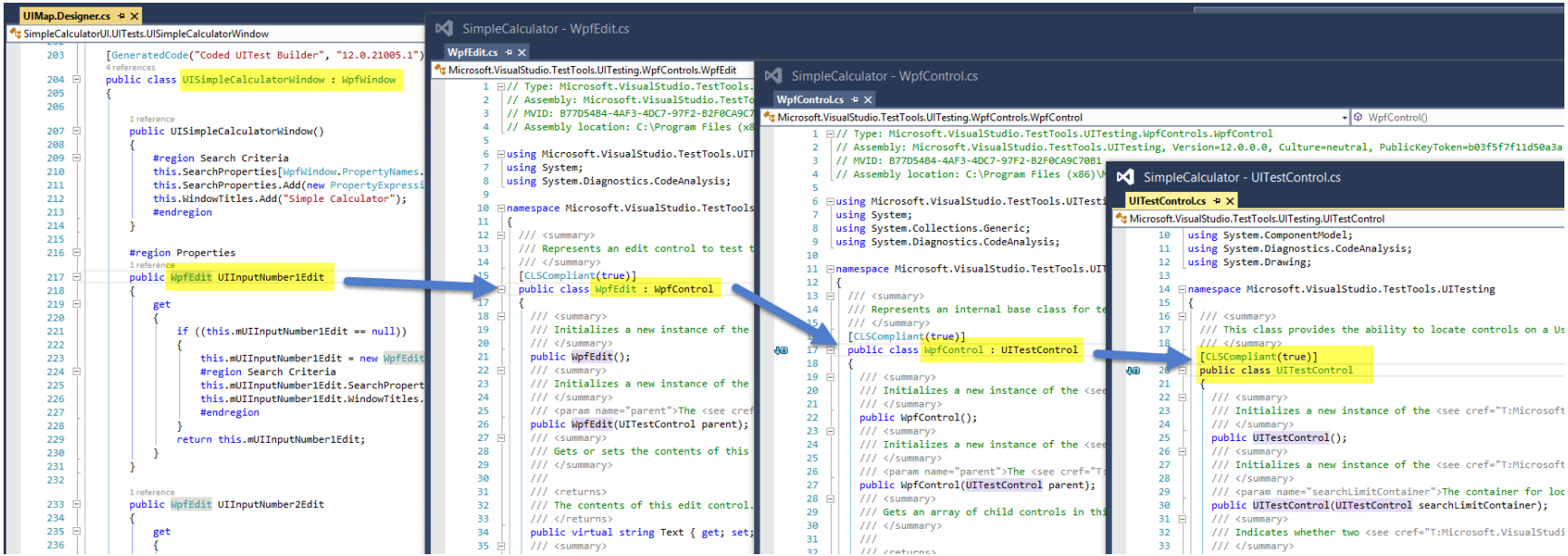
Technologieabhängige Implementierung:

