

2.– 5. September 2013
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

Einführung in Test-Driven Development

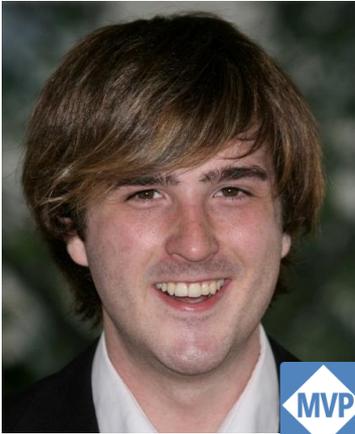
Möge der Test beginnen
Entwickler vs. Architekt

Gregor Biswanger & Robert Walter

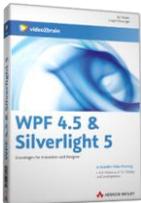
Freier Consultant & Trainer

complement AG

Über Gregor



Microsoft MVP für Client App Dev



- Freier Consultant und Trainer
- Schwerpunkte .NET-Architektur, Agile Prozesse und XAML
- Technologieberater für die Intel Developer Zone
- Speaker auf Konferenzen und User Groups
- Freier Autor bei heise.de, WindowsDeveloper-magazin, dotnetpro, visualstudio1, O'Reilly und Microsoft Press.
- Video-Trainer bei video2brain (Addison-Wesley) und für Webcasts bei Microsoft.

dotnet-blog.net

Über Robert



Microsoft Certified Professional Developer
Web Developer .NET 4

Microsoft Certified Solution Developer
Windows Store Apps using C#

- .NET-Softwareentwickler der complement AG in Nürnberg
- Liebhaber von Desktopanwendungen mit WPF
- Code Monkey im Coding Dojo
- Organisator und Sprecher von INdotNET, der [.NET Developers Group Ingolstadt](#)

indot.net

Das tägliche Leben...

„Das zu testen ist mir zu aufwändig. Der Kunde testet das schon...“

„Warum funktioniert das jetzt nicht mehr? Das hat doch schon funktioniert!“

„Wir haben doch keine Zeit für Unit-Tests!“

Ich muss jetzt noch dringende Fehler beheben...“

Aber wie testen?



