

Versionsverwaltung

Inhaltsverzeichnis:

- Versionsverwaltung
 - Copy-Modify-Merge
 - Subversion SVN
 - GIT
 - Mercurial
 - Zusammenfassung der Unterschiede
 - Lock-Modify-Write
 - Dropbox
 - Google Drive

Versionsverwaltung

In der Softwareentwicklung ist es unumgänglich das an einem Projekt mehrere Programmierer gleichzeitig arbeiten können. Es wird nicht nur gleichzeitig am selben Code geschrieben sondern möglicherweise auch Features parallel d.h. unabhängig voneinander implementiert. Dieses sogenannte „branching“ ist ein elementares Merkmal einer Versionsverwaltung für Softwareprojekte. Daneben gibt es Möglichkeiten die Codeänderungen zu kommentieren, alte Versionen zu archivieren und im Notfall auch zu diesen zurück springen zu können. Die Folgende Liste soll die meistverwendeten Lösungen auf dem Markt vergleichen und die Vor- und Nachteile jedes Systems zeigen.

Neben klassischen Versionskontrolltools werden auch Cloudspeicher wie Dropbox aufgeführt, da sie mittlerweile auch einige Features wie Archivierung und Rollback aufgenommen haben. Sie eignen sich nicht sehr gut für Programmcode, werden allerdings meist parallel eingesetzt um Binary-Formate z.B. zwischen Artists und Codern auszutauschen.

Copy-Modify-Merge

Moderne Versionsverwaltung die für Programmcode entworfen wurde. Es kann ohne Lock an einer Datei gleichzeitig gearbeitet werden.

Subversion SVN

Vorteile: Open Source unter der Apache Lizenz. Zentrales System bei der die komplette Versionsgeschichte nur auf dem Server gespeichert wird (dies spart dem Client Speicherplatz). Unterstützt branching, merging, tagging und ein erwachsenes Benutzer- und Rechtssystem.

Nachteile:

Es muss ein eigener Server betrieben werden. Ein Zentrales System setzt immer eine Netzanbindung des Client zum Server voraus. Server muss gewartet, überwacht und regelmäßig gesichert werden um einen Datenverlust zu vermeiden. Binaries können nicht in diffs gespeichert werden d.h. sie werden nicht unter Versionsverwaltung gestellt. OS: Clientsoftware für Linux, Mac und Windows erhältlich

(Lizenz)Kosten: **0 USD**

<http://subversion.apache.org>

GIT

Vorteile: Open Source unter GNU GPL. Dezentrales System bei dem die komplette Versionsgeschichte auf dem Client gespeichert wird. Dies ermöglicht das offline Arbeiten und benötigt später lediglich eine Synchronisation mit den anderen Clients. Möglichkeiten zum branchen, mergen und taggen sind vorhanden. Ebenso Benutzer und Rechteverwaltung.

Nachteile:

Noch relativ junges Projekt welches von den Erstellern noch nicht als „fertig“ angesehen wird. Dezentrales System bedeutet für den User einen höheren Speicherbedarf auf seiner lokalen Maschine. Kann ebenso schlecht mit Binaries umgehen. Seine hohe Konfigurierbarkeit macht es schwer zu managen. Daher versagt es für Kleintteams und Soloentwickler, da es zu viel Overhead birgt

OS: Clientsoftware für Linux, Mac und Windows erhältlich (Lizenz)Kosten: **0 USD**

<http://git-scm.com>

Mercurial

Vorteile:

Sehr ähnlich zu GIT. Open Source unter GNU GPL. Dezentrales System mit Versionierung, Benutzer- und Rechteverwaltung. Lässt sich einfacher verwalten ist daher besser für kleine Teams geeignet.

Nachteile:

Selbe Nachteile wie GIT. Mercurial ist im Allgemeinen etwas langsamer wenn es für größere Projekte verwendet wird. OS: Clientsoftware für Linux, Mac und Windows erhältlich

(Lizenz)Kosten: **0 USD**

<http://mercurial.selenic.com>

Zusammenfassung der Unterschiede

Ein Blogger fasste es so zusammen:

„[...]Mercurial is Denzel Washington [...]Git is Wesley Snipes. [...]People admire Denzel Washington. But Wesley Snipes is simply more interesting. [...]People also ask me why I don't write more about Subversion. That's easy too: Subversion is Morgan Freeman“^[1]

Bei der Wahl zwischen diesen drei Versionskontrollsystemen kommt es also am meisten auf den persönlichen Geschmack an. Wenngleich SVN langsam an Bedeutung verliert und der Trend in Richtung dezentralen Systemen wie Mercurial und GIT geht, ist SVN noch das am weitesten verbreitete.

Lock-Modify-Write

Die nächsten beiden Cloudspeicherdienste erlauben es in der Desktopumgebung mit ihrer Clientsoftware Daten einfach zu sichern und auszutauschen aber verbieten das gleichzeitige Bearbeiten.

Dropbox

Vorteile:

Einfache Handhabung. Nahtlose Integration in ein vorhandenes Dateisystem. Lädt im Hintergrund Daten hoch und runter. Kann hervorragend mit Binaries umgehen. Der User muss sich nicht mit update, commit und merges beschäftigen. Einfache Implementation einer Rollback-Funktion.

Nachteile:

Rollback leider nur über ein Webinterface erreichbar und nicht unendlich lang zurückgehbar. Kein Open Source und Monatliche Kosten wenn man mehr Speicherplatz braucht. Die Daten sind nicht mehr in der eigenen Hand und werden auf fremden Servern gespeichert. Nur simple Benutzerverwaltung.

OS: Clientsoftware für Linux, Mac, Windows, iOS, Android usw erhältlich

Kosten: bis 2gb(Freundschaftswerbung bis 18gb) kostenlos danach ab **120 USD/100gb/Jahr**

<http://www.dropbox.com>

Google Drive

Vorteile:

Bietet ähnliche Vorteile wie Dropbox mit dem Desktop Client. Zusätzlich kann über ein Webinterface simultan an simplen Dokumenten und Spreadsheets gearbeitet werden. Bessere als Dropbox aber immer noch simple Rechteverwaltung.

Nachteile:

Bietet Versionierung und Archivierung nur für online Spreadsheets und Dokumente. Sollten Client-Desktop-Daten gelöscht oder zerstört werden sind sie verloren. Gmail Account nötig um Google Drive nutzen zu können. Daten liegen wieder auf fremden Servern. Ebenfalls fallen Kosten an.

OS: Clientsoftware für Mac, Windows, iOS, Android erhältlich

Kosten: bis 5gb kostenlos danach ab **59,88 USD/100gb/Jahr**

<http://drive.google.com>

	Benutzer- & Rechteverwaltung	Entwicklungszweige	Binaries	OS Integration/ Multiplattform	Datenhoheit	Archivierung
Dro pb ox	-	--	++	++	--	o
Go ogl eD rive	o	--	++	o	--	-
Su bve rsi on SVN	++	+	o	+	+	++
GIT	++	++	-	+	++	+
Me rcu rial	++	++	-	+	++	+

++ sehr gut	+ gut	o mittel
-- sehr schlecht	- schlecht	