

svn vs git

Sowohl svn als auch git sind sehr etablierte Werkzeuge. Es gibt selbstverständlich auch eine ganze Reihe weiterer VCS, die ebenfalls erfolgreich in verschiedensten Projekten verwendet werden.

Hier sollen subversion und git verglichen werden.

Vorher noch eine Anmerkung: Machen Sie keine Religion aus der Tool-Auswahl, sondern prüfen Sie rational, welche Anforderungen Sie haben und welches Tool diese Anforderungen besser erfüllt.



Was haben die beiden Tools gemeinsam?

- Beide sind Open Source, erzeugen also keine Lizenzkosten.
- Beide sind sehr gut etabliert.
- Beide sind sehr ausgereift und zuverlässig – die Verwendung in extrem vielen Projekten bestätigt das.
- Für beide gibt es etablierte Hosting Plattformen.
- Für beide gibt es umfangreiche Dokumentation, Tutorials und Lehrbücher.
- Beide lassen sich gut in IDEs integrieren.



svn vs git - Unterschiede

- **Bedienbarkeit:**

subversion ist (für die meisten Anwender) leichter zu erlernen. Das hängt natürlich auch damit zusammen, dass git mehr Szenarien der Anwendung ermöglicht.

- **Nutzung:**

Bei der Verwendung von subversion ist die Vorgehensweise ziemlich klar und unterscheidet sich von Projekt zu Projekt nicht so arg. Bei git sollten Sie auf jeden Fall im Team den Entwicklungsworkflow festlegen und Sie haben hier mehr Optionen.

- **Netzwerk**

Mit git sind die meisten Interaktionen lokal. Bei Subversion nicht. Allerdings sollte man sich nicht verleiten lassen, 2 Wochen lokal zu arbeiten und dann die Ergebnisse verschiedener Entwickler zusammenzuführen. Daher ist zu prüfen, ob dieses Argument in einem speziellen Projekt Relevanz hat.



svn vs git - Unterschiede

- **Rechte**

subversion ermöglicht es, Zugriffsrechte zu setzen, git nicht. Wenn Sie aber git im Rahmen einer Plattform wie z.B. gitlab nutzen, dann haben Sie auch die Möglichkeit einer Rechteverwaltung.

- **Branching**

bei subversion ist branching ein zentrales Konzept, bei git ein lokales. Bei git ist branching sehr leichtgewichtig und einfach. Es ist die Intention bei git, branching in der täglichen Arbeit, z.B. für feature branches, zu nutzen.

- **Nutzen von branches**

Der Wechsel zwischen verschiedenen branches bei der Entwicklungsarbeit ist in git extrem einfach. Bei subversion haben Sie dafür unterschiedliche Arbeitsbereiche in der Entwicklungsumgebung.



git vs svn

- **Merges**

git hat einen sehr effektiven Merge Mechanismus, der darauf beruht, dass sich ein commit seine Eltern merkt. Subversion hat diesen Mechanismus nicht.

- **Binäre Dateien**

Bei subversion arbeiten Sie immer nur mit einer Revision, bei git haben Sie lokal ein komplettes repository und Sie synchronisieren immer das gesamte Repository. Wenn Sie viele Versionen von binären Dateien verwalten, wächst Ihr git repository stark an und die Synchronisation mit anderen Repositories wird langsam.

Hier ist das Client-Server-Modell von svn im Vorteil. Zudem verwendet svn intern einen effektiven delta Mechanismus, der auch für binäre Files anwendbar ist. Dadurch wächst der Speicherbedarf eines svn repositories bei Verwendung von binären Files nicht so stark an wie bei git.



git vs svn

- **Sicherheit gegen Datenverlust:**

Es wird immer als Vorteil ins Feld geführt, dass die Daten in git durch die redundante Verteilung sicherer gegen Verlust geschützt sind. Wenn Sie professionell arbeiten, müssen Sie jedoch bei beiden Tools jedes zentrale Repository durch regelmäßige Backups sichern.

- **Verbreitung**

git scheint subversion immer mehr den Rang abzulaufen und langfristig das stärker verbreitete Tool zu werden.

- **Allgemeine Nutzung**

git bietet durch stash und staging Möglichkeiten, die Sie mit subversion nicht haben.