... und

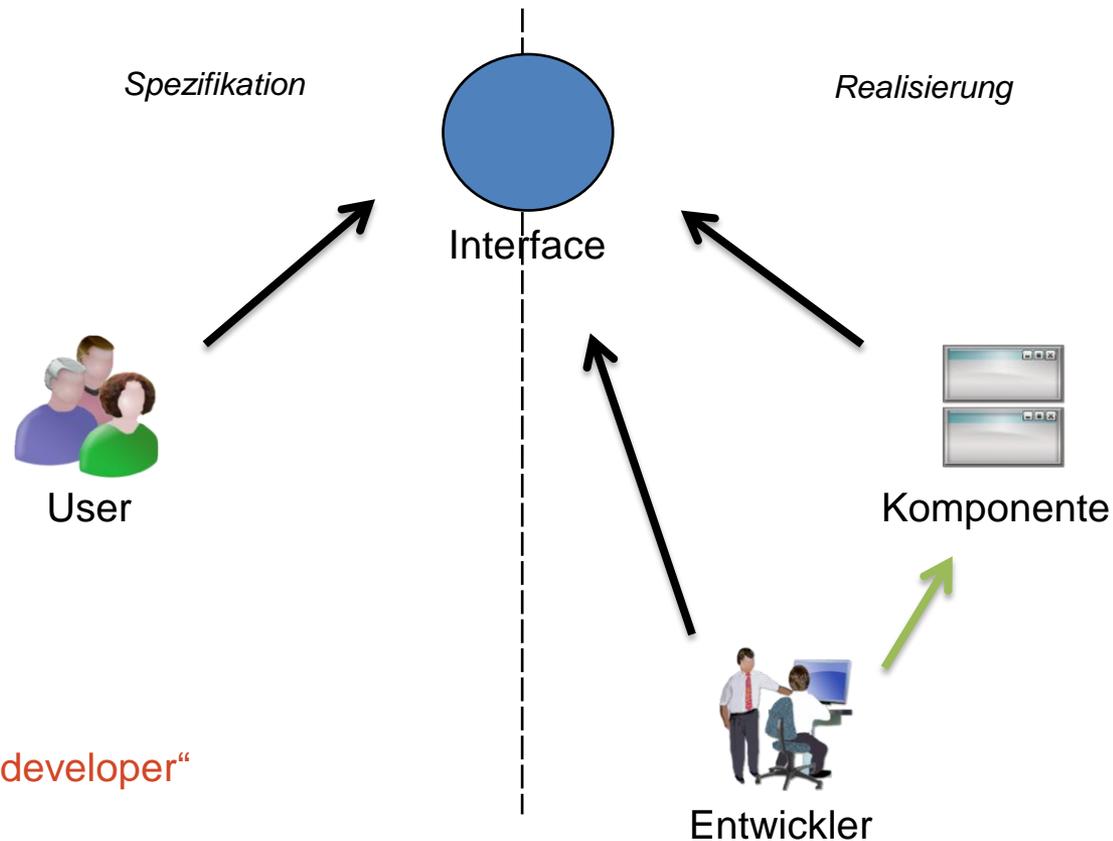
Test-Driven Development

