

MSDN Webcast:

TFS 2008 für Entwickler: Arbeiten mit der neuen Versionsverwaltung
Visual Studio Team System (Teil 4 von 10)

Veröffentlicht: 12.03.2008

Presenter:

Neno Loje, MVP für Team System

www.teamsystempro.de



Wer sind Sie?

- Sie
 - ...wollen weg von ihrem jetzigen System?
 - Performanz, weltweite Teams, Parallelarbeit
 - ... haben bereits auf TFS gewechselt?
 - ... möchten sich informieren welche Vorteile ein Wechsel zum TFS für Sie hätte?

Agenda

- TFVC vorgestellt
- SCM-Konzepte im TFS
- Zugriffsmöglichkeiten
- Weitere Themen



Agenda

- **TFVC vorgestellt**
- SCM-Konzepte im TFS
- Zugriffsmöglichkeiten
- Weitere Themen



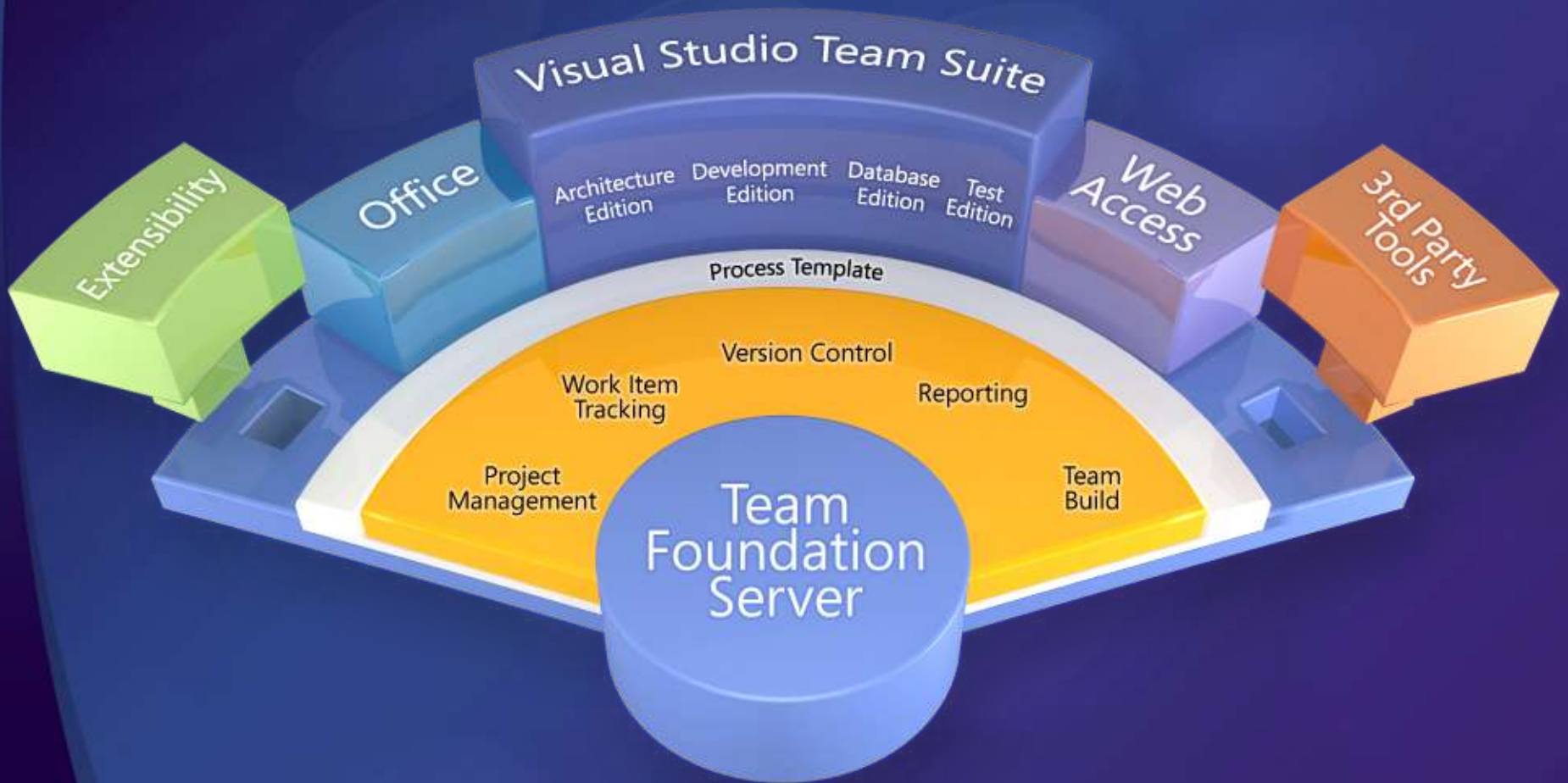
Begrifflichkeiten

- VSTS
 - Visual Studio Team System
- TFS
 - Team Foundation Server
- TFVC
 - Team Foundation Version Control
(die Versionsverwaltung im TFS)

Marketingnamen

- VSTS
 - Microsoft® Visual Studio® Team System 2008
- TFS
 - Microsoft® Visual Studio® Team System 2008
Team Foundation Server
- TFVC
 - Team Foundation Version Control
(die Versionsverwaltung im TFS)

Visual Studio Team System



TFVC vorgestellt

- TFVC
 - ... ist eine vollständige Neuentwicklung
 - ... wurde für große Teams, die auch weltweit verteilt arbeiten, optimiert

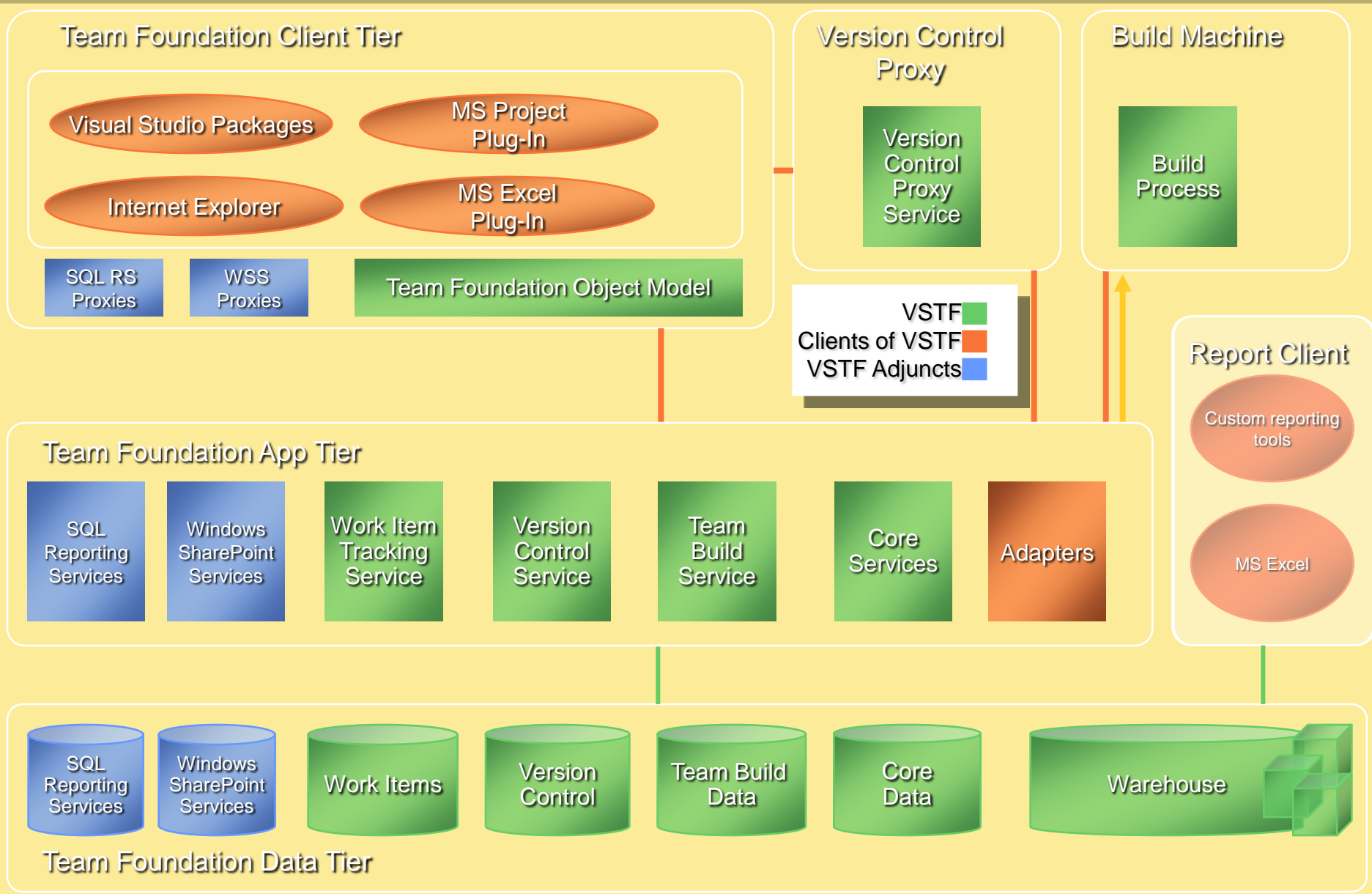
Was erwarten wir von einer Versionsverwaltung?

