

MSDN Webcast:

Sicherstellen einer hohen Codequalität mit dem Team Foundation Server 2008 Visual Studio Team System (Teil 9 von 10)

Veröffentlicht: Mai 2008

Presenter:

Neno Loje, MVP für Team System

www.teamsystempro.de



Wer sind sie?

- Sie...
 - ... sind Entwickler und Tester und sollen Qualität liefern?
 - ... möchten einen Einblick in die aktuelle Qualität bekommen und den Trend analysieren können?
 - ... suchen nach einer Lösung Qualitätsstandards einzuführen und durchzusetzen?
- Was machen wir nicht?
 - Qualitätsmanagement

Agenda

- Motivation
- Die Werkzeuge
- Qualitätsstandards



Herausforderungen

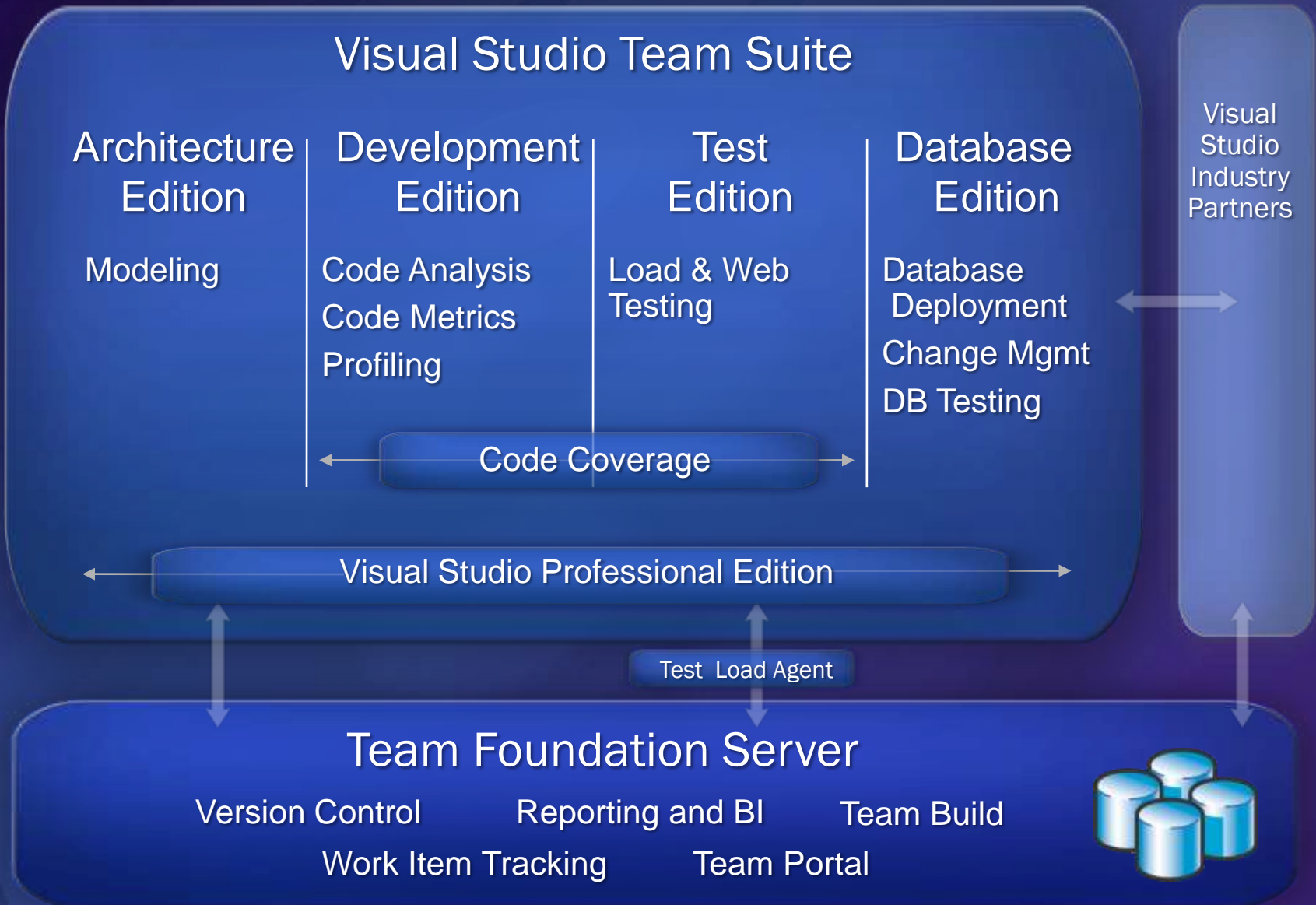
- Gestiegene Qualitätserwartung
- Gestiegene Komplexität
 - Entwickler müssen in vielen Disziplinen "Weltmeister" sein
- Durchsetzung einheitlicher Qualitätsstandard häufig schwierig