soll da

wirklich

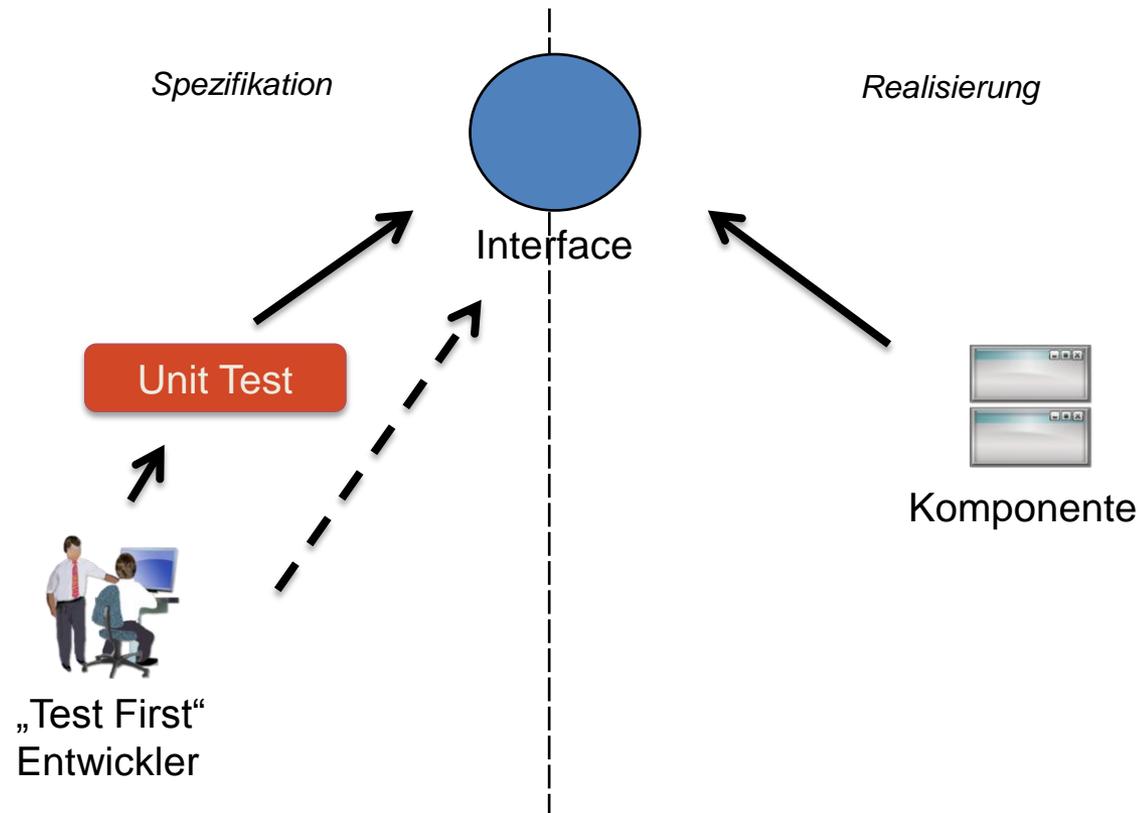
weiterhelfen?