- Die Aufbewahrung unseres "Master"-Quellcodes
- Die "Versionierung" des Quellcodes
 - d.h. den Mastercode zu einem bestimmten Punkt auf der Zeitachse (z.B. zu einem Build)

Was gehört alles in eine Versionsverwaltung?

- Alles, was sie dem Auftraggeber später aushändigen
 - Quellcode
 - Tests
 - Setup
 - Endkundendokumentation & Beispieldateien
 - Datenbank-Schema
 - ...

Architektur der TFVC



Wie komme ich zum TFS?

- Hardware / Software
- Installation & Konfiguration
- Migration

Migration?

1. Manuelle Migration ("zu Fuß")
 - Verlust der Historie
2. Toolgestützte Migration ("Einbahnstraße")
 - Historie bleibt erhalten, Aufwand für Mapping von Benutzernamen und Ordnern
3. Synchronisation ("Spiegelung")

Agenda

- TFVC vorgestellt
- **SCM-Konzepte im TFS**
- Zugriffsmöglichkeiten
- Weitere Themen



Team Foundation Version Control

Workspaces

Changesets

Locks

Policies

Shelves

Branches

Team Foundation Version Control

Workspaces

Changesets

Locks

Policies

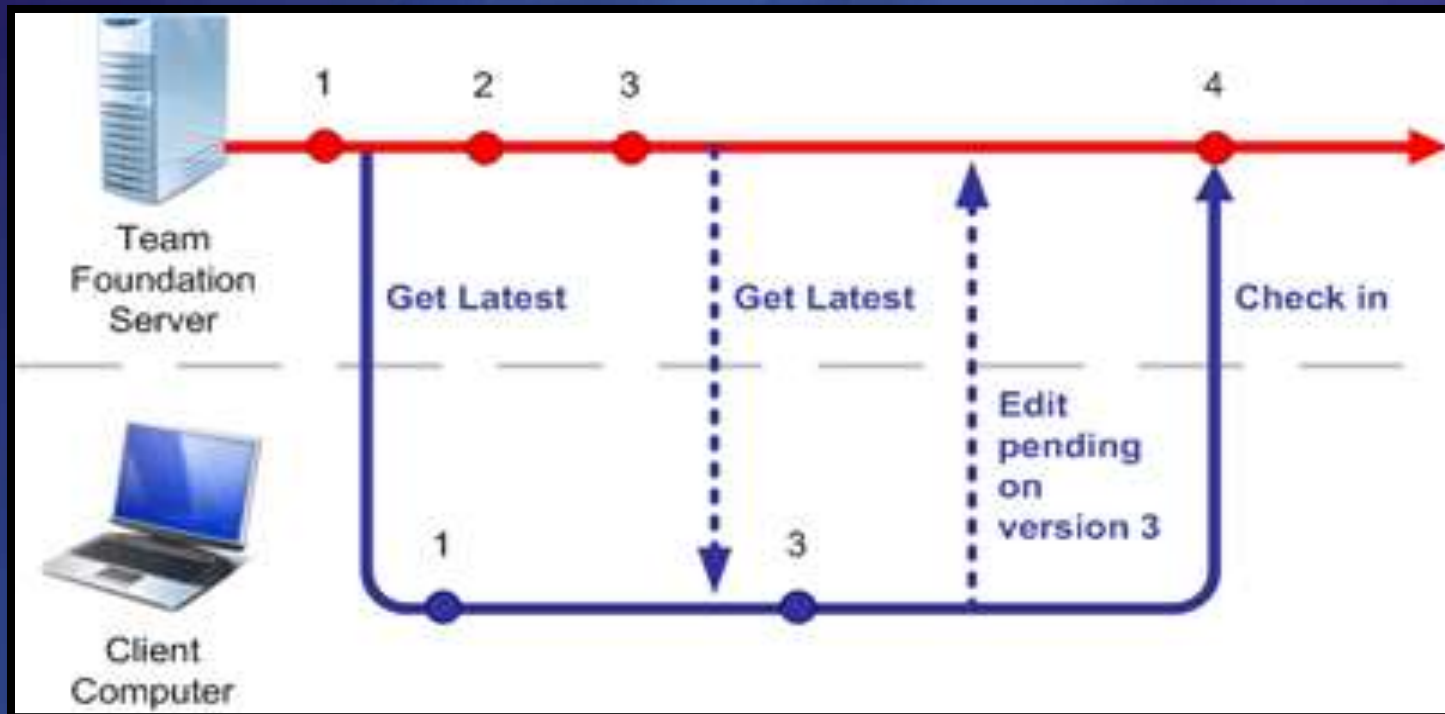
Shelves

Branches

Workspaces

- Lokaler "Sandkasten" (= clientseitige Kopie), in dem gearbeitet wird
 - Befehle auf dem Sandkasten werden immer erst lokal "gemerkt" (ausstehende Eincheckvorgänge)
- Zuordnung zwischen lokalen Ordnern und Ordnern in der TFVC
 - $\$ \backslash \text{ProjektA} \rightarrow D: \backslash \text{Projekte} \backslash \text{ProjektA}$
 - $\$ \backslash \text{ProjektB} \rightarrow D: \backslash \text{Projekte} \backslash \text{ProjektB}$
- Workspaces gehören jeweils einem Benutzer und einem Computer

Workspaces: Bildlich



Workspaces: Empfehlungen

- Empfehlung: Workspaces von Hand anlegen!
 - VS legt automatisch einen Workspace an
- Lokale Struktur auf Festplatte = TFS-Struktur
 - `$/TeamProject1` → `D:\Projekte\TeamProject1`
 - `$/TeamProject2` → `D:\Projekte\TeamProject2`
 - ...
- Man kann auch mehrere Workspaces auf einer Maschine einrichten
 - „Pending Changes“ Fenster in VS zeigt jeweils nur Elemente aus einem Workspace an