- WinTable
- WpfTable
- XamlTable
- HTMLTable

Sind CLR-Objekte

Lassen sich über Properties / Methoden ändern

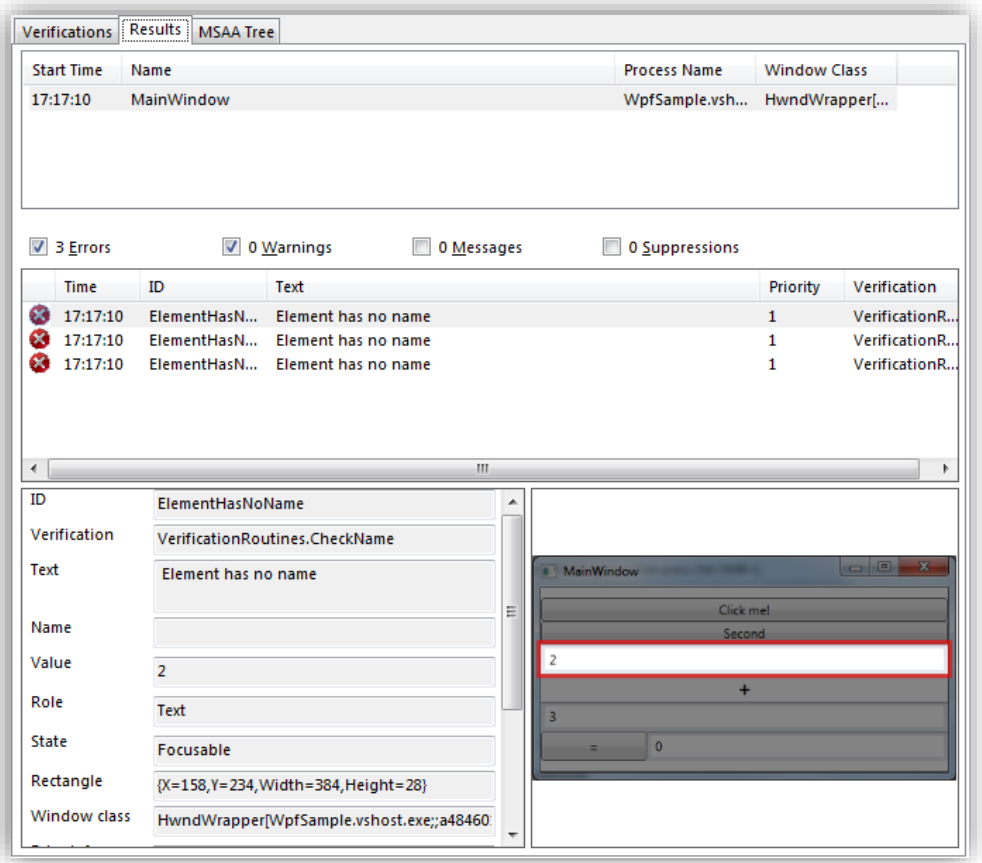
UI Automation Controls



Testautomatisierung ohne UIMaps

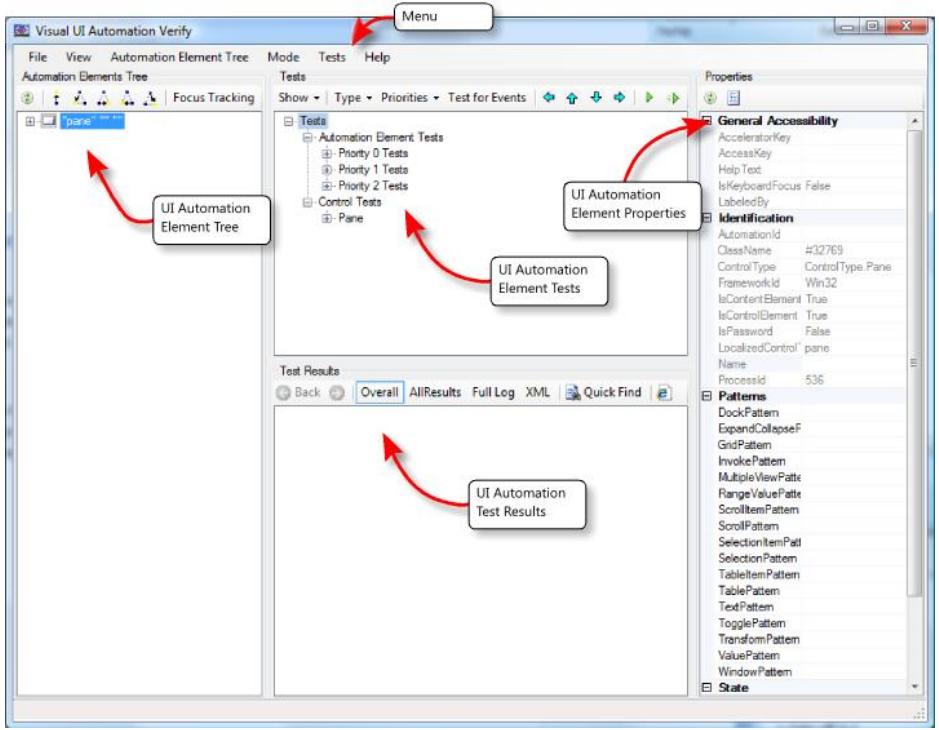
```
public virtual WpfControl FindControlById(WpfControl parentControl, string id)
{
    WpfControl control = new WpfControl(parentControl);
    control.SearchProperties[WpfControl.PropertyNames.AutomationId] = id;
    return control;
}

...

public void ClickManageEventsButton()
{
    WpfControl button = this.FindControlById(this.MyEventsMainWindow,
        "ManageEventsButton");
    Point clickPoint = new Point();
    button.TryGetClickablePoint(out clickPoint);
    Mouse.Click(button, clickPoint);
}
```