Bisherige Testverfahren

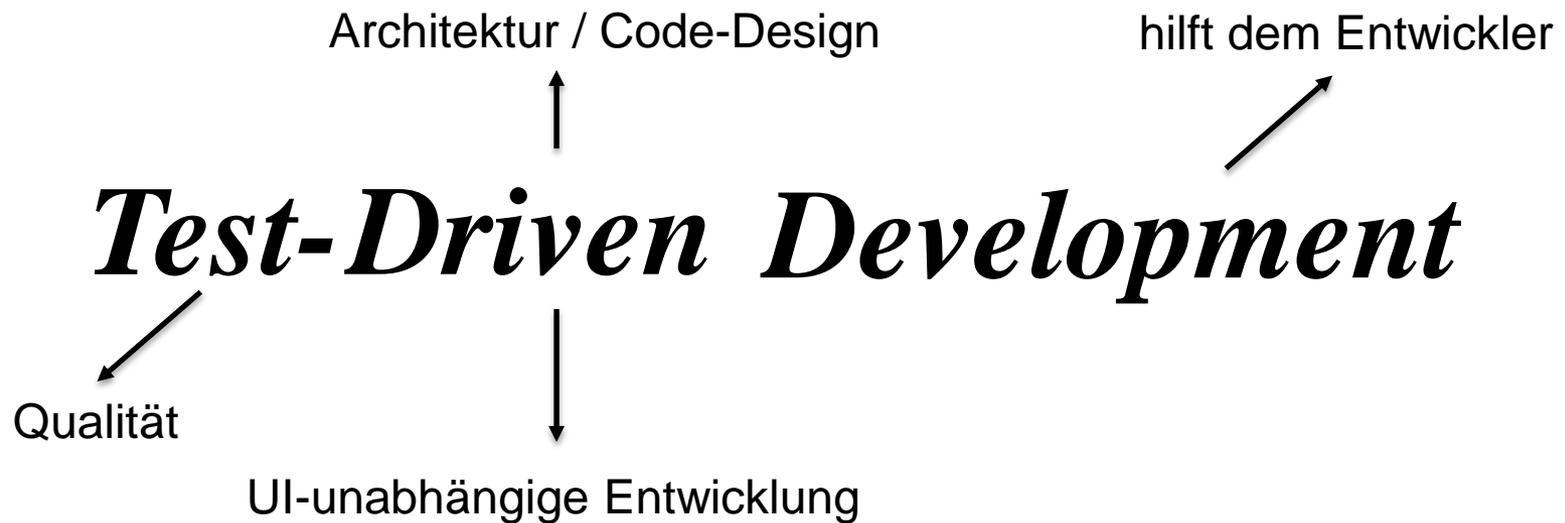


„Test last or never, developer“

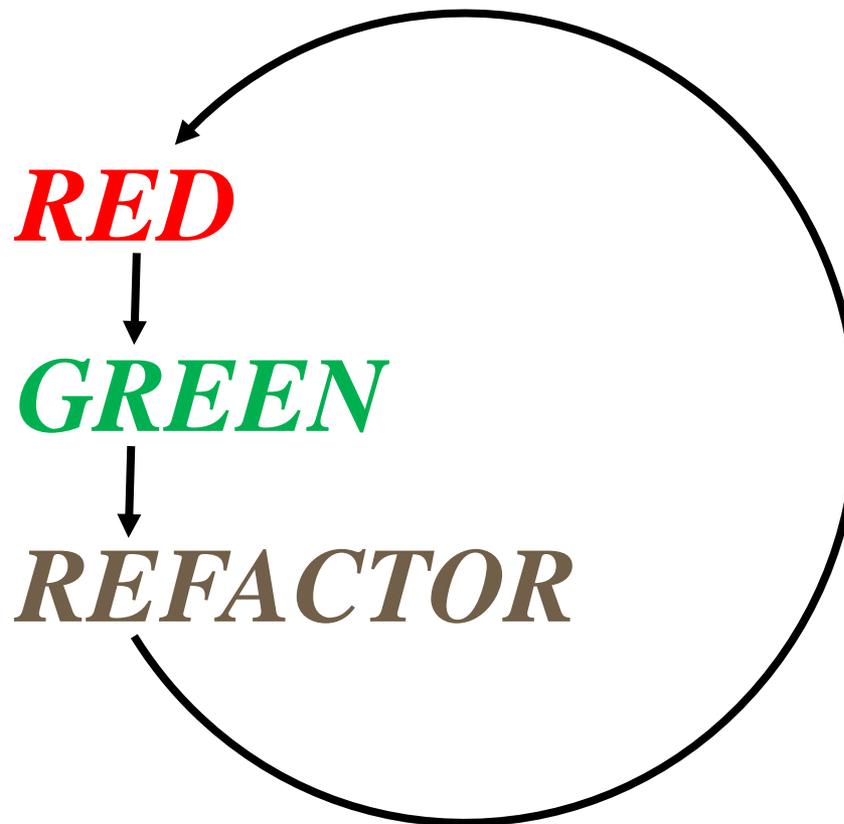
Testgetrieben (TDD)



Was ist TDD?



Wie geht TDD?



Wie geht das genau?

1. Schreibe den Testcode.
2. Mache ihn kompilierbar.
3. Starte den Test und lasse ihn fehlschlagen.
4. Setze nur die Anforderung um, damit der Test abgedeckt ist.
5. Starte den Test und prüfe, ob er erfolgreich ist.
6. Führe ein Refactoring zur besseren Übersichtlichkeit und Vermeidung von Coderedundanzen durch.
7. Starte von Vorne mit dem nächsten Test.

Ein Beispiel!