Die VSTS-Werkzeuge

- Entwickler
 - Statische Codeanalyse
 - Codemetriken
 - Unit Test & Codeabdeckungsanalyse
 - Profiler
 - Kodierter Webtest
 - T-SQL-Unit Test
- Tester
 - Web-Oberflächentest
 - Lasttest

Visual Studio Team System

MSF Process and Guidance



Codeanalyse

Statische Codeanalyse

(auch bekannt als "FxCop")

- Frühzeitige Erkennung bekannter Probleme
 - Läuft bei jedem Kompilieren und als Teil des Serverseitigen Buildprozesses
- Regelbasierte Prüfung des Kompilats
 - Von Namenskonventionen bis zu bekannten Security- und Performanceproblemen
- Erweiterbar durch eigene Regeln
- Erfordert: VSTS Development Edition

Demo

Prüfen des Quellcodes auf
bekannte Probleme



Mehr Informationen?

- Visual Studio Code Analysis Team Blog
<http://blogs.msdn.com/fxcop/>
- .NET Framework Developer's Guide: Design Guidelines for Developing Class Libraries
<http://msdn.microsoft.com/en-us/library/ms229042.aspx>

Code-Metriken

Code-Metriken

- Aufgabenstellung:
 - Welche Stellen in meinem Quellcode bedürfen einer Verschönerung? (Refactoring)
- Metriken
 - Cyclomatic Complexity
 - Code "Verwurschtlung"
 - Vererbungstiefe
 - Anzahl Codezeilen (Lines of Code; LOC)
- Erfordert: VSTS Development Edition

Code Metrics

Code Metrics Results

Filter: None Min: Max: [Icons]

Hierarchy	Maintainability Index	Cyclomatic Complexity	Depth of Inheritance	Class Coupling	Lines of Code
BusinessLayer (Release)	38	545	1	9	565
BusinessLayer	38	545	1	9	565
Address	37	265	1	7	275
Address(int, string, string)	76	1		0	4
Id.get() : int	98	1		0	1
LoadAddress(int) : Address	18	102		7	108
Save() : void	7	159		3	160
StreetAddress1.get() : string	98	1		0	1
StreetAddress2.get() : string	98	1		0	1
Customer	38	280	1	7	290
Address.get() : Address	98	1		1	1
Customer(int, string, string)	76	1		0	4
FirstName.get() : string	98	1		0	1
Id.get() : int	98	1		0	1
LastName.get() : string	98	1		0	1
LoadCustomer(int) : Customer	8	146		6	152
Save() : void	13	129		2	130
DataAccessLayer (Release)	95	6	1	2	6
MainApplication (Release)	84	10	7	5	16

Cyclomatic Complexity

Complexity

1

```
bool ParseCommandLine(string[] arguments)
```

```
{
```

2

```
    if (arguments.Length == 0)
```

```
    {
```

```
        ShowHelp();
```

```
        return false;
```

```
    }
```

3

```
    for (int i = 0; i < arguments.Length; i++)
```

```
    {
```

4

```
        if (arguments[i] == "/?")
```

```
        {
```

```
            ShowHelp();
```

```
            return false;
```

```
        }
```

5

```
        if (arguments[i] == "/input")
```

```
        {
```

6,7

```
            if (arguments.Length > 1 && File.Exists(arguments[i + 1]))
```

```
            {
```

```
                InputFileName = arguments[i++];
```

```
            }
```