Team Foundation Version Control

Workspaces

Changesets

Locks

Policies

Shelves

Branches

Einchecken

- Die Änderungen werden samt Metadaten als so genanntes "Changeset" übertragen
 - Das Einspielen erfolgt transaktional
- Metadaten
 - Kommentar
 - Beliebige Anzahl an Eincheck-Notizen
 - Verknüpfung zu Aufgaben

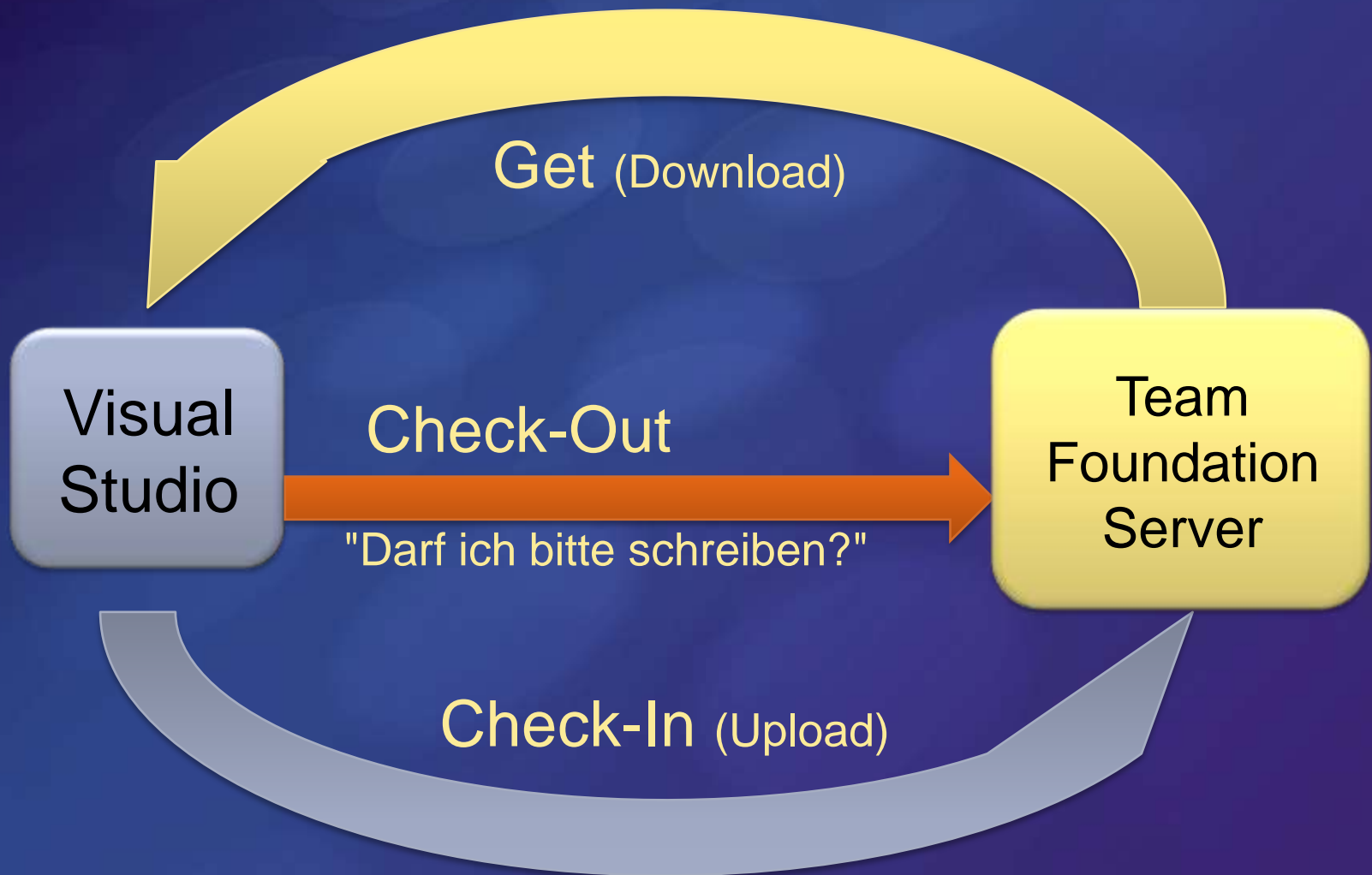
Verhalten vom Auschecken

- Auschecken ist missverständlich
 - Auschecken bedeutet:
"Ich möchte die Datei **eventuell** bearbeiten"
- TFVC macht die Dateiversion im Workspace **schreibbar**. Mehr nicht.
 - Um aktuellste Version zu bekommen muss vorher manuell **Get** (Latest) aufgerufen werden!

TFVC-Vokabular

- Check-In
= Upload
- Check-Out
= "Ich-möchte-schreiben"
- Get
= Download

Check-In vs. Check-Out vs. Get



Team Foundation Version Control

Workspaces

Changesets

Locks

Policies

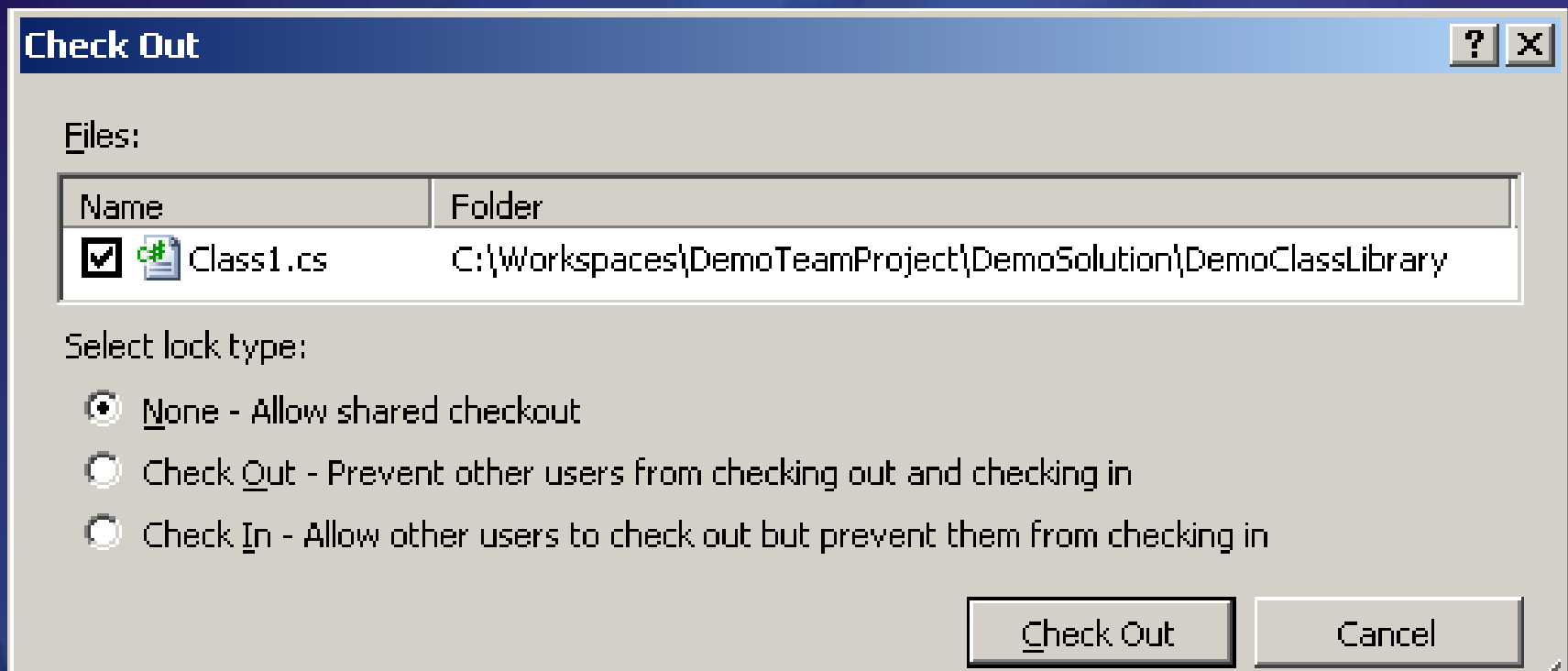
Shelves

Branches

Sperren

- Zwei Extreme:
 - Jeder soll alles gleichzeitig bearbeiten dürfen ("Multiple Checkout")
 - Es darf nur einer zur Zeit an etwas arbeiten ("Exclusive Checkout")
- Standardmäßig nutzt der TFS ersteres
 - An-/ausschaltbar pro Projekt
 - Pro Dateityp (für den gesamten TFS gültig)
 - Bei jedem Checkout explizit sperrbar