DEMO

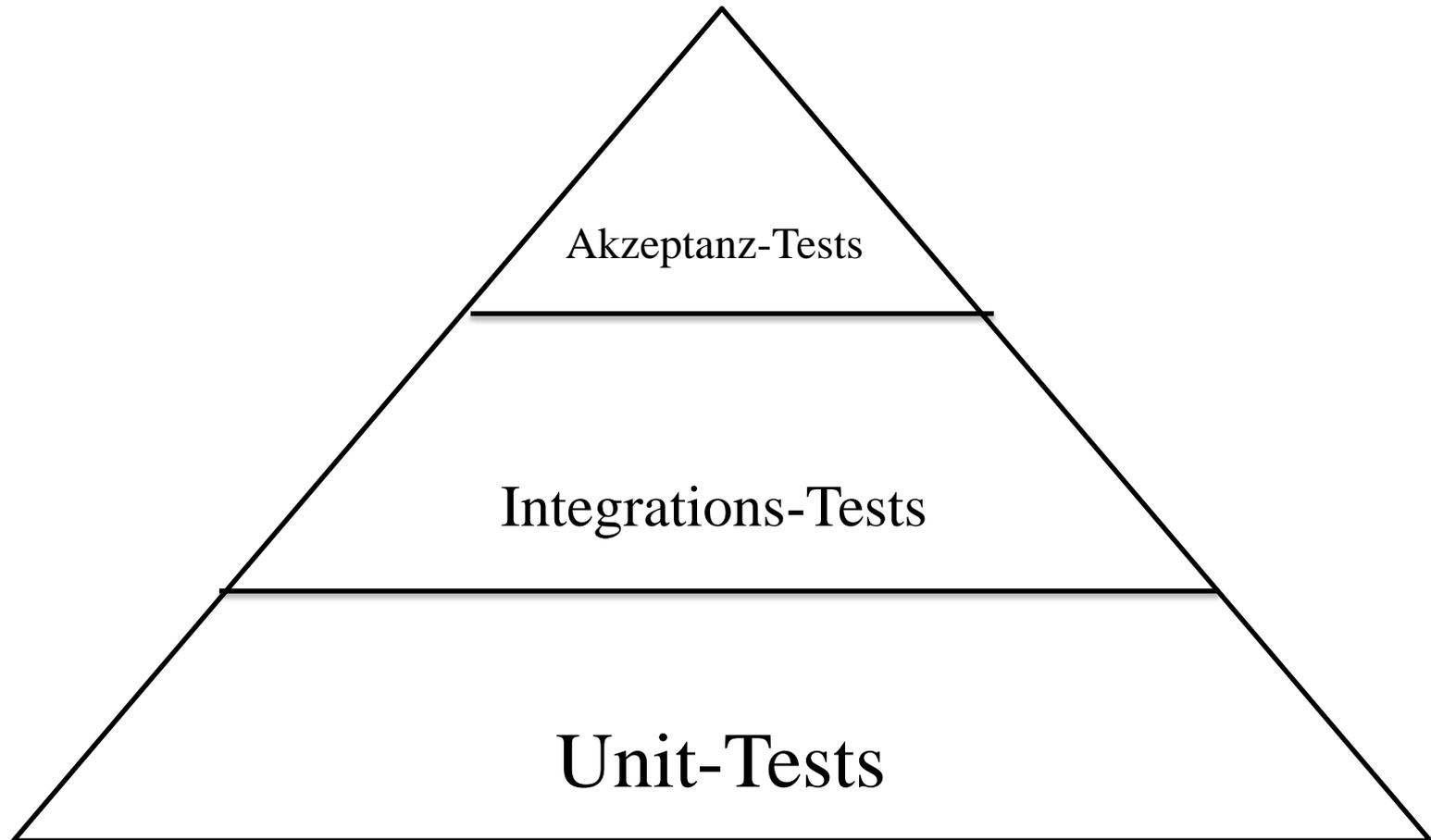
85049 Ingolstadt

90449 Nürnberg

???

