TECHTALK

AUTOMATISIERTE TESTS IN .NET

Werkzeuge zum automatisierten Testen in Visual Studio

Paul Rohorzka
@paulroho, paul.rohorzka@techtalk.at
Software-Developer

SNEK4 • donkarl.com/?snek I Nürnberg I April 2015 I Copyright © TechTalk 2015 • www.techtalk.at

AUTOMATISIERTES TESTEN VON SOFTWARE

UnitTesting
TDD
BDD

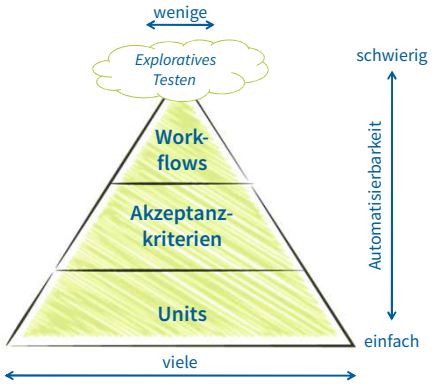
WARUM SOFTWARE TESTEN?

- Funktionalität
- Qualität
- Effizienz
- Fokus

ARTEN VON TESTS

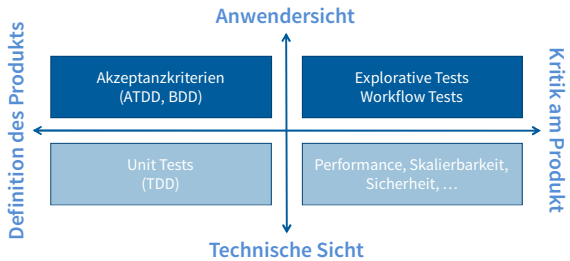
Funktionale Tests
Whitebox
Nach Methode
Blackbox
Systemtest
Exploration Test
Systemintegrationstest
Smoke Test
Developertests
Akzeptanztest
Nach Ebene
Unit Test
Usabilty Test
Regressionstest
Lasttest
Integrationstest
Performancetest
Securitytest
Usability-Test
Nicht-funktionale Tests

PYRAMIDE DER TESTAUTOMATISIERUNG



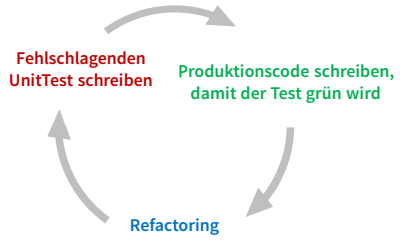
nach Mike Cohn

TEST QUADRANTEN

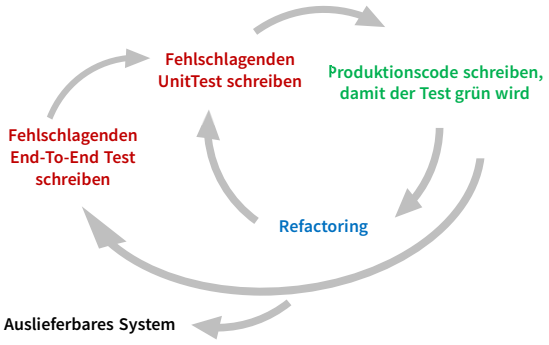


nach Brian Marick

TDD – TEST DRIVEN DEVELOPMENT



BDD – BEHAVIOUR DRIVEN DEVELOPMENT
ATDD – ACCEPTANCE DRIVEN DEVELOPMENT

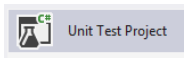


VISUAL STUDIO

Werkzeuge für automatisierte Tests
MSTest
Test-Explorer

MSTEST: PROJEKT – KLASSE – METHODE

Projektvorlage



Testklasse

```
[TestClass]
public class SomeTestClass
{
    [TestMethod]
    public void SomeTestMethod()
    {
        ...
    }
}
```

AUFBAU EINER TEST-METHODE

Getestete Methode | Getesteter Zustand | Erwartetes Verhalten

```
[TestMethod]
public void GetDiscount_WithSmallAmount_GivesNoDiscount()
{
    // Arrange
    var calculator = new Calculator();
    const decimal amount = 12.34M;

    // Act
    var actual = calculator.GetDiscount(amount);

    // Assert
    Assert.AreEqual(0.0M, actual);
}
```

MSTEST: WEITERE ATTRIBUTE

```
[Ignore]    // TODO: Warum geht das nur bei mir?

[AssemblyInitialize]
public static void EnsureDatabaseIsReady(TestContext context)
{
    //...
}

[TestInitialize]
public void CleanupDatabase()
{
    ...
}

[TestCleanup]
public void CheckLeftOverConnections()
{
    ...
}
```