Locks: Bildlich



Team Foundation Version Control

Workspaces

Changesets

Locks

Policies

Shelves

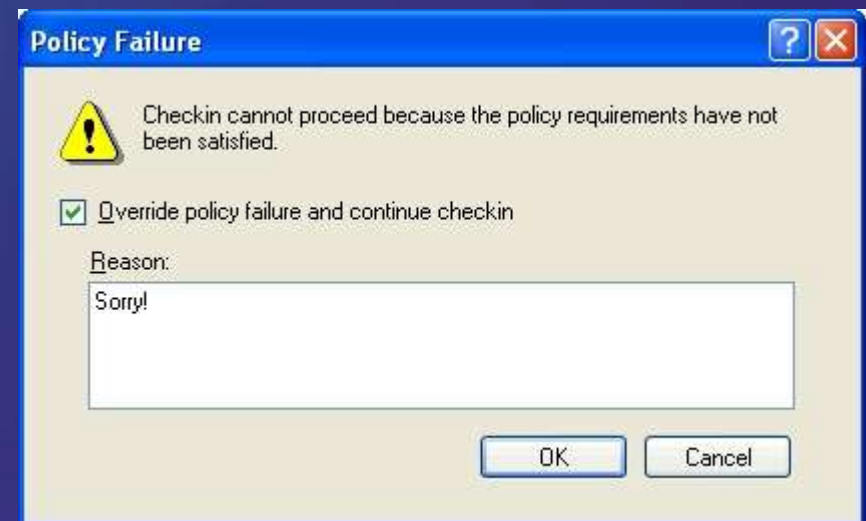
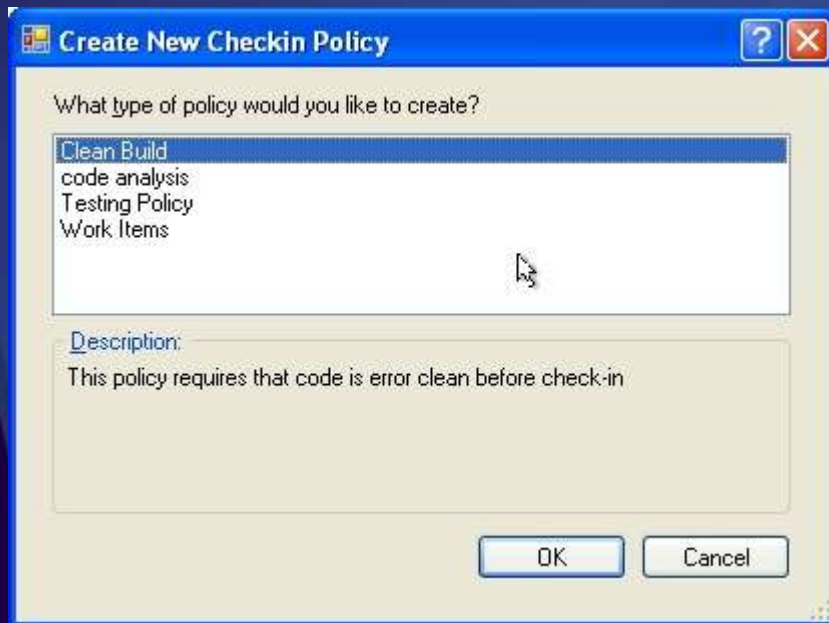
Branches

Eincheckrichtlinien

Check-In Policies



- Pro Projekt vom Projektleiter einstellbar
- Richtlinien prüfen den Code lokal und entscheiden ob eingchecked werden darf
- Erweiterbar durch eigene Eincheckregeln



Policy Override



- Eincheckrichtlinien sollen den Benutzer "erinnern", nicht etwas erzwingen
- Benutzer können das negative Ergebnis der Eincheckrichtlinien umgehen ("Override")
- Kann man diese Funktion abschalten?
 - Leider nein, aber sich benachrichtigen lassen
 - Entweder sofort (per E-Mail) oder über Berichte

Fertig einsetzbare Eincheckrichtlinien

- In VS 2008 eingebaute Richtlinien:
 - Work Item Policy
 - Code Analysis
 - Test Policy
 - CI Build not failed Policy
 - Custom Path Policy
 - Forbidden Patterns Policy
 - Changeset Comments Policy
 - Work Item Query Policy



Ideen für neue Eincheckrichtlinien

- Geplante und verwirklichte Eigenproduktionen:
 - Branching Forbidden Policy
 - Checkin Notes Pattern Policy
 - Checkin Time Restriction Policy
 - Clean Build Policy
 - Confirmation Required Policy
 - Deleting Forbidden Policy
 - Maximum File Size Policy
 - No Checkins Allowed Policy
 - Local Backup Policy
 - Run Custom Tool Policy
 - Latest Policies Installed Policy



Nicht vergessen:
DLL muss auf allen Clients
installiert werden

Team Foundation Version Control

Workspaces

Changesets

Locks

Policies

Shelves

Branches

Grundlagen: Shelving

- Das Dilemma:
 - Es soll täglich eingecheckt werden
 - Gründe: Datensicherung, mögl. Erkrankung, ...
 - Der Code in der Versionsverwaltung soll von hoher Qualität sein und immer kompilieren
 - Gründe: Daily Build, Prüfen von Coderegeln, ...



- Die Lösung: „Shelves“

Shelving in Aktion



- Ausgangsfrage: Habe ich "fertigen" Code?
 - Fertig = erfüllt die Unternehmensrichtlinien
- Ja?
 - Check-In: In die Versionsverwaltung einchecken
 - Eincheckrichtlinien werden geprüft
 - Änderungen fließen in den Buildprozess mit ein
- Nein?
 - Shelve-In: In eigenen Shelve ablegen
 - Keine Auswirkungen auf andere Benutzer

Einsatzmöglichkeiten für Shelving

- Unfertige Arbeit sichern
 - z.B. bei Unterbrechungen, Feierabend, etc.
- Weitergabe von unfertiger Arbeit
 - Kollege ist Experte auf einem bestimmten Gebiet
- Code Review
 - Freiberufler sollen nicht direkt einchecken
- Zwischenspeicherung (Checkpoints)
 - Aktuellen Workspace-Arbeitsstand sichern
- Wechsel zwischen parallelen Aufgaben

Hinweis zum Shelving und exklusiven Auschecken

- Shelves setzen multiple Checkout voraus.