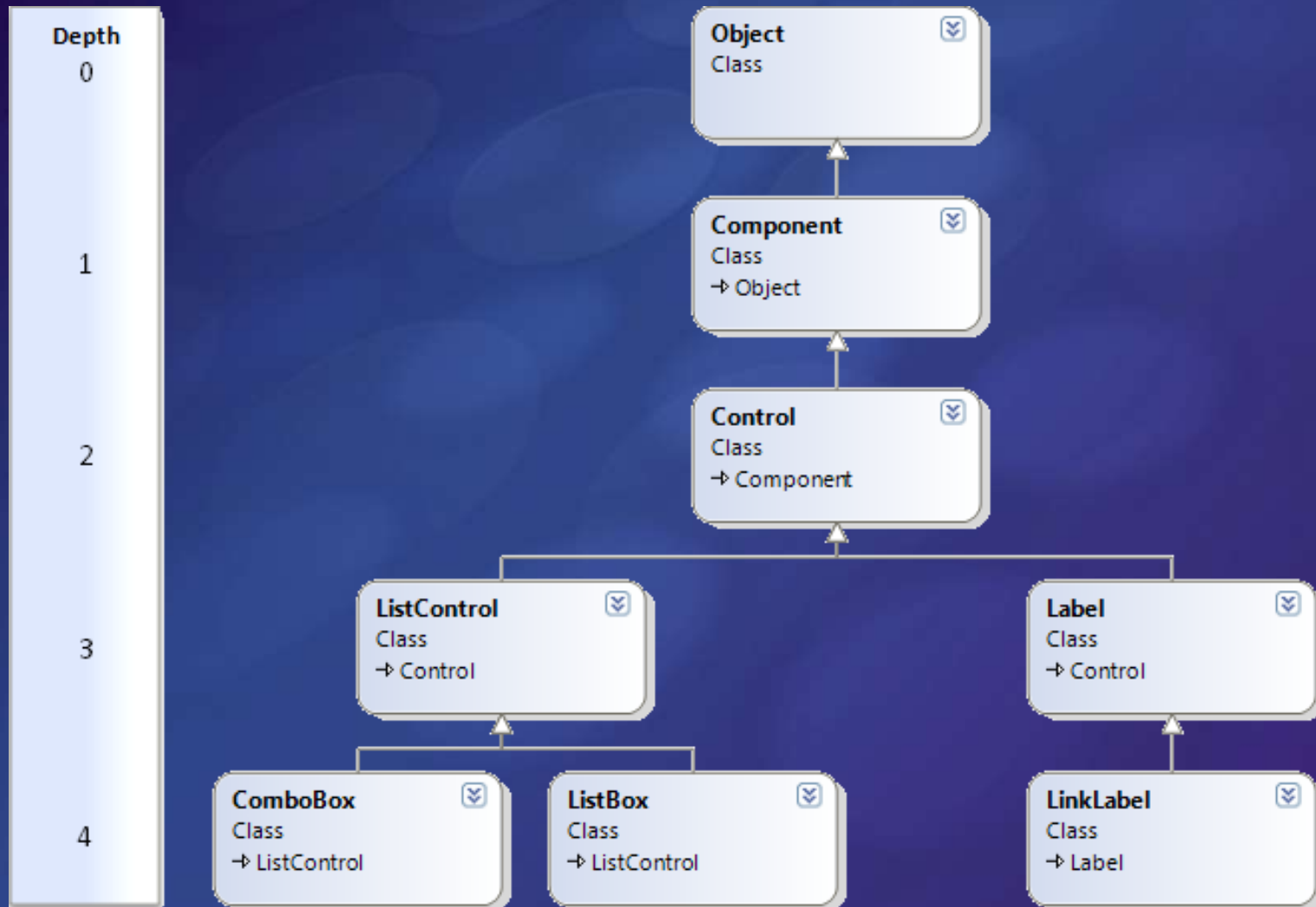
```
        }
```

```
    }
```

```
    return true;
```

```
}
```




Depth of Inheritance



Class Coupling



Berechnung

- Maintainability Index
= $\text{MAX}(0, (171 - 5.2 * \ln(\text{Halstead Volume}) - 0.23 * (\text{Cyclomatic Complexity}) - 16.2 * \ln(\text{Lines of Code})) * 100 / 171)$
- Ergebnisse
 - 0 – 9 : rote Wertung 
 - 10 – 19 : gelbe Wertung 
 - 20 – 100: grüne Wertung 
- Mehr Informationen
 - <http://blogs.msdn.com/fxcop/archive/2007/11/20/maintainability-index-range-and-meaning.aspx>
 - <http://www.sei.cmu.edu/str/descriptions/mitmpm.html>

Demo

Analyse der Codewartbarkeit



Unit Test & Codeabdeckung

Unit Test & Codeabdeckung

- Code schreiben um Produktivcode zu testen
- Zwei Vorgehensweisen
 - Testen existierenden Codes
 - Testgetriebene Entwicklung (Test Driven Dev.)
- Datengetriebene Tests
 - Sparen Arbeit
 - Testwerte lassen sich von Nicht-Entwicklern ändern
 - Finden Fehler
- Erfordert: VS Pro (für Unit Test) und VSTS Development Edition (für Codeabdeckung)

