## **Was ist Subversion?**

Subversion ist ein freies/Open-Source Versionskontrollsystem. Das bedeutet, Subversion verwaltet Dateien und Verzeichnisse und die Änderungen an ihnen im Lauf der Zeit. Das erlaubt Ihnen, alte Versionen Ihrer Daten wiederherzustellen oder die Geschichte der Änderungen zu verfolgen. Unter diesem Blickwinkel denken viele Leute bei einem Versionskontrollsystem an eine Art „Zeitmaschine“.

Subversion kann netzwerkübergreifend arbeiten, was die Benutzung durch Menschen an verschiedenen Computern ermöglicht. Auf einer bestimmten Ebene fördert die Fähigkeit unterschiedlicher Personen dieselbe Menge an Daten bearbeiten und verwalten zu können die Zusammenarbeit. Ohne auf einen einzigen Kanal, über den alle Änderungen abgewickelt werden müssen, beschränkt zu sein, kann das Vorankommen beschleunigt werden. Und weil die Arbeit versioniert ist, braucht nicht befürchtet zu werden, dass die Qualität bei Verlust dieses Kanals geopfert wird – falls irgendeine falsche Änderung an den Daten gemacht wird, kann man sie einfach zurücknehmen.

Manche Versionskontrollsysteme sind auch Software-Konfigurationsmanagement-Systeme. Diese Systeme sind maßgeschneidert, um ganze Verzeichnisbäume mit Quellcode zu verwalten und verfügen über viele Merkmale, die spezifisch für Software-Entwicklung sind – etwa das Verstehen von Programmiersprachen oder das Bereitstellen von Werkzeugen zum Bauen von Software. Jedoch gehört Subversion nicht zu diesen Systemen. Es ist ein allgemeines System, das verwendet werden kann, um alle möglichen Sammlungen von Dateien zu verwalten. Für Sie mag es sich dabei um Quellcode handeln – für andere mag es dabei um alles von Einkaufslisten bis zu digitalen Videomischungen und weit darüber hinaus gehen.

### Ist Subversion das richtige Werkzeug?

Falls Sie ein Anwender oder Systemadministrator sind und den Einsatz von Subversion erwägen, sollte die erste Frage, die Sie sich stellen, sein: "Ist es das richtige Werkzeug für die Aufgabe?" Subversion ist ein fantastischer Hammer, achten Sie jedoch darauf, dass Sie nicht jedes Problem als einen Nagel sehen.

Falls Sie alte Datei- und Verzeichnisversionen aufbewahren, sie eventuell wiedererwecken müssen, oder Protokolle darüber auswerten möchten, wie sie sich im Lauf der Zeit geändert haben, ist Subversion das genau passende Werkzeug für Sie. Subversion ist auch geeignet, wenn Sie mit mehreren Leuten gemeinsam (üblicherweise über das Netz) an Dokumenten arbeiten und verfolgen müssen, wer welche Änderung gemacht hat. Deshalb wird Subversion so oft in Softwareentwicklungsumgebungen eingesetzt – die Arbeit in einem Entwicklerteam ist von Natur aus eine soziale Tätigkeit und Subversion vereinfacht die Zusammenarbeit mit anderen Programmierern. Natürlich ist die Benutzung von Subversion nicht umsonst zu bekommen: es kostet administrativen Aufwand. Sie müssen ein Daten-Projektarchiv verwalten, das die Informationen und ihre gesamte Geschichte speichert, und Sie müssen sich gewissenhaft um Sicherheitskopien kümmern. Wenn Sie täglich mit den Daten arbeiten, werden Sie sie nicht auf die gleiche Art kopieren, verschieben, umbenennen oder löschen können wie gewohnt. Stattdessen müssen Sie dafür Subversion verwenden.

Unter der Annahme, dass Ihnen die zusätzlichen Arbeitsabläufe nichts ausmachen, sollten Sie trotzdem sicher sein, dass Sie Subversion nicht für die Lösung eines Problems verwenden, das andere Werkzeuge besser lösen könnten. Zum Beispiel wird Subversion, weil es die Daten an alle Beteiligten verteilt, als generisches Verteilsystem missbraucht. Manchmal wird Subversion zum Verteilen von umfangreichen Bildersammlungen, digitaler Musik oder Softwarepaketen verwendet. Das Problem damit ist, dass sich diese Art Daten für gewöhnlich überhaupt nicht verändert. Die Sammlung selber wächst stetig, jedoch werden die einzelnen Dateien der Sammlung nicht verändert. In diesem Fall ist die Benutzung von Subversion zu viel des Guten.[[2](http://svnbook.red-bean.com/de/1.5/svn.intro.whatis.html#ftn.idp5828912)] Es gibt einfachere Werkzeuge, die hervorragend Daten replizieren, ohne dabei Änderungen mitzuverfolgen, etwa ***rsync*** oder ***unison***.