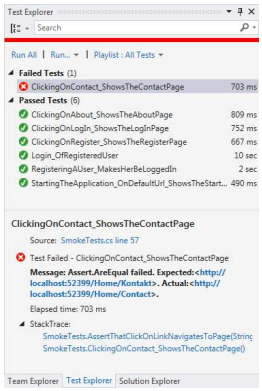
MSTEST: ASSERTIONS

Visual Studio

```
Assert.AreEqual("SNEK", actual);
Assert.IsTrue(isSaved);
Assert.IsNotNull(customer);
Assert.IsInstanceOfType(actual, typeof(Customer));
Assert.Inconclusive("WIP");


StringAssert.Contains(message, "abgelaufen");
StringAssert.EndsWith(message, "bis Ende des Jahres.");
StringAssert.Matches(line, new Regex(@"Name: (\w+)"));


CollectionAssert.AreEquivalent(new[] { "Nürnberg", "Wien" }, cities);
CollectionAssert.Contains(cities, "Wien");
CollectionAssert.AllItemsAreUnique(customerIds);
```

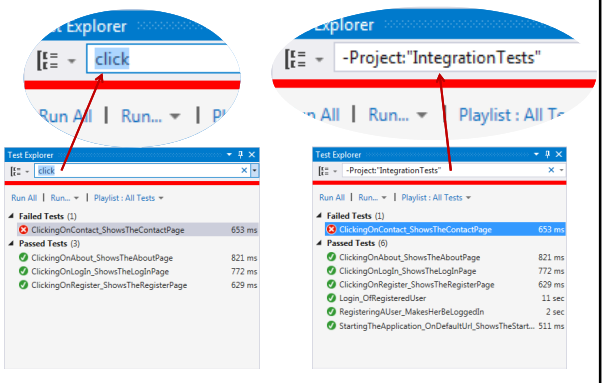
TEST EXPLORER

Visual Studio



TEST EXPLORER: FILTER

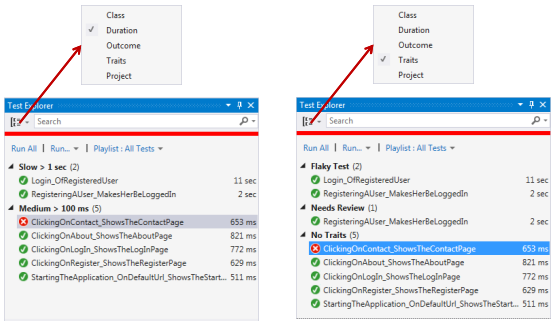
Visual Studio



TEST EXPLORER: GRUPPIERUNGEN

Visual Studio

```
[TestCategory("Needs Review")]
[TestCategory("Flaky Test")]
```



17

TASTATURKÜRZEL ZUR TESTAUSFÜHRUNG

Visual Studio

- Strg+R, **A** – Alle Tests ausführen
- Strg+R, **Strg+A** – Alle Tests mit Debugger ausführen

- Strg+R, **T** – Tests in aktuellem Bereich ausführen
  - Methode
  - Klasse
  - Namespace
- Strg+R, **Strg+T** – Tests in aktuellem Bereich mit Debugger ausführen

- Strg+R, **L** – Zuletzt gelaufene Tests nochmal ausführen

18

specflow

SPECFLOW

Ausführbare Spezifikationen
Lebende Dokumentation

Ausführbare Spezifikation

Open Source

Specification By Example

ATDD

BDD

http://www.specflow.org/

WAS MACHT SPECFLOW?

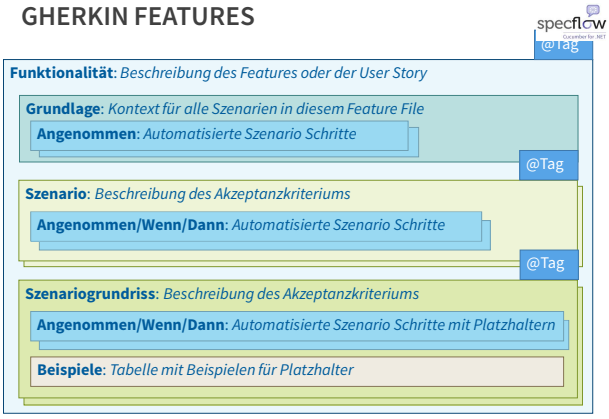
```
Szenario: Addition zweier Zahlen
Arrange | Angenommen ich habe 50 eingegeben
        | Und ich habe 70 eingegeben.
Act     | Wenn ich die Funktion Addieren aufrufe
Assert  | Dann sollte das Ergebnis 120 sein.
```

WIE MACHT DAS SPECFLOW?