Team Foundation Version Control

Workspaces

Changesets

Locks

Policies

Shelves

Branches

Quellcodestruktur und Branching

- Die Quellcodestruktur sowie die Branches (=Kopien) sind nicht in Stein gemeißelt und lassen sich jederzeit wieder umorganisieren, wenn es die Umstände erfordern
- Branches sind leichtgewichtig
 - Keine Performanzeinbußen bei vielen Branches
 - Aber: Organisationsoverhead
- Branches sind nur logische Kopien
 - Werden als Ordner dargestellt (und im lokalen Workspace auch physikalisch als Kopie angelegt)

Quellcodestruktur (Standard)

- Automatisch von VS vorgegebene Standardstruktur:
- \$/
 - TeamProjekt
 - Visual Studio-Projektmappe (.sln)
 - Visual Studio-Projekt (.csproj)
 - Projektunterordner

Branching-Anforderungen

- Dazwischen werden jetzt noch benötigte Branching-Ebenen (Ordner) eingeschoben
- Beispielszenario:
 - Es sollen immer die aktuelle Version auf dem Markt und die nächste in Entwicklung befindliche Version zum Bearbeiten zur Verfügung stehen.
 - Das Projekt besteht aus Solutions, die in jeder Version unterschiedlich sind (keine Gemeinsamkeiten haben)

Branching-Anforderungen (II)

- Mögliche Lösung:
 - `$/TeamProjekt/`
 - Branch "**Produktion**" (= beim Kunden)
 - Visual Studio-Projektmappe (.sln)
 - Visual Studio-Projekt (.csproj)
 - Projektunterordner
 - weitere Ordner...
 - weitere Visual Studio-Projekte ...
 - weitere Projektmappen ...
 - Branch "**Entwicklung**" (= Zukunft)
 - Visual Studio-Projektmappe (.sln)
 - Visual Studio-Projekt (.csproj)
 - Projektunterordner
 - Solutions (die nicht "gebrancht" werden)

Paralleles Arbeiten: Isolationsstufen

1. Workspaces

- Lokale
- Für 1 Computer und 1 Benutzer

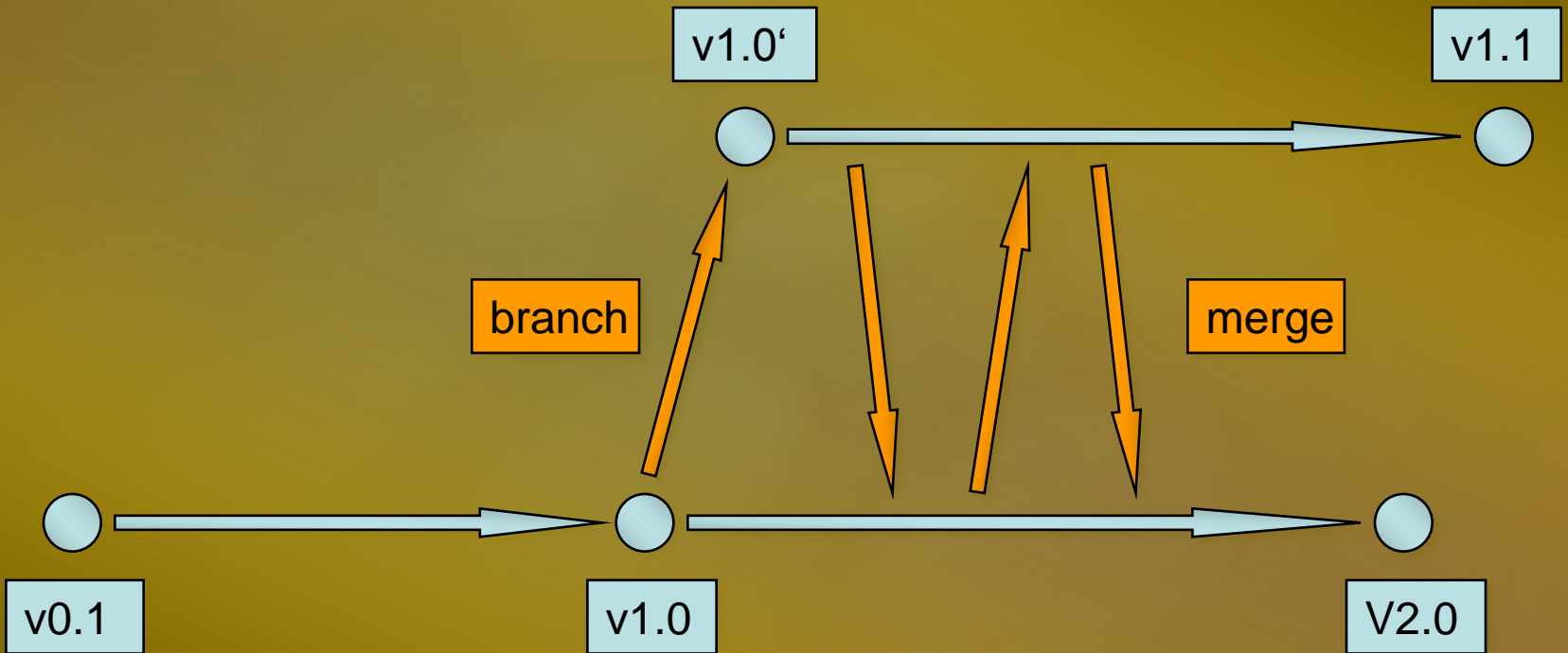
2. Shelves

- Serverseitig
- Es kann 1 Person zur Zeit dran arbeiten

3. Branches

- Serverseitig
- Mehrere Personen gleichzeitig

Branches



Wie viele Branches braucht mein Projekt?

- Branches
 - ... verursachen Aufwand
 - ... können auch nachträglich angelegt werden
- Gründe für Branches
 1. Unterstützung ausgelieferter Versionen
 2. Isolation von paralleler Arbeit
 3. Experimentelles Arbeiten

"Shared Components"

- Umgang mit gemeinsam genutzten Komponenten
 - Branches!

Shared Components mit Branches

- Gemeinsam genutzte Komponenten

- Komponenten-DLLs

- Projekt A

- Quellcode

- Komponenten-DLLs

- Projekt B

- Quellcode

- Komponenten-DLLs

ref



ref



Agenda

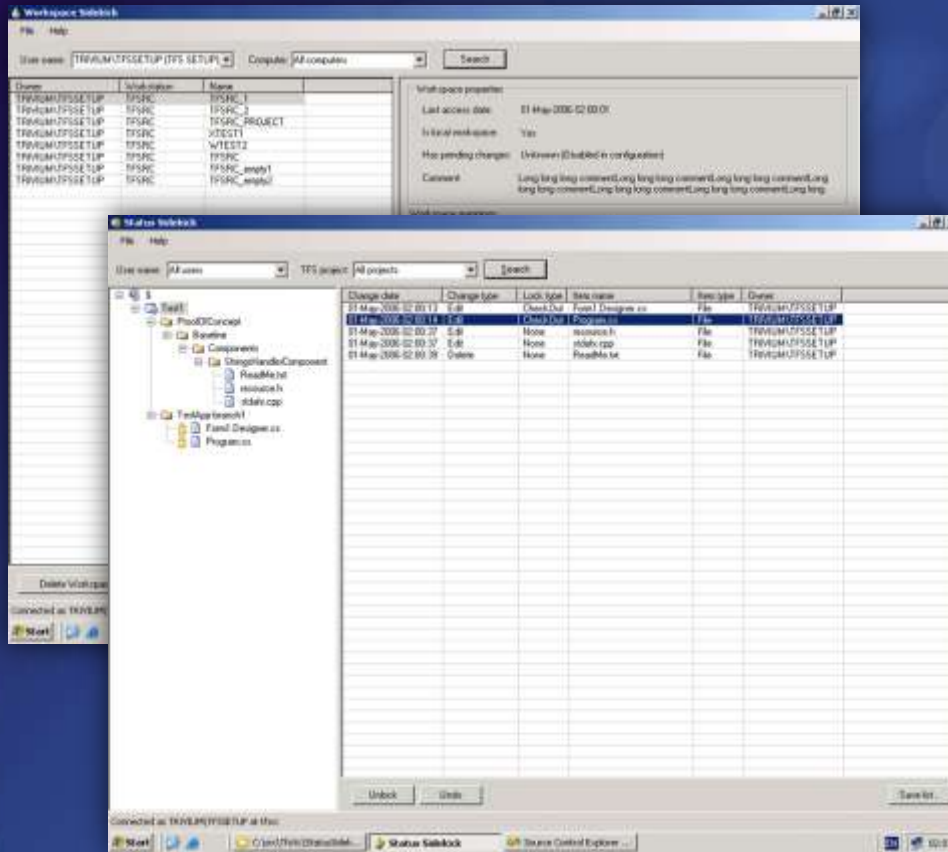
- TFVC vorgestellt
- SCM-Konzepte im TFS
- **Zugriffsmöglichkeiten**
- Weitere Themen