### Die Geschichte von Subversion

Anfang 2000 begann CollabNet, Inc. ([http://www.collab.net](http://www.collab.net/)) Entwickler zu suchen, die einen Ersatz für CVS schreiben sollten. CollabNet bietet eine Software-Suite namens CollabNet Enterprise Edition (CEE) für die Zusammenarbeit an, die auch eine Komponente für Versionskontrolle beinhaltet. Obwohl CEE ursprünglich CVS als Versionskontrollsystem verwendete, waren die Einschränkungen von CVS von Anfang an offensichtlich, und CollabNet war sich bewusst, dass letztendlich etwas Besseres gefunden werden musste. Unglücklicherweise war CVS der de-facto Standard in der Open-Source-Welt geworden, hauptsächlich deshalb, weil es nichts Besseres gab, zumindest nicht unter einer freien Lizenz. Also beschloss CollabNet, ein vollständig neues Versionskontrollsystem zu schreiben, welches die grundlegenden Ideen von CVS beibehalten, jedoch die Fehler und Fehlentwicklungen vermeiden sollte.

Im Februar 2000 nahmen sie Verbindung mit Karl Fogel auf, dem Autor von Open Source Development with CVS (Coriolis, 1999), und fragten ihn, ob er an diesem neuen Projekt mitarbeiten wolle. Zufälligerweise besprach Karl bereits einen Entwurf für ein neues Versionskontrollsystem mit seinem Freund Jim Blandy. Im Jahr 1995 gründeten die beiden Cyclic Software, eine CVS-Beratungsfirma, und sie benutzten, obwohl sie die Firma später verkauften, bei ihrer täglichen Arbeit immer noch CVS. Ihre Enttäuschung über CVS veranlasste Jim, sorgfältig über bessere Möglichkeiten zur Verwaltung versionierter Daten nachzudenken. Er hatte sich nicht nur bereits den Namen „Subversion“ ausgedacht, sondern auch den grundsätzlichen Entwurf der Subversion-Datenablage. Als CollabNet rief, stimmte Karl sofort der Mitarbeit am Projekt zu, und Karl gelang es, dass sein Arbeitgeber Red Hat Software ihn praktisch auf unbestimmte Zeit dem Projekt spendete. CollabNet stellte Karl und Ben Collins-Sussman ein und der detaillierte Entwurfsprozess begann im Mai. Dank einiger Stupser von Brian Behrendorf und Jason Robbins von CollabNet sowie Greg Stein (zu dieser Zeit als unabhängiger Entwickler aktiv im der WebDAV/DeltaV Spezifikationsprozess), zog Subversion schnell eine Gemeinde aktiver Entwickler an. Es stellte sich heraus, dass viele Leute dieselben enttäuschenden Erfahrungen mit CVS gemacht hatten und nun die Gelegenheit begrüßten, etwas daran zu ändern.

Das ursprüngliche Designteam einigte sich auf einige einfache Ziele. Sie wollten kein Neuland in Versionskontrollmethodik betreten, sondern einfach CVS reparieren. Sie beschlossen, dass Subversion dieselben Merkmale und dasselbe Entwicklungsmodell wie CVS haben solle, wobei die Fehler von CVS aber nicht noch einmal gemacht werden sollten. Und obwohl es nicht als ein hundertprozentiger Ersatz für CVS gedacht war, sollte es dennoch ähnlich genug sein, so dass ein leichter Wechsel für einen CVS-Anwender möglich wäre.

Nach vierzehn Monaten Programmierung wurde Subversion am 31. August 2001 „selbstbewirtend“, d.h., die Subversion-Entwickler hörten auf, CVS für den Quellcode von Subversion zu verwenden und benutzten stattdessen Subversion.

Obwohl CollabNet das Projekt startete und immer noch einen großen Batzen der Arbeit finanziert (sie zahlen die Gehälter einiger Vollzeit-Subversion-Entwickler), läuft Subversion wie die meisten Open-Source-Projekte, geführt von einer Anzahl lockerer, transparenter Regeln, die die Meritokratie fördern. Die Urheberrechtslizenzen von CollabNet sind einvernehmlich mit den Debian Free Software Guidelines. Mit anderen Worten: Jeder darf Subversion nach Belieben herunterladen, ändern und weitergeben; es bedarf hierzu keinerlei Zustimmung durch CollabNet oder sonst jemanden.

### Die Architektur von Subversion

[Abbildung 1, „Die Architektur von Subversion“](http://svnbook.red-bean.com/de/1.5/svn.intro.whatis.html#svn.intro.architecture.dia-1) illustriert einen „kilometerhohen“ Blick auf das Design von Subversion.

**Abbildung 1. Die Architektur von Subversion**



An einem Ende ist das Projektarchiv von Subversion, das die gesamten versionierten Daten enthält. Am anderen Ende ist das Subversion-Client-Programm, das die lokale Spiegelung von Teilen dieser versionierten Daten verwaltet („Arbeitskopien“ genannt). Zwischen den entgegengesetzten Enden befinden sich mehrere Wege über verschiedene Projektarchiv-Zugriffsschichten. Einige dieser Pfade gehen über Computernetzwerke und über Netzwerkserver, die dann auf das Projektarchiv zugreifen. Andere lassen das Netz links liegen und greifen direkt auf das Projektarchiv zu.