München

Die Testpyramide



Fazit

- ✓ Automatisierte Tests decken viele Szenarien ab

„Das zu testen ist mir zu aufwändig. Der Kunde testet das schon...“

- ✓ Seiteneffekte auf andere Funktionalitäten werden durch Tests verringert

„Warum funktioniert das jetzt nicht mehr? Das hat doch schon funktioniert!“

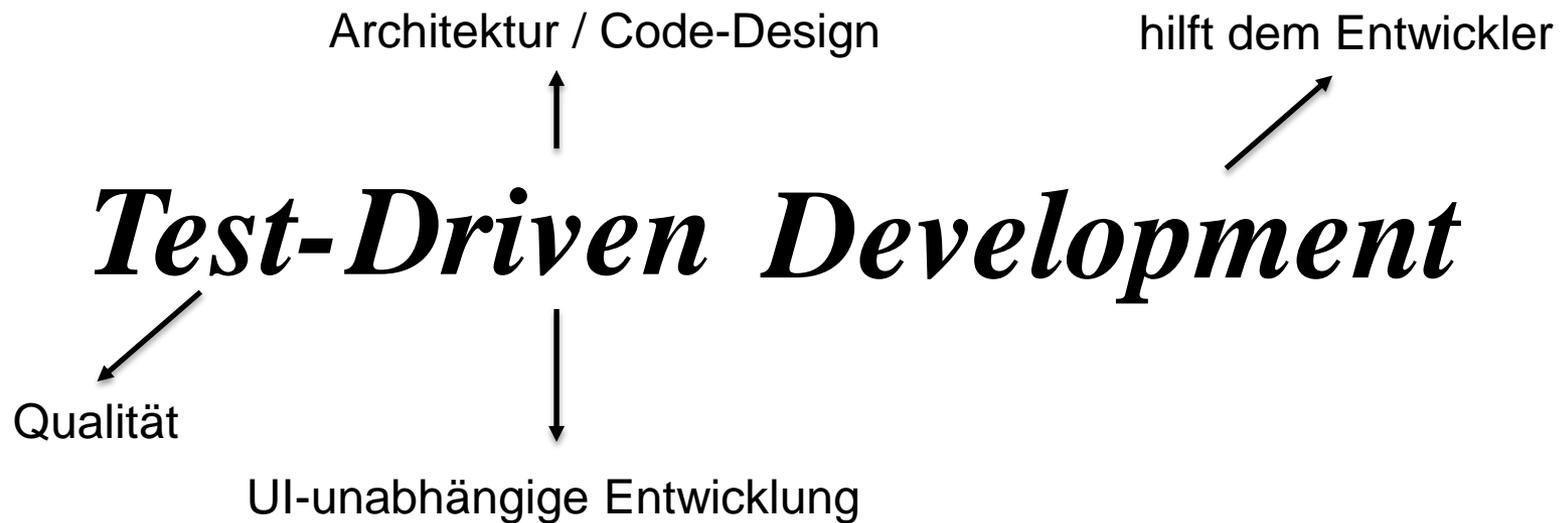
- ✓ Am Anfang: Mehr Zeit für Tests benötigt
- ✓ Später: Zeit wird gespart, da keine GUI-Ausführung notwendig ist

„Wir haben doch keine Zeit für Unit Tests!“

Ich muss jetzt noch dringende Fehler beheben...“

Fazit

- ✓ Bessere Schnittstellen (für Aufrufer)
- ✓ „Grünes Erfolgserlebnis“



- ✓ Automatisierte Tests mit Testabdeckung
- ✓ Keine wiederholten wilden Klick-Tests mehr

dotnet-blog.net

DotNet-Blog.NET

Every day is an experience!

Home Archiv Kontakt Abonnieren Log in

Über den Autor
<< Rx – Lösung zur ersten Webcast-Übung | Reactive Extensions (Rx)-Asynchrone Prozesse >>



MVP

Microsoft
Most Valuable Professional

mein XING[®] profil

Gregor Biswanger (Microsoft MVP für Client App Dev) ist freier Consultant, Trainer, Autor und Speaker.

Seine Schwerpunkte liegen im Bereich der .NET-Architektur, XAML und agilen Prozessen. Er veröffentlichte vor kurzem seine DVD's mit Video-Trainings zum Thema „Meine erste Windows 8 App“ und „WPF 4.5 und Silverlight 5“ bei Addison-Wesley von video2brain.

Biswanger ist auch freier Autor, Speaker und Microsoft CLIPier der #NoToNET (Ingotlädter .NET Developers Group).

Video über mich:
http://www.youtube.com/watch?v=mx_6SILxjk

SharePoint 2007 - Video-Training

8 Stunden Video-Training

.NET-Architektur mit „GoFish“ – Vision, Diagnose und Analyse

By Gregor Biswanger 5. August 2011 08:44



In meiner Familie ist derzeit ein hohes Interesse am Erlernen von C# aufgetaucht. Noch interessanter sind die Übungsaufgaben der heutigen Lehrliteratur. Die Ziele sind jeweils kleine Spiele zu programmieren, die das Lernen durch einen hohen Spaßfaktor versüßt. Dabei fand ich folgendes Spiel sehr amusant: „GoFish“.

