

Versionsverwaltung

Wechseln zu: [Navigation](#), [Suche](#)

Inhaltsverzeichnis

- 1 Definition
- 2 Open-Source-Versionsverwaltungen
 - 2.1 SSCS
 - 2.2 RCS
 - 2.3 CVS
 - 2.3.1 Vorteile
 - 2.4 SVN
 - 2.4.1 Vorteile
- 3 Quellen

1 Definition

Eine Versionsverwaltung hat die Aufgabe, den Werdegang eines Projektes zu protokollieren. Sie übernimmt die Versionierung und die Archivierung der Daten und ermöglicht den gemeinsamen Zugriff darauf. Dabei kann jede Änderung und jeder Stand verfolgt, rückgängig gemacht oder wiederhergestellt werden.

2 Open-Source-Versionsverwaltungen

2.1 SSCS

Das **Source Code Control System** (SCCS) war eine der ersten Versionsverwaltungen. Es wurde Anfang der Siebziger von Marc Rochkind bei AT&T entwickelt. Dieses System konnte schon mehrere Versionen einer Datei verwalten. Da es aber in jeder Hinsicht modernen Systemen unterlegen ist, existieren heutzutage praktisch keine Projekte mehr damit.

2.2 RCS

Das **Revision Control System** (RCS) war die erste große Versionsverwaltung im **Open-Source-Bereich**. Es wurde 1985 von Walter Tichy an der Purdue University entwickelt. Dabei kam zum ersten Mal eine Technik mit dem Namen Lock-Modify-Unlock-Mechanismus zum Einsatz. Damit können mehrere Benutzer auf eine gemeinsame Datei lesend zugreifen. Soll eine Änderung vorgenommen werden, muß die entsprechende Datei zuerst gesperrt werden. Danach kann kein anderer Benutzer die Datei mehr bearbeiten, bis diese wieder freigegeben wird. So wird zu jeder Zeit sichergestellt, daß nicht mehrere Benutzer gleichzeitig eine Datei verändern.

2.3 CVS

Die Weiterentwicklung von RCS ist das **Concurrent Versions System** (CVS). In den Anfängen, noch von Dick Grune in Shell-Skripten implementiert, wurde es dann 1989 von Jeff Polk und Brian Berliner in **C** komplett neu implementiert. Die größten Neuerungen waren sicherlich die eingeführte Client-Server-Architektur und der implementierte Copy-Modify-Merge-Mechanismus. Im Gegensatz zu RCS wird nichts gesperrt, sondern der aktuelle Stand vor einer Änderung in einen lokalen Workspace kopiert, indem dann auch die Änderungen stattfinden. Bei der abschließenden Veröffentlichung der Daten werden die Änderungen mit dem zentralen Repository wieder abgeglichen. Dadurch war es zum ersten Mal möglich, daß viele Benutzer gleichzeitig am selben Projekt, ja sogar in denselben Dateien, arbeiten konnten. Unterstützend kam noch hinzu, daß nicht, wie bisher, jede Datei komplett unabhängig von den anderen betrachtet wurde, sondern, daß nun Gruppen von Dateien zu Modulen zusammengefaßt werden konnten.

2.3.1 Vorteile

Die Hauptvorteile – als Verbesserung zu RCS – sind:

Keine dateizentrierte Sichtweise

Entwicklungsstil: Kopieren-Modifizieren-Zusammenfügen d.h. simultanes Arbeiten mehrerer von einander unabhängiger Entwickler möglich

Informationssystem für alle (änderungsinteressierte) Entwickler des aktuellen Projekts

Netzwerkfähig, d.h. Historie ist auf andere Rechner übertragbar und durchgeführte Änderungen lassen sich zuordnen

2.4 SVN

Bis zum Jahr 2000 entwickelte sich CVS zum de-facto-Standard in der Open-Source-Gemeinde. Allerdings tauchten zunehmend Bugs und umständliches und fehlerhaftes Verhalten von CVS auf, was Anfang 2000 CollabNet Inc. dazu veranlaßte, Entwickler für eine Ablösung von CVS zu suchen. Ziel war es, eine komplett neue Versionsverwaltung, auf einigen guten Ideen von CVS aufbauend, zu entwickeln – allerdings ohne dessen Fehler und zusätzlich den Anforderungen der modernen Softwareentwicklung angepaßt. Im Februar 2000 begannen dann Karl Fogel und sein Freund Jim Blandy mit der Entwicklung von **Subversion** (SVN).