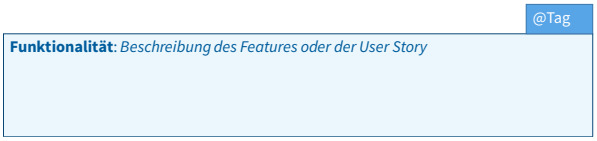
```
Szenario: Addition zweier Zahlen
    Angenommen ich habe 50 eingegeben
    Und ich habe 70 eingegeben.
    Wenn ich die Funktion Addieren aufrufe
    Dann sollte das Ergebnis 120 sein.
```

```
[Given(@"ich habe (\d+) eingegeben\.")]
public void AngenommenIchHabeEingegeben(int number)
{
    Calculator.Push(number);
}

[When(@"ich die Funktion Addieren aufrufe")]
public void WennIchDieFunktionAddierenAufrufe()
{
    Result = Calculator.Add();
}

[Then(@"sollte das Ergebnis (-?\d+) sein\.")]
public void DannSollteDasErgebnisSein(int expectedErgebnis)
{
    Assert.AreEqual(expectedErgebnis, Result);
}
```

GHERKIN FEATURES

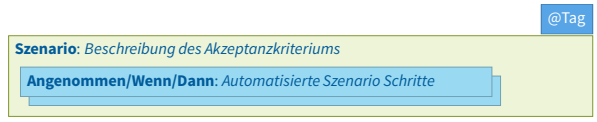


FUNKTIONALITÄT




```
#language:de-de
@done
Funktionalität: Warenkorb
  Als Webshop Besucher
  will ich Bücher in einem Warenkorb sammeln
  weil ich mehrere Bücher auf einmal bezahlen können will.
```

SZENARIO




```
@inprogress
Szenario: Das gleiche Buch kann mehrmals hinzugefügt werden
Angenommen mein Warenkorb enthält 1 Exemplar von "Harry Potter"

Wenn ich das Buch "Harry Potter" in den Warenkorb lege

Dann soll mein Warenkorb 2 Exemplare von "Harry Potter" enthalten
Und eine Warnung wird ausgegeben: "Gleiches Buch hinzugefügt"
```

SZENARIOGRUNDRISS

specflow

Szenariogrundriss: *Beschreibung des Akzeptanzkriteriums*

Angenommen/Wenn/Dann: *Automatisierte Szenario Schritte mit Platzhaltern*

Beispiele: *Tabelle mit Beispielen für Platzhalter*

```
Szenariogrundriss: Kandidatenprofil kann nicht gespeichert werden,
  wenn eines der Pflichtfelder (Email, Name, Vorname oder Geburtsdatum)
  nicht ausgefüllt wurde

  Wenn ich einen gültigen Kandidaten anlege, ohne das Feld <Pflichtfeld> anzugeben
  Dann sehe ich folgenden Validierungsfehler: 'Feld <Pflichtfeld> leer'

Beispiele:
    | Pflichtfeld |
    | Email       |
    | Name        |
    | Vorname     |
    | Geburtsdatum |
```

GRUNDLAGE

specflow

Grundlage: *Kontext für alle Szenarien in diesem Feature File*

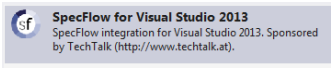
Angenommen: *Automatisierte Szenario Schritte*

```
Grundlage:
Angenommen die folgenden Bücher existieren im Produktkatalog
| Titel                 | Autor                    |
| Specification-By-Example | Gojko Adzic           |
| Agile Testing         | Lisa Crispin, Janet Gregory |
| Scrum Field Guide     | Mitch Lacey              |
```

VORBEREITUNG

specflow

Visual Studio: Extension



Projekt: Nuget-Package

```
Install-Package SpecFlow
```

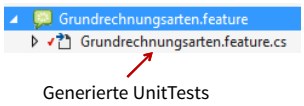
ev. Konfiguration

```
<specFlow>
    <unitTestProvider name="MsTest" />
    <language feature="de" />
</specFlow>
```

EINSATZ

Dateityp

Solution Explorer



Generierte UnitTests

Bindings in C#/VB-Datei

```
[Binding]
public class Bindings
{
    ...
}
```

SELENIUM WEBDRIVER

Automatisieren von UI-Tests

SELENIUM WEBDRIVER?