Zugriffsmöglichkeiten auf die TFVC

- Windows: Team Explorer
 - TE und VS-Version müssen übereinstimmen d.h. TE2005 für VS2005, auch bei TFS 2008
- DOS-Feeling: Kommandozeile
- Web: Team System Web Access
 - Nur lesend
- VS6, VB6, etc.: MSSCCI-Provider
- Eclipse IDE, Linux, MacOS: Teamprise (€)
- Windows Explorer: Diverse Plug-Ins

Nette Helferlein



Kostenfreies Werkzeug:

- **Workspace Sidekick**
Oberfläche zur komfortablen Administration von Workspaces
- **Status Sidekick**
Anzeige aller anstehenden und erfolgten Vorgänge
- **History Sidekick**
- **Label Sidekick**
- **Shelve Sidekick**

<http://www.attrice.info/cm/tfs/>

Agenda

- TFVC vorgestellt
- SCM-Konzepte im TFS
- Zugriffsmöglichkeiten
- **Weitere Themen**



Weitere Themen

- Unterschiede VSS zu TFVC
- Offline arbeiten
 - Mit Visual Studio 2005 und Power Tools
 - Mit Visual Studio 2008
- Verteiltes Arbeiten
 - Version Control Proxy
- Skalierbarkeit & Performanz

Weitere Themen II

- Destroy
- Label
- Security Permissions
- Command Line support
- Compare
- Annotate
- Power Tools

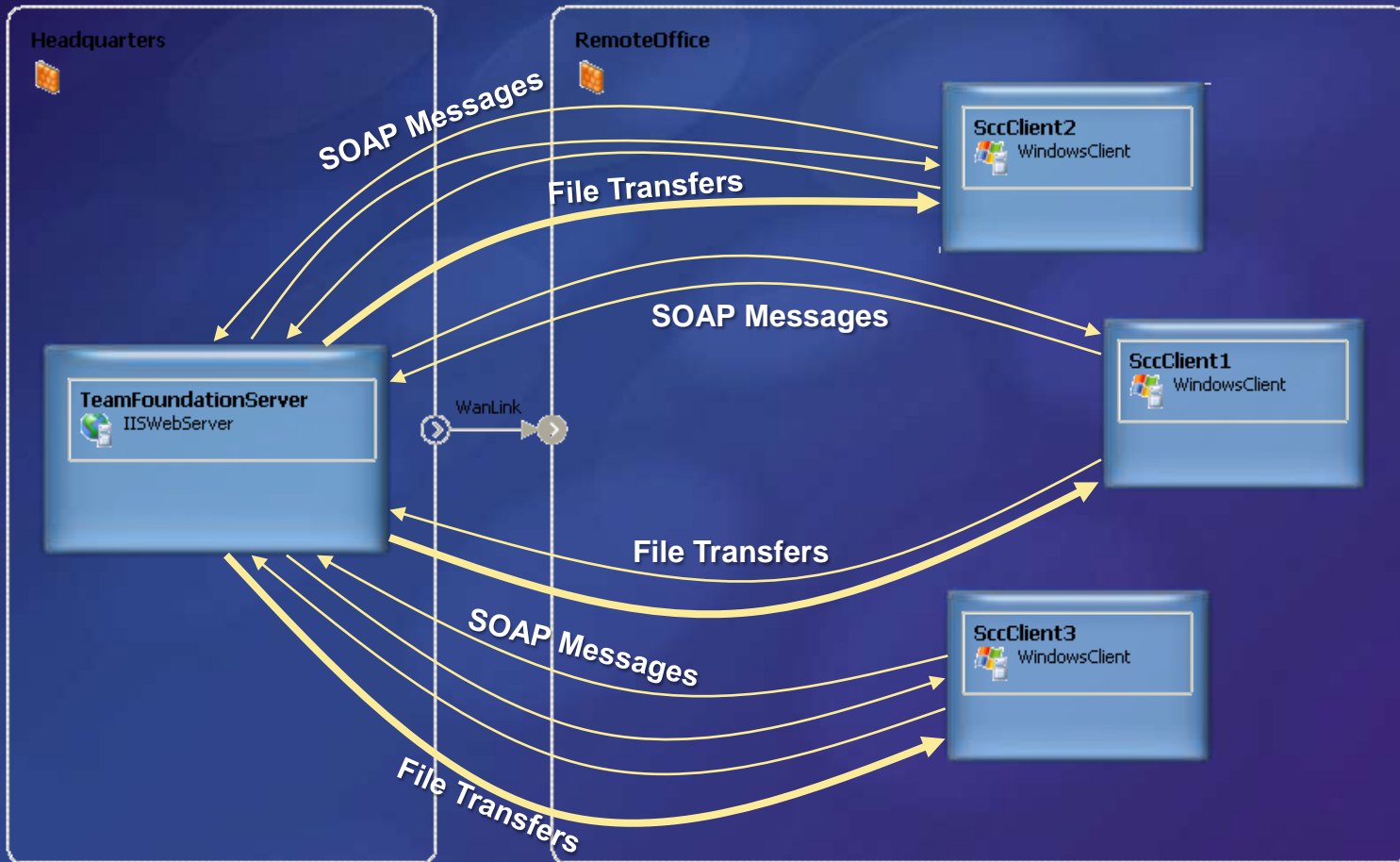
Source Control Feature Support

	Yes	No
Atomic Checkin	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Work Item Integration	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Checkin Policies	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Shelving	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Delta File Storage	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Delta Binary File Storage	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Large File Support (>4GB)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Distributed Team Support	<input checked="" type="checkbox"/>	<input type="checkbox"/>

