# Versionskontrolle – Vorteile und Nachteile

## Ablauf

### Zentrale Versionierung

Es gibt einen zentralen Server, auf dem alle Versionen Verwaltet werden. Clients können die Dateien von dem Server Auschecken. Das heißt, dass die Versionsgeschichte nur auf dem Server verwaltet wird.

Ein Beispiel von Zentraler Versionierung ist CVS (Concurrent Versions System). Es wird hauptsächlich im Zusammenhang mit Software – Quelltext verwendet.



### Dezentrale Versionierung

Hier gibt es auch einen Server, aber wenn Clients an den Versionen arbeiten wollen, müssen sie sich eine komplette Kopie des Repositorys herunterladen. (Github, kein Datum)

Ein Beispiel für Dezentrale Versionierung ist Git. Git wird vor allem für die Entwicklung für Open Source Projekten eingesetzt.



## Vorteile

### Zentrale Versionierung

* Jeder weiß mehr oder weniger, wer gerade an was Arbeitet.
* Administratoren haben die Möglichkeit, detaillierte Rechte zu vergeben.

### Dezentrale Versionierung

* Jeder Client hat ein eigenes Backup des Projektes lokal gespeichert.
* Mehr Clients können gleichzeitig an einer Datei arbeiten.

## Nachteile

### Zentrale Versionierung

* Wenn der Server ausfällt ist im schlimmsten Fall alles weg.
* Es kann nur ein Client an einer Datei arbeiten.

### Dezentrale Versionierung

* Wenn das Projekt größer ist, braucht man mehr Lokalen Speicher und bei Versionierung kann das schnell viel werden.
* Clients könnten Redundant an einem Projekt arbeiten, wenn z.B. zwei an derselben Datei arbeiten.

# Literaturverzeichnis

Github. (kein Datum). *Git-SCM*. Von Git-SCM: https://git-scm.com/book/de/v2/Los-geht%E2%80%99s-Was-ist-Versionsverwaltung%3F abgerufen

Rawson, R. (2018). *Time Doctor*. Von https://biz30.timedoctor.com/git-mecurial-and-cvs-comparison-of-svn-software/ abgerufen