ACChecker



4tecture AIT

UIAVerify

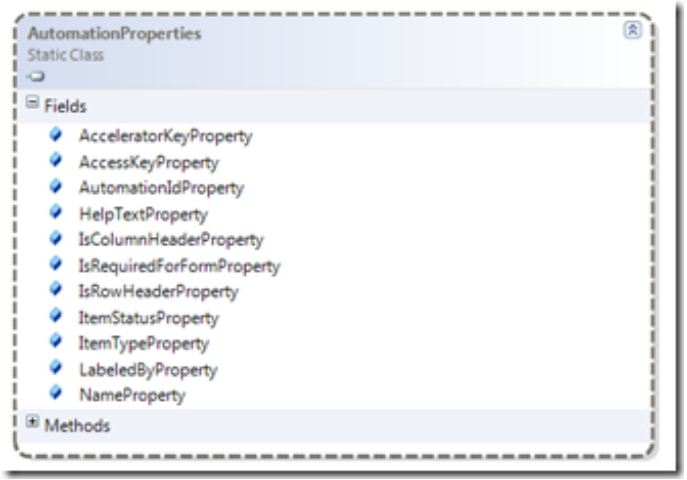

Menu
Visual UI Automation Verify
File View Automation Element Tree Mode Tests Help
Automation Elements Tree
Focus Tracking
Tests
Show Type Priorities Test for Events
Tests
Automation Element Tests
Priority 0 Tests
Priority 1 Tests
Priority 2 Tests
Control Tests
Pane
Properties
General Accessibility
Identification
Patterns
State
UI Automation Element Tree
UI Automation Element Tests
UI Automation Element Properties
Test Results
Back Overall AllResults Full Log XML Quick Find
UI Automation Test Results

Optimierung WPF für Testautomatisierung

UIA ist fester Bestandteil von WPF

Wichtige Eigenschaften

- AutomationProperties.AutomationId
- AutomationProperties. Name



WPF Codebeispiel

```
<Button
  AutomationProperties.AutomationId=„InsertCardButton"
  AutomationProperties.Name=„InsertCardButton"
  Grid.Row="1" Grid.Column="0" Grid.ColumnSpan="1"
  Margin="5"
  Command="{Binding Path=InsertCardCommand}">
    Insert Credit Card
</Button>
```

UIA

User Interface Automation

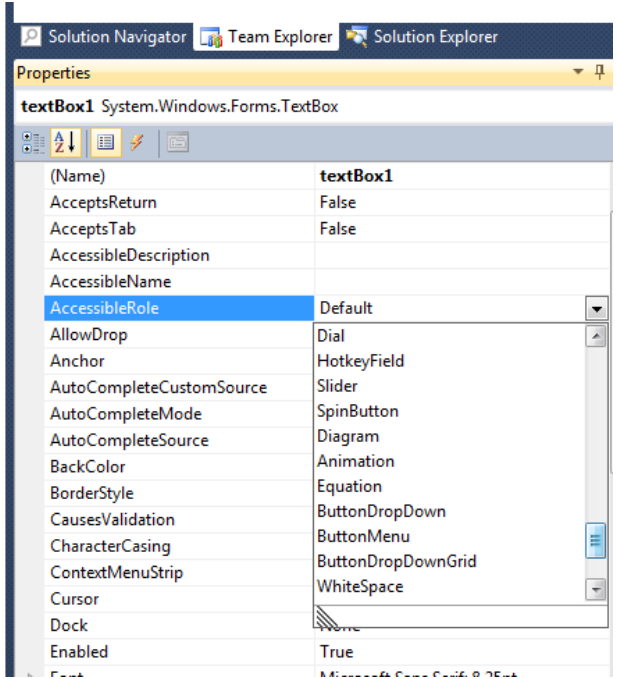
- COM-based technology
- Features COM-based and managed interfaces that ware introduced with Microsoft .NET Framework
- Used for automating WPF and XAML based Applications

Optimierung WinForms für Testautomatisierung

Ansteuerung von WinForms Controls erfolgt per MSAA

Wichtige Properties

- AccessibleName
- AccessibleDescription



MSAA

Microsoft Active Accessibility

- COM-based technology
- Integrated in into Windows Operating System since Windows 98
- Used for automating Win Forms applications

Optimierung Web Applications

Ansteuerung von Webanwendungen erfolgt per IE DOM bzw. Selenium Adapter (VS 2012 Update 2)

Wichtige Eigenschaften für HTML Tags

- **class**="klassenname“
- **id**="idname“

Web

Access Document Object Model (DOM) in Browser

- Supports IE out of the box
- Cross Browser Support for Chrome and Firefox
  - by using Selenium Web Driver

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

4tecture AIT

4tecture©

empower your software solutions

Marc Müller

Principal Consultant
für DevOps, ALM, TFS /VS, .NET

MVP Microsoft Most Valuable Professional

4tecture

E-Mail: marc.mueller@4tecture.ch
Webseite: http://www.4tecture.ch
Schulungen: http://4tecture.ch/trainings
Blog: http://4tecture.ch/blog
Twitter: @muellermarc

4tecture GmbH
Aathalstrasse 84
CH-8610 Uster

+41 44 508 37 00
info@4tecture.ch

4tecture©

empower your software solutions

Weiterführende Informationen

- MSPress Engineering For Accessibility
  http://download.microsoft.com/download/5/0/1/501FF941-E93D-423F-868B-C7BB2EC08C56/engineering_for_accessibility_eBook.pdf
- Beispiel Code
  https://github.com/marc-mueller/Demo-CodedUI