

# Einführung in SVN

Jonathan Dimond

April 2009

# Aufgaben einer Versionsverwaltung

- zeichnet alle Veränderungen an einem Projekt in einem zentralen Repository auf
- bietet die Möglichkeit Änderungen rückgängig zu machen
- erlaubt das gleichzeitige Arbeiten am gleichen Projekt
- ermöglicht so effiziente Teamarbeit

# Warum benutzt man eine Versionsverwaltung?

- Teamarbeit wäre ohne Versionsverwaltung kaum möglich
- selbst schwerwiegende Fehler sind in Sekunden rückgängig gemacht
- Arbeiten mit Trunk/Branches/Tags:
  - trunk/: Hauptentwicklungszweig
  - tag/: markiert die stabilen Versionen
  - branch/: für alternative Versionen

# Geschichte von SVN

- ab 2000 entwickelt
- 2004 stabile Version 1.0 herausgekommen
- Nachfolger von CVS
- behebt einige Schwächen von CVS

# Vorteile von SVN

- Einfach zu bedienen
- Gute Integrierung in Eclipse mit Subclipse

# Wichtigsten Aktionen von SVN

- Checkout: Erstellt eine lokale Kopie(sogenannte Working Copy) des Repositories auf dem Rechner
- Update: Aktualisiert die lokale Kopie auf den Stand des Repositories
- Commit: Fügt die lokalen Änderungen dem Repository hinzu
- Add: Fügt eine neue Datei ins Repository ein (nicht vergessen!)
- Merge: Führt Änderungen bei Konflikten zusammen
- Revert: Setzt das Repository auf eine alte Revision zurück

# Subclipse

Subclipse ist ein Plugin für Eclipse das es ermöglicht direkt aus Eclipse heraus auf das SVN Repository zuzugreifen

# Installation

Installationsanleitung findet man unter:

`http://www-128.ibm.com/developerworks/opensource/library/os-ecl-subversion/`

# Arbeiten mit Subclipse

- am Anfang jedes neuen Projekts ein Checkout durchführen
- vor jeder Projektbearbeitung ein Update durchführen
- nach jeder wichtigen Änderung ein Commit
- bei jedem Konflikt ein Merge, sonst droht Chaos
- optional nach einer stabilen, vorführfertigen Version ein Tag setzen um jederzeit auf die stabile Version zugreifen zu können