Bei Lehrbüchern wird oft nur vorgegeben, wie es zum Lösungsweg kommen soll. Dabei stehen oft unfertige Klassen und Funktionen als Starthilfe bereit. Ich nahm mir das Beispiel „GoFish“ und entwarf daraus ein komplett neues Vorzeigeprojekt. Das Vorzeigeprojekt sollte vom Umfang her nicht gleich explodieren und gerade für den Einstieg in die Architektur, Vorgehensweisen und Technologien bieten.

Das Thema Softwarearchitektur ist sehr umfangreich, so dass diese Architektur-Serie nur recht oberflächlich auf die meinerseits vorgegangene Vorgehensweise zu diesem Beispiel eingehen wird. Dabei werden wir gemeinsam die sechs Phasen der Entwicklung durchgehen:

Diagnostic

Analysis

Design

Development

Deployment

Operation

Die Serie selbst wird folgende Technologien und Vorgehensweisen aufzeigen:

Visual Studio 2010 Ultimate - Architecture Tools (Use Cases, Activity Diagram, UML, Layer Diagram, Component Diagram), Sketching, Prototyping, Domain-Driven Design (DDD), Model-Driven Architecture (MDA), Test-Driven Development, Workflow-Foundation 4 (WF4), Aspektorientierte Programmierung (AOP), Reactive Extensions (Rx), Model-View-ViewModel (MVVM), WPF, Silverlight, ASP.NET MVC 3 und vieles mehr.

Ich wünsche Euch viel Spaß und bin für jede Diskussion offen!

Die Vision

Von der Idee zur Vision. So beginnt es mit jedem Softwareprojekt. In der Regel hat der Kunde eine Vision, nämlich eine Softwarelösung für sein Anliegen zu erhalten. Bei diesem Beispiel kommt die Vision eher von einem Lehrbuch. Somit also auch die Spezifikation. Dieses gibt vor ein Kartenspiel namens „GoFish“ zu programmieren. Dabei gibt es folgende Spielregeln (*funktionale Anforderungen*¹⁾:

- Das Spiel beginnt mit einem Satz aus 52 Karten. Jeder Spieler erhält fünf Karten. Die Spieler fragen einander abwechselnd nach einem Wert (Zum Beispiel: „Hat einer ein Ass?“). Hat ein anderer Spieler Karten mit diesem Wert, muss er sie abtreten. Hat kein anderer Spieler eine Karte mit diesem Wert, muss der Spieler „Fischen gehen“, indem er eine Karte aus dem Stapel nimmt.
- Das Ziel des Spieles ist es, Quartette zu sammeln. Ein Quartett ist ein vollständiger Satz aller vier Karten mit dem gleichen Wert. Der Spieler, der am Ende des Spieles die meisten Quartette hat, ist der Gewinner. Sobald ein Spieler ein Quartett zusammen hat, legt er es offen auf den Tisch, damit jeder sehen kann, wer welche Quartette hat.
- Bei dieser Computerversion von GoFish Gibt es zwei Computerspieler und einen normalen Spieler. Jede Runde beginnt damit, dass der Spieler unter seinen angezeigten Handkarten eine aussucht. Das macht er, indem er eine der Karten auswählt und anzeigt, dass er danach fragen wird. Dann fragen die beiden Computerspieler nach ihren Karten. Das Ergebnis der einzelnen Runden wird angezeigt. Das wird wiederholt, bis es einen Sieger gibt.
- Das Spiel kümmert sich automatisch um den Austausch der Karten und das Ablegen der Quartette. Gibt es einen Sieger, ist das Spiel vorüber. Dann zeigt das Spiel den Namen des Siegers an (oder der Sieger, falls es unentschieden ausgeht). Es kann nichts mehr



indot.net



Ingolstädter .NET Developers Group



[Home](#)
[Events](#)
[Empfehlungen](#)
[XING-Gruppe](#)
[Kontakt](#)
[Impressum](#)

[DotNet-Blog.NET](#)



Wir stellen uns vor.....

Wer oder was ist die Ingolstädter .NET Developers Group?