2.4.1 Vorteile

Die Hauptvorteile – als Verbesserung zu CVS – sind nach dem offiziellen Handbuch:

Verzeichnisversionierung: Nicht nur einzelne Dateien, sondern der komplette Verzeichnisbaum wird mitversioniert.

Echte Versionsgeschichte: Dateien und Verzeichnisse können nun auch kopiert und umbenannt werden, ohne deren Historie zu verlieren.

Atomare Commits: Ein Befehl wird nun entweder komplett oder gar nicht ausgeführt. So ist es unmöglich, daß nach einem Fehler in einer Datei alle bisherigen Veränderungen erhalten bleiben.

Versionierte Metadaten: Jede Datei oder jedes Verzeichnis kann beliebige, zusätzliche Informationen – in Form von Schlüssel- und Wertepaaren – enthalten. Diese Informationen werden mitversioniert.

Auswahl der Netzwerkschicht: Subversion kommt entweder als Apache-Erweiterung oder als leichtgewichtiger Standalone-Server.

Konsistentes Datenmanagement: Nun können nicht nur aus Textdateien sondern auch aus

Binärdaten Diffs erzeugt werden.

Effizientes Branching und Tagging: Der Aufwand steigt nun nicht mehr proportional zur Projektgröße an sondern ist immer gleich effizient.

API: Subversion kommt als eine Sammlung gemeinsamer C-Bibliotheken mit einer gut definierten API, auf die auch von externen Anwendungen zugegriffen werden kann.

Ein weiterer großer Vorteil, der zu dem großen Erfolg von Subversion geführt hat, ist die einfache Konvertierung eines alten CVS-Repositories in ein Subversion-Repository ohne Informationsverlust mit nur einem Befehl: cvs2svn (CVS to Subversion Repository Converter).

3 Quellen

Don Bolinger, Tan Bronson, Applying RCS and SCCS, O'REILLY, 1. September 1995, ISBN 1565921178

Gregor N. Purdy, CVS kurz & gut, O'Reilly Verlag, Köln, 1., korrigierter Nachdruck Dezember 2001, ISBN 3-89721-229-3

Derek Robert Price & Ximbiot, CVS - Concurrent Versions System, [CVS-Homepage](#)

Karl Fogel, Moshe Bar, Open Source-Projekte mit CVS, Verteilte Softwareentwicklung mit dem Concurrent Versions System, mitp-Verlag, Bonn, 2., erweiterte und überarbeitete Auflage 2002, ISBN 3-8266-0816-x

Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato, Version Control with Subversion, For Subversion 1.1, book compiled from Revision 1337, 2005, <http://svnbook.red-bean.com/>

CollabNet, Inc., Subversion Homepage, [Subversion-Homepage](#)

[Vergleich von 13 Versionsverwaltungen](#) (en.)

Dieser Artikel ist [GlossarWiki-konform](#).

In diesem Artikel sollten die Quellenangaben überarbeitet werden.
Bitte die Regeln der [GlossarWiki-Quellenformatierung](#) beachten.

Kategorien:

[Versionsverwaltung](#)

[Glossar](#)

Diese Seite wurde zuletzt am 13. September 2016 um 14:09 Uhr bearbeitet.

Inhalt verfügbar unter [CC BY-NC-SA 4.0](#), falls Dokument nach dem 5. 3. 2011 erstellt wurde, sonst [CC BY-SA DE 3.0](#).