Demo

Unit Testing & Code Coverage



Profiler

Profiler

- Dient der Analyse von:
 - Ausführungszeit: **Wo ist der Flaschenhals?**
 - Speicher: **Wo wird er verbraucht?**
- Trendanalyse durch Vergleich möglich
- Zwei Betriebsmodi: **Instrumentation & Sampling** (für .NET und nativen Code)
- Lässt sich auch auf Tests anwenden
- Kann auch als eigenständiges Programm auf einem Produktivserver mitlaufen
- Erfordert: VSTS Development Edition

Demo

Einsatz vom Profiler



Informationen zum Profiler

- 4-Teilige Serie zur Analyse mit dem Profiler
 - <http://blogs.msdn.com/ianhu/archive/2005/03/08/389849.aspx>
 - <http://blogs.msdn.com/ianhu/archive/2005/03/15/395853.aspx>
 - <http://blogs.msdn.com/ianhu/archive/2005/03/31/404173.aspx>
 - <http://blogs.msdn.com/ianhu/archive/2005/04/08/406548.aspx>
- Auswertung spezifischer Berichte
 - <http://blogs.msdn.com/ianhu/archive/2005/08/25/456531.aspx>
 - <http://blogs.msdn.com/ianhu/archive/2005/12/19/505702.aspx>
 - <http://blogs.msdn.com/ianhu/archive/2005/09/15/467655.aspx>
- Videos auf Channel 9
 - <http://channel9.msdn.com/ShowPost.aspx?PostID=310324>
 - <http://channel9.msdn.com/ShowPost.aspx?PostID=46401>
- Comparing reports -
<http://blogs.msdn.com/ianhu/archive/2007/04/03/comparing-performance-reports-with-the-visual-studio-team-system-profiler.aspx>
- Hot path -
<http://blogs.msdn.com/ianhu/archive/2007/09/14/pinpoint-a-performance-issue-using-hotpath-in-visual-studio-2008.aspx>
- Noise reduction -
<http://blogs.msdn.com/colinth/archive/2008/01/31/noise-reduction-in-vs2008.aspx>

Web-Oberflächentest

Web-Oberflächentest

- Zeichnet HTTP-Verkehr auf und kann diesen wieder abspielen
 - Nutzt keinen Browser, d.h. kein Testen von clientseitigem JavaScript
 - Unterstützt AJAX
- Einfach an geänderte UI anpassbar
- Parameter aus Datenquelle einlesbar
- Kodierter Webtest für mehr Flexibilität
- Erfordert: VSTS Test Edition

Demo

Einsatz von Web- Oberflächentests



Lasttests

(auch bekannt als Stresstest)

- Motivation: Wie stelle ich sicher, meine Anwendung läuft auch unter der vereinbarten Benutzerlast?
- Gründe u.a.:
 - Wie viele Benutzer unterstützt meine Anwendung?
 - Unterstützt meine Anwendung X Benutzer?
 - Was müsste getan werden um noch weitere Y Benutzer mit der Anwendung arbeiten zu lassen?
 - Ausdauerstest
- Erfordert: VSTS Test Edition

T-SQL Unit Tests

T-SQL Unit Tests

- Testen von Logik in der Datenbank
 - Stored Procedures
 - User-Defined Functions
 - Triggers
 - etc.
- Erfordert: VSTS Database Edition

Demo

Einsatz von T-SQL Unit Tests



Windows UI-Tests?

Was ist mit Windows UI-Tests?

- Noch nicht Teil von Team System
 - Funktionalität kommt in Version 3 ("Rosario")
- Es gibt im .NET Framework UI-Testklassen
 - MSDN Webcast: UI Automation Framework - UI Tests automatisieren
<http://www.microsoft.com/germany/msdn/webcasts/library.aspx?id=1032376565>

Qualitätsstandards

Qualitätsstandards

- Ziel: Sicherstellen einer einheitlichen Qualität und Konventionen
- Realisierung mit dem TFS
 - Nachverfolgung
 - Integration in Buildprozess
 - Trendanalyse
 - Eincheckrichtlinien / Quality Gates

Nachverfolgung

Nachverfolgung

- Dokumentation & Abbildung des Workflows
 - Zustandsmodell erweiterbar, z.B. um Code Review
- Lückenlose Nachverfolgbarkeit
 - Wer hat wann welchen Zustand geändert?
- Auswertung
 - Bugraten, etc.