Die Ingolstädter .NET Developers Group ist eine unabhängige Gemeinschaft für Interessierte der Microsoft .NET Technologie aus dem Großraum Ingolstadt. Sie besteht aus professionellen Entwicklern, Hobbyentwicklern, Akademikern und Studenten, die Ihre Erfahrung mit anderen teilen möchten.

Ziel dieser Vereinigung ist es, Wissen und Erfahrung durch persönliche Treffen zu transferieren. Hierzu treffen wir uns regelmäßig zu Vorträgen, zu Workshops oder einfach nur zum Informationsaustausch. Diese Treffen sind kostenlos und an keine Mitgliedschaft gebunden.

Unsere Türen stehen allen .NET Interessierten offen. Machen Sie beim aktiven Erfahrungsaustausch mit!

Nächstes Treffen:
 Unser Juni-Treffen findet am Montag, den **17.06.2013 um 19:00 Uhr** statt. Wir treffen uns wie üblich im **Hotel-Gasthof zum Anker** in der Tränktorstraße 1, 85049 Ingolstadt. Die Adresse findet ihr auch unter **Bing Maps**.

Was ist neu in .NET 4.5 und C# 5?

Features abseits der App-Entwicklung - Was haben .NET 4.5 und Visual Studio 2012 am Unterbau für Oberflächen und Kommunikation geändert?
 (von Jan Schweda, complement AG)

Version 4.5 ist das zweite große Update für das .NET Framework in nicht einmal 1,5 Jahren. Notwendig wurde das Update aufgrund der vielen neuen Features für Windows 8. Aber auch, wenn die eigene Anwendung nicht über den Store vertrieben werden soll, gibt es Features, die das Arbeiten mit der aktuellen Version des Frameworks effizienter, einfacher und schneller machen. Ein wesentlicher Teil der Session werden Parallelisierung und asynchrones Ausführen von Code sein. In WCF werden wir uns die wichtigsten neuen Features wie UDP-Binding, WebSockets und IntelliSense in der Konfiguration anschauen. In WPF sind einige der neuen Features, die in dieser Session beleuchtet werden, das Sortieren von Daten zur Laufzeit, das DataBinding an statische Eigenschaften sowie das Ribbon-Control.

Über den Sprecher

Jan Schweda ist Software Engineer bei der complement AG in Nürnberg. Sein Fokus liegt vor allem auf verteilten Systemen und serviceorientierter Architektur. Jan entwickelt seit mehreren Jahren Consumer und Enterprise-Anwendungen im .NET Umfeld und hat hierbei Erfahrungen in den unterschiedlichen Bereichen des .NET Technologie Stacks gesammelt. Sein Wissen gibt er regelmäßig und gerne als (Microsoft Certified) Trainer, Fachautor und Sprecher weiter.

.NET 4.5 Feature Catalog

 WinRT <small>.NET for Windows Store Apps</small>	 MVC <small>ASP.NET MVC 4</small>	 Await <small>Asynchrone Programmierung</small>	 UTF-16 <small>Unicode support for Unicode</small>
 Security <small>AD.NET Security</small>	 ASP.NET <small>ASP.NET User Voice</small>	 Server <small>Background Server Settings</small>	 Entity Framework <small>Entity Framework 6</small>

MSDN Online Deutschland

MSDN Online Deutschland: News & Infos auf einen Blick; das Neueste rund um und von MSDN für Entwickler (RSS-Feed)

Office-Apps: Web Services über die ASP.NET Web API bereitstellen
 23.06.2013
 Diese Schritt-für-Schritt-Anleitung zeigt Entwicklern, wie sich über die ASP.NET Web API Web Services erstellen lassen, um anschließend die Erstellung von Office-Apps darauf...

PHP 5.5 als Release Candidate 3 verfügbar

2.– 5. September 2013
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

Vielen Dank!

Gregor Biswanger & Robert Walter

Freier Consultant & Trainer

gregor.biswanger@web-enliven.de

complement AG

robert.walter@complement.de