

# Intro: Versionskontrolle in der Softwareentwicklung

---