- API für den Browser
- Wird ein W3C Standard
- Unterstützung:
  - Gängige Desktop-Browser (default: Firefox)
  - Internet Explorer mittels IEDriverServer.exe (32 Bit!)
  - Mobilbrowser: iOS-driver, Selendroid
  - PhantomJS headless browser
  - Selenium Grid (unterschiedliche Browser als Service)

EINFACHE VERWENDUNG

```
Install-Package Selenium.WebDriver
```

```
[TestMethod]
public void BrowsingWithLocalDriver()
{
    using (var driver = new FirefoxDriver())
    {
        driver.Url = "http://donkarl.com?snek";

        Assert.AreEqual("donkarls Access-Seiten", driver.Title);
    }
}
```

WEBDRIVER MIT INTERNET EXPLORER

```
Install-Package Selenium.WebDriver.IEDriver
```

- Ev. Probleme mit IE11 nach Windows Update: http://stackoverflow.com/q/23770843/571213

ELEMENT AUFFINDEN

```
[TestMethod]
public void InspectingAnElement()
{
    using (var driver = new FirefoxDriver())
    {
        driver.Url = "http://donkarl.com?snek";

        var thirdLink = driver.FindElements(By.TagName("a"))[2];

        Assert.AreEqual("SNEK", thirdLink.Text);
    }
}
```

Einige Alternativen

```
var navRoot = driver.FindElement(By.ClassName("navigation"));
var snekLink = driver.FindElement(By.CssSelector("a[href='SNEK']"));
```

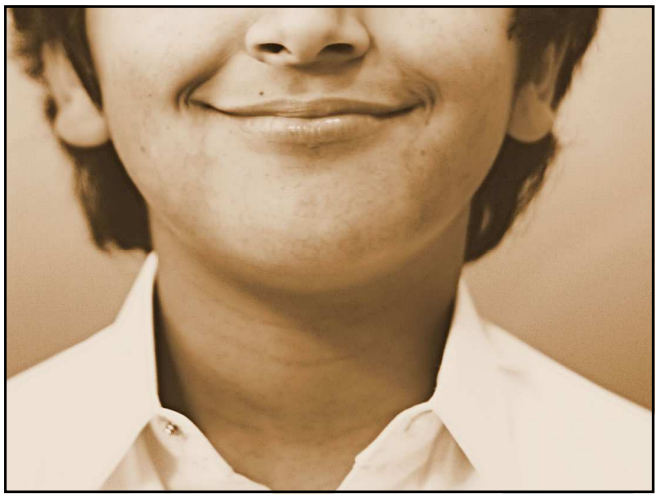
INTERAGIEREN

```
[TestMethod]
public void EnteringTextAndClicking()
{
    using (var driver = new FirefoxDriver())
    {
        driver.Url = "http://donkarl.com/FAQ/FAQSearch.asp";

        var searchTextBox = driver.FindElement(By.Name("txt_Search"));
        searchTextBox.SendKeys("Formular geöffnet");

        var searchButton = driver.FindElement(By.Name("btn_submit"));
        searchButton.Click();

        var link = driver.FindElement(By.PartialLinkText("Prüfen"));
        link.Click();

        StringAssert.Contains(driver.Url, "4.19");
    }
}
```

KEINE DETAILS: PAGEOBJECT PATTERN

- Details über die Seite kapseln

```
[TestMethod]
public void EnteringTextAndClickingWithPageObject()
{
    using (var driver = new FirefoxDriver())
    {
        driver.Url = "http://donkarl.com/FAQ/FAQSearch.asp";
        var page = new DonkarlSearchPageObject(driver);

        page.SearchText = "Formular geöffnet";
        page.PerformSearch();
        page.FollowResultContaining("Prüfen");

        StringAssert.Contains(driver.Url, "4.19");
    }
}
```



PRAKTISCH: SCREENSHOTS

```
[TestMethod]
public void TakeScreenshot()
{
    using (var driver = new FirefoxDriver())
    {
        driver.Url = "http://donkarl.com/?snek";

        var screenshot = driver.GetScreenshot();

        screenshot.SaveAsFile("DonkarlSnek.png", ImageFormat.Png);
    }
}
```

Tipp: Automatisieren bei fehlgeschlagenen Tests

```
[TestCleanup]
public void TakeScreenshotOnFailedTest()
{
    if (TestContext.CurrentTestOutcome == UnitTestOutcome.Failed)
    {
        var screenshot = WebDriver.GetScreenshot();

        ...
```

39

EINIGE HINWEISE

- Gut für End-To-End-Tests, Smoke-Tests
- Nicht die gesamte Funktionalität über UI testen
- Web-Tests müssen nicht FullStack-Tests sein
  - Z.B. Datenbank mocken
- Infrastruktur verstecken
  - z.B. Basisklasse für Webtests
  - PageObject Pattern
- Auf die Lesbarkeit achten!

40

TECHTALK

DANKE!

- paul.rohorzka@techtalk.at
- @paulroho
- paulroho.com/blog