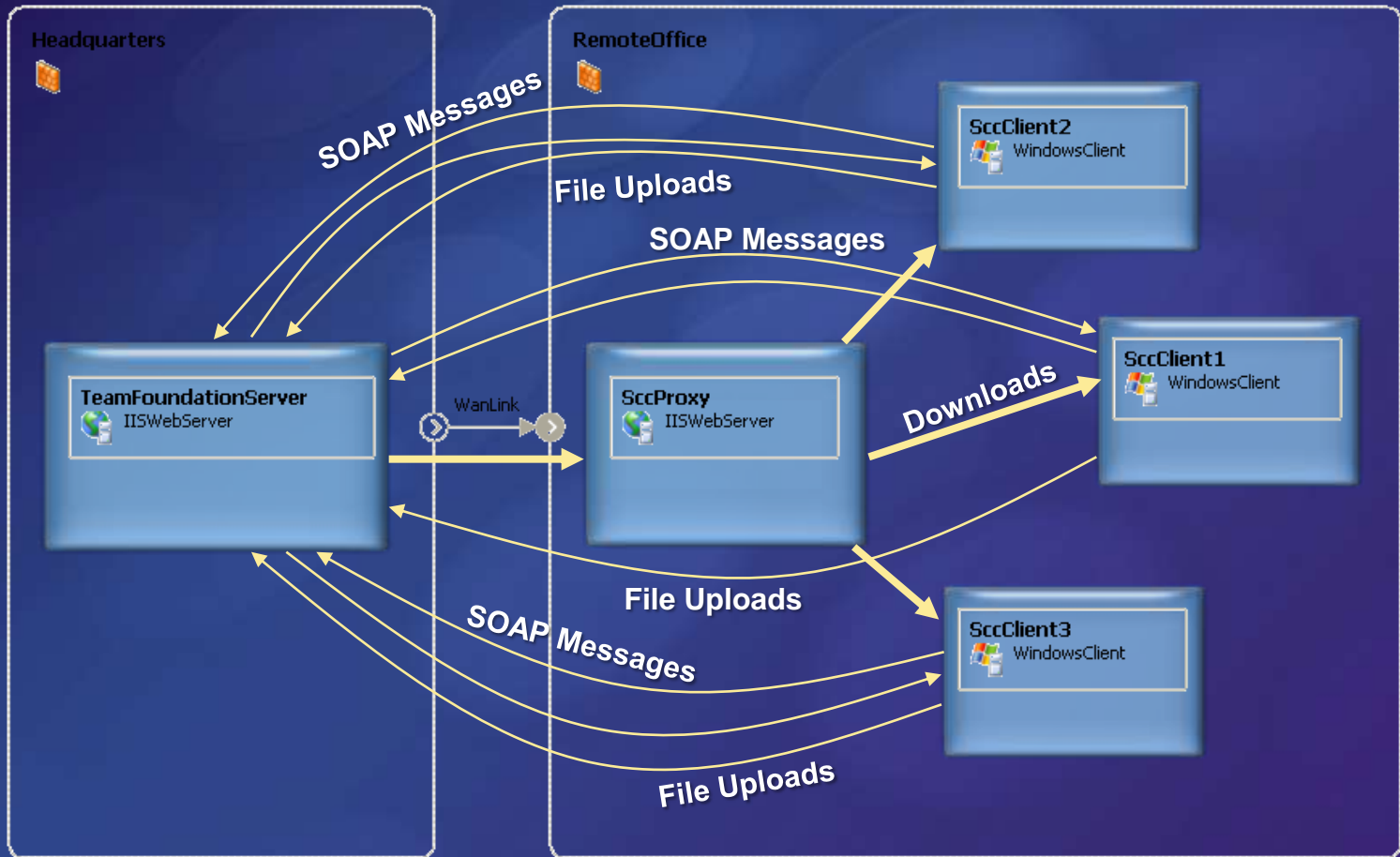
Source Control Feature Support

	Yes	No
E-mail Checkin Notification	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Non-Windows Support	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Diff Tool Extensibility	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Shared Checkout	<input checked="" type="checkbox"/>	<input type="checkbox"/>
VS 2003 Integration	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Keyword Expansion	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Pinning & Sharing	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Shadow Folders	<input type="checkbox"/>	<input checked="" type="checkbox"/>

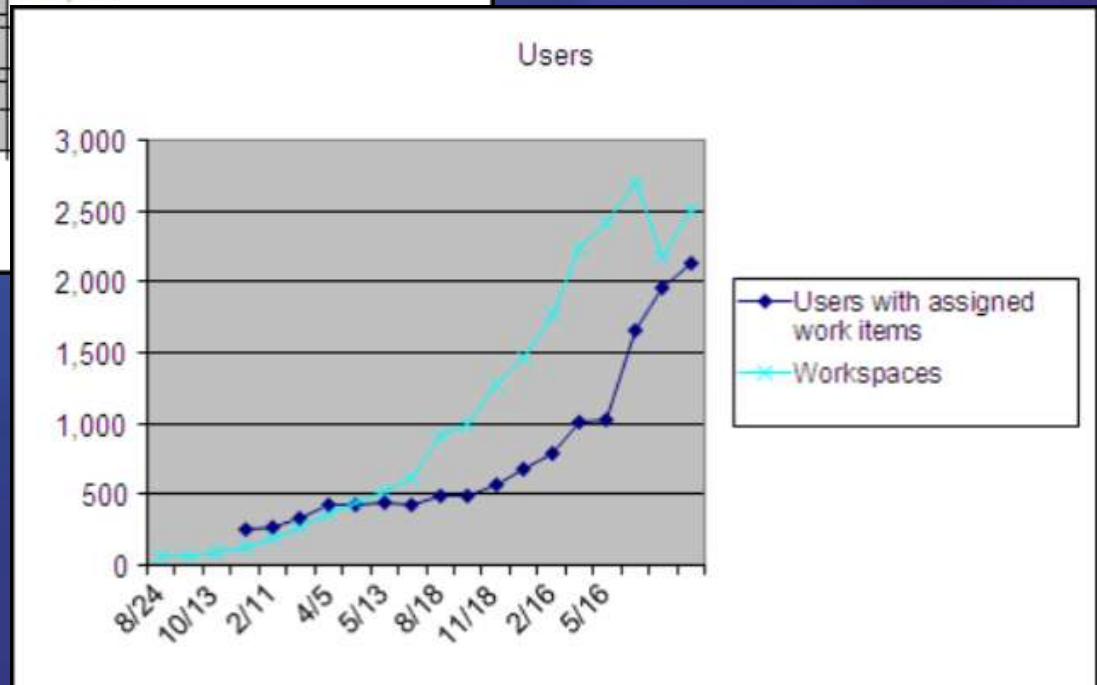
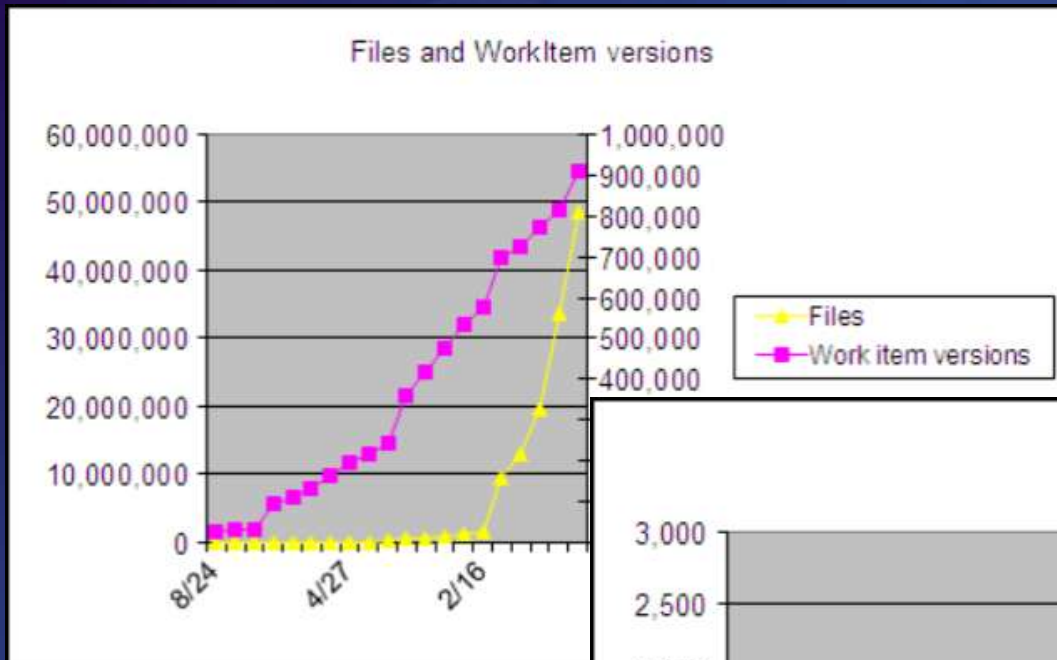
Verteiltes Arbeiten ohne Version Control Proxy