Demo

Nachverfolgung von Work Items



Buildprozess

Integration in den Buildprozess

- Ziele vom serverseitigen Buildprozess:
 - Frühe Erkennung von Integrationsproblemen
 - Bewertung der Qualität (mehr als Kompilierung)
- Auf Wunsch per Mausklick integriert
 - Statische Codeanalyse ("FxCop")
 - Unit Tests & Codeabdeckung (Code Coverage)
 - Änderungsrate am Code ("Code Churn")
- Integration mit Work Items
 - Was ist neu? Welche Bugs wurden behoben?

Demo

Qualitätsmessung als Teil
vom serverseitigen Buildprozess



Trendanalyse

Trendanalyse

- Häufige Fragen:
 - Wie entwickelt sich die Qualität?
 - Muss ich einschreiten?
 - Kann ich ruhigen Gewissens ausliefern?
- Buildfrequenz und Berichte
 - Anzahl erfolgreicher / fehlgeschlagener Tests
 - Codeabdeckung & Codeänderungsrate
 - Fehler & Warnungen über Konventionen
 - Bugrate

Demo

Auswertung von Qualitätstrends



Eincheckrichtlinien

Eincheckrichtlinien

- Herausforderung:
 - Wie definiere ich verbindliche Standards?
 - Wie kann ich sicherstellen, dass nur qualitativ hochwertiger Code eingecheckt wird?
- Beschränken auf Ordner oder Branches
- Shelving als Alternative für unfertigen Code
 - Einchecken in einen privaten Bereich

Demo

Einsatz von Eincheckrichtlinien
zur Qualitätssicherung



Ausblick

- Team Foundation Server 3.0 ("Rosario")
 - Testmanagement
 - Eine Reihe neuer Qualitäts- und Testwerkzeuge

Zusammenfassung

- Spezielle Qualitätswerkzeuge für Entwickler und Tester
- Sicherstellen projektweiter oder unternehmensweiter Qualitätsstandards auf dem Server
 - Buildprozesse, Eincheckrichtlinien, etc.

Nächste Schritte

1. Bestandsaufnahme: Wo stehen wir?
 - Von welchen Problemen wissen wir?
2. Qualität sicherstellen
 - Einsatz von Werkzeugen wie der Statischen Codeanalyse zur Prüfung der Richtlinien
3. Gezielter Einsatz spezieller Werkzeuge
 - z.B. Lasttest um zu Prüfen ob die versprochene Benutzeranzahl wirklich unterstützt wird

Weitere Informationen

- Eigene Codeanalyse-Regeln (engl.)
<http://www.binarycoder.net/fxcop/>
- Code Analysis and Code Metrics Team Blog (engl.)
<http://blogs.msdn.com/fxcop/>

Webcast Serie zu VSTS 2008

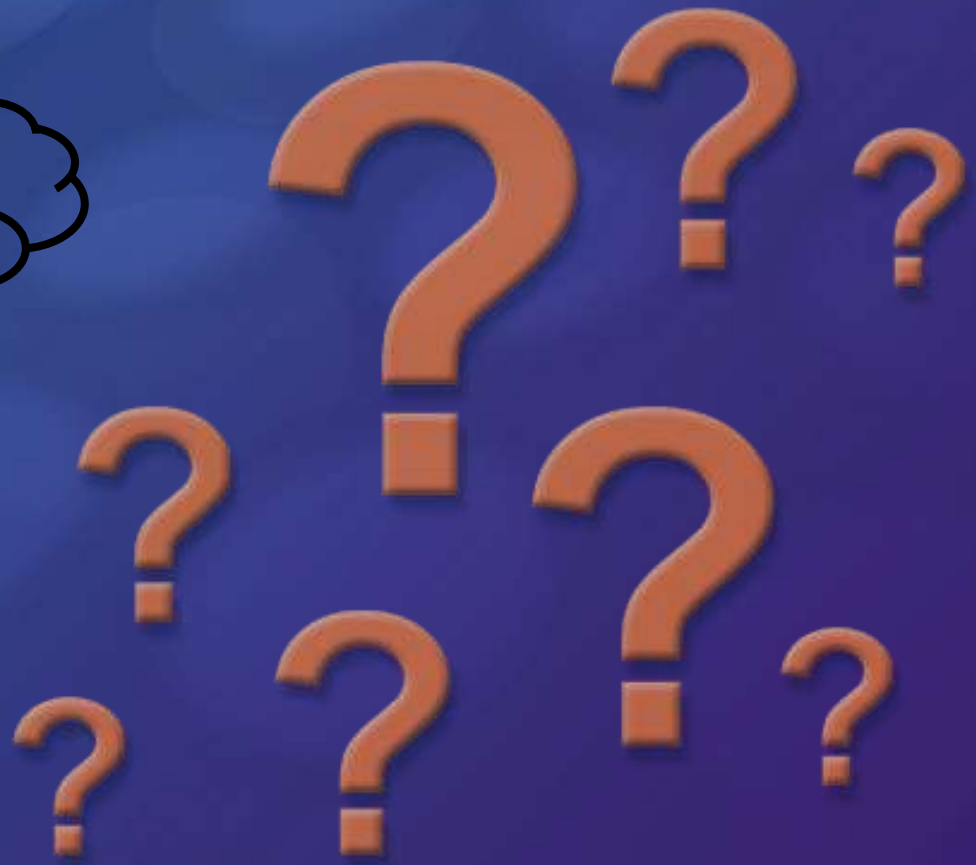
<p>i Visual Studio Team System 2008 (Teil 1 von 10) - Team Foundation Server - Mehr als nur eine Versionsverwaltung! Presenter: Neno Loje Typ: Serien-Webcast Technologiebereich: Dauer: N/A</p>	Teamentwicklung, Quellcodeverwaltung und Testen	100	-	20.02.2008
<p>i Visual Studio Team System 2008 (Teil 2 von 10) - Team Foundation Server 2008 - Was ist neu? Presenter: Neno Loje Typ: Standard Webcast Technologiebereich: Dauer: N/A</p>	Teamentwicklung, Quellcodeverwaltung und Testen	100	-	27.02.2008
<p>i Visual Studio Team System 2008 (Teil 3 von 10) - Auf geht's: Installation und Migration auf den Team Foundation Server 2008 Presenter: Neno Loje Typ: Serien-Webcast Technologiebereich: Dauer: N/A</p>	Teamentwicklung, Quellcodeverwaltung und Testen	100	-	05.03.2008
<p>i Visual Studio Team System 2008 (Teil 4 von 10) - TFS 2008 für Entwickler: Arbeiten mit der neuen Versionsverwaltung Presenter: Neno Loje Typ: Serien-Webcast Technologiebereich: Dauer: N/A</p>	Teamentwicklung, Quellcodeverwaltung und Testen	200	-	12.03.2008
<p>i Visual Studio Team System 2008 (Teil 5 von 10) - TFS 2008: Mit definierten Buildprozessen und Continuous Integration zu mehr Softwarequalität Presenter: Neno Loje Typ: Serien-Webcast Technologiebereich: Dauer: N/A</p>	Teamentwicklung, Quellcodeverwaltung und Testen	200	-	19.03.2008
<p>i Visual Studio Team System 2008 (Teil 6 von 10) - TFS 2008 für Projektleiter: Projektmanagement, Arbeitsaufgaben und Berichte Presenter: Lars Roith Typ: Serien-Webcast Technologiebereich: Dauer: N/A</p>	Teamentwicklung, Quellcodeverwaltung und Testen	100	-	27.03.2008
<p>i Visual Studio Team System 2008 (Teil 7 von 10) - Datenbankentwicklung als Teil des Softwareentwicklungsprozesses Presenter: Dariusz Parys Typ: Serien-Webcast Technologiebereich: Dauer: N/A</p>	Teamentwicklung, Quellcodeverwaltung und Testen	200	-	02.04.2008
<p>i Visual Studio Team System 2008 (Teil 8 von 10) - Qualitätssicherung und Presenter: Neno Loje Typ: Serien-Webcast Technologiebereich: Dauer: N/A</p>	Teamentwicklung, Quellcodeverwaltung und Testen	100	-	09.04.2008

microsoft.de/msdn/webcasts/serien/MSDNWCS-0802-01.msp



Noch Fragen?:

nenno@teamfoundationserver.de



Microsoft[®]

Urheberrechtlicher Hinweis

- Dieser Vortrag wurde von Neno Loje (www.teamsystempro.de) im Auftrag der Microsoft Deutschland GmbH konzipiert.
- Einige Teile (C) Microsoft, USA.