

# Computergrundlagen

# Versionskontrollsysteme

**Axel Arnold**

Institut für Computerphysik  
Universität Stuttgart

Wintersemester 2011/12

- Speichern platzsparend ältere Versionen von Dateien
- Speichern Informationen zu den Versionen
- Erlauben verteilten Zugriff
- Erleichtern gleichzeitiges Arbeiten an mehreren Versionen
- Versuchen, gleichzeitige Änderungen an Dateien zusammenzuführen
- Server-basiert oder verteilt

## Anwendungsgebiete

- Softwareentwicklung
- Gemeinsame Arbeit an ( $\text{\LaTeX}$ -)Dokumenten
- Regelmäßige Sicherung eigener Arbeiten (insbesondere Bachelor-, Masterarbeit!)

- CVS (<http://www.nongnu.org/cvs>)
  - Server-basiert (auch lokal über Dateien)
  - Älter, aber einfach zu bedienen
- Subversion (<http://subversion.tigris.org>)
  - Server-basiert (auch lokal über Dateien)
  - Ähnlich CVS, unterstützt mehr Funktionen
- GIT (<http://git-scm.com>)
  - Verteiltes Versionskontrollsystem
  - Erlaubt auch lokal in einem Verzeichnis mehrere Entwicklungszweige
  - VCS der Linux-Kernel-Entwicklung
- Mercurial (<http://mercurial.selenic.com>)
  - Verteiltes Versionskontrollsystem wie GIT
  - Wird z.B. auf Google Code angeboten (neben Subversion)

- Anlegen eines **Repositories** (Datenbank mit Historie)
- Eventuell Anlegen einer Kopie des Verzeichnisses (**Checkout** einer **Arbeitskopie**)
- Ändern von Dateien im Verzeichnis
- Holen von Änderungen aus anderen Quellen (**Update**)
- Diese werden mit den eigenen Änderungen verschmolzen (**Merge**)
- Dabei kann es zu nicht automatisch auflösbaren Überschneidungen kommen (**Conflict**)
- Diese müssen manuell behoben werden
- Übergabe der Änderungen an das Repository (**Commit**)
- Eventuell Erstellen von Entwicklungszweigen (**Branches**)



- Anlegen eines neuen Repositories: **git init *Repo-Name***
  - Dies legt ein Verzeichnis *Repo-Name* an
  - Darin können Dateien und Verzeichnis wie gewohnt angelegt werden (Ausnahme: das `.git`-Verzeichnis nicht anfassen!)
- **git status**: Status der Dateien im Verzeichnis
  - Dateien sind erst einmal nicht verfolgt (**untracked**), also nicht versioniert
  - Liste aller geänderten versionierten Dateien
- Hinzufügen zur Versionskontrolle durch **git add *Datei ...***
- **git commit -a** speichert alle Änderungen als neue Version
  - Erwartet einen Kommentar, damit man später weiss, was man committet hat
  - Man kann auch einzelne Dateien committen
- **git log** listet alle Commits und vor allem ihre IDs
- **git revert *ID*** macht den Commit mit ID *ID* rückgängig.  
Nur der Anfang von *ID* ist nötig, solange das eindeutig ist