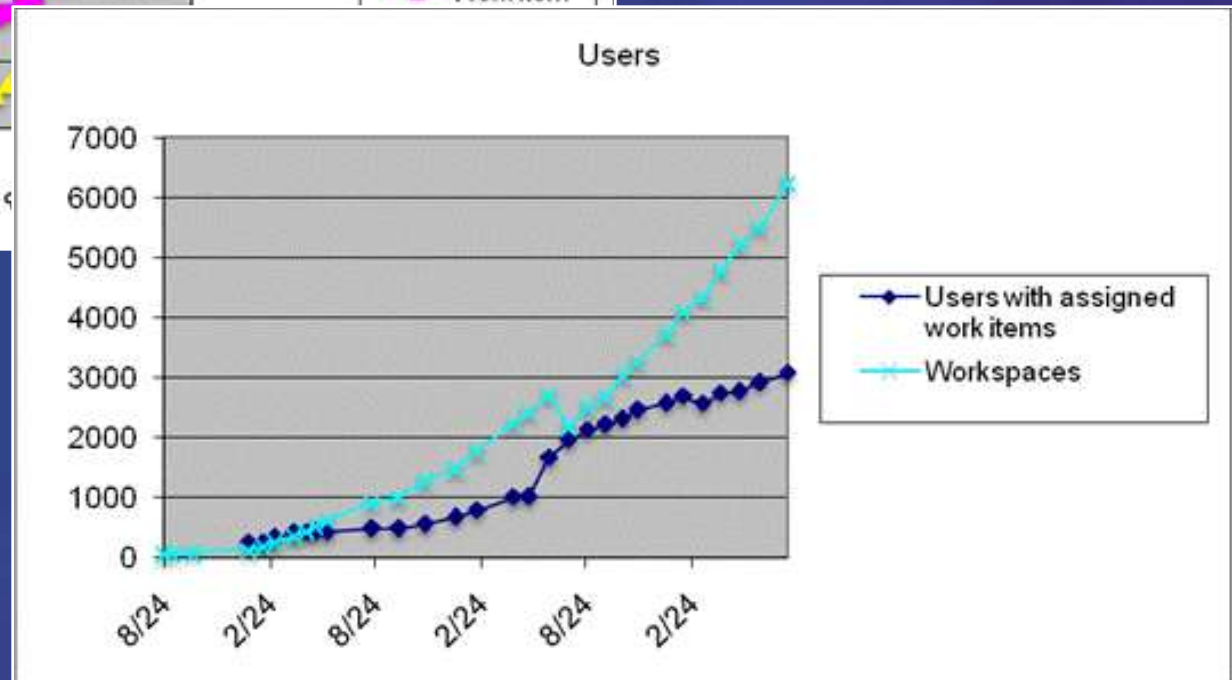
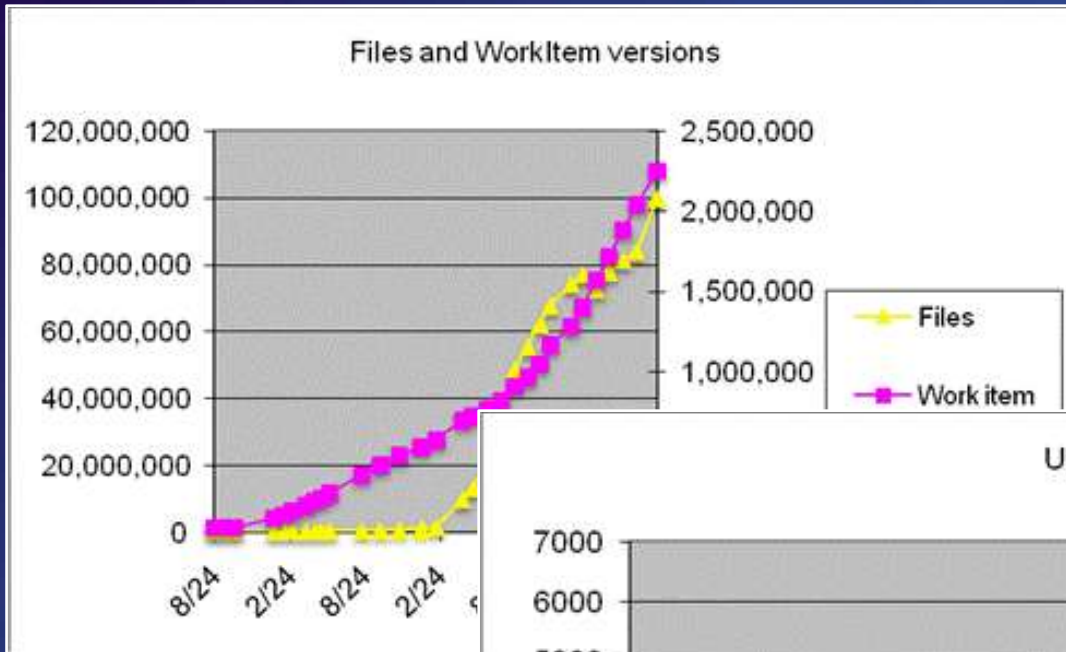
Verteiltes Arbeiten mit Version Control Proxy



"Hundefütterung" bei Microsoft (2006)



"Hundefütterung" bei Microsoft (2007)



TFS Dogfood Statistics – August 07

Users

- Recent users: 1,015 (down 15)
- Users with assigned work items: 3,073 (up 157)
- Version control users: 2,649 (up 231)

Version control

- Files/Folders: 99,559,163/21,692,272 (up 15,613,488/up 3,850,160)
- Total compressed file size: 1,170 GB (up 181 GB)
- Checkins: 253,993 (up 22,629)
- Shelvesets: 12,844 (up 1,745)
- Merge history: 232,689,548 (up 38,096,682)
- Pending changes: 3,934,204 (up 1,533,477)
- Workspaces: 6,224 (up 719)
- Local copies: 695,274,358 (up 90,790,224)

Work Items

- Work Items: 268,635 (up 23,239)
- Areas & Iterations: 7,575 (up 72)
- Work item versions: 2,249,745 (up 205,077)
- Attached files: 98,989 (up 11,251)
- Queries: 18,936 (up 1,065)

Commands (last 7 days)

- Work Item queries: 318,036 (down 18,800)
- Work Item updates: 31,651 (up 3,495)
- Work Item opens: 225,279 (down 115,081)
- Gets: 586,788 (up 521,611)
- Downloads: 35,919,563 (up 16,712,610)
- Checkins: 5,275 (up 3,011)
- Uploads: 134,830 (down 225,997)
- Shelves: 1,313 (up 381)

Zusammenfassung

- Der TFS hat keine VSS-Altlasten
 - Performanz, verteiltes Arbeiten, neue Ideen
 - Einiges ist anders, aber nichts unlösbar

Anders als in VSS

- Änderungen erst lokal im Workspace
 - erst nach Check-In ist es für alle sichtbar
- Check-Out holt nicht die neuste Version
 - einstellbar (ab TFS 2008)
- Links werden durch Branching abgebildet und haben nun einen expliziten Schritt
- Standardmäßig ist mehrfaches Auschecken erlaubt

Nächste Schritte

1. Ziehen Sie um
 - Migration des Quellcodes
2. Starten Sie mit zwei Branches
 - Und erzeugen bei Bedarf neue
3. Nutzen Sie "Shelves" für unfertige Arbeit

Weitere Informationen

- Team Development with TFS Guide – Pattern & Practices
<http://www.codeplex.com/TFSGuide>
- Operations Guidance for Team Foundation Server
[http://msdn2.microsoft.com/en-us/library/bb663036\(VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/bb663036(VS.80).aspx)

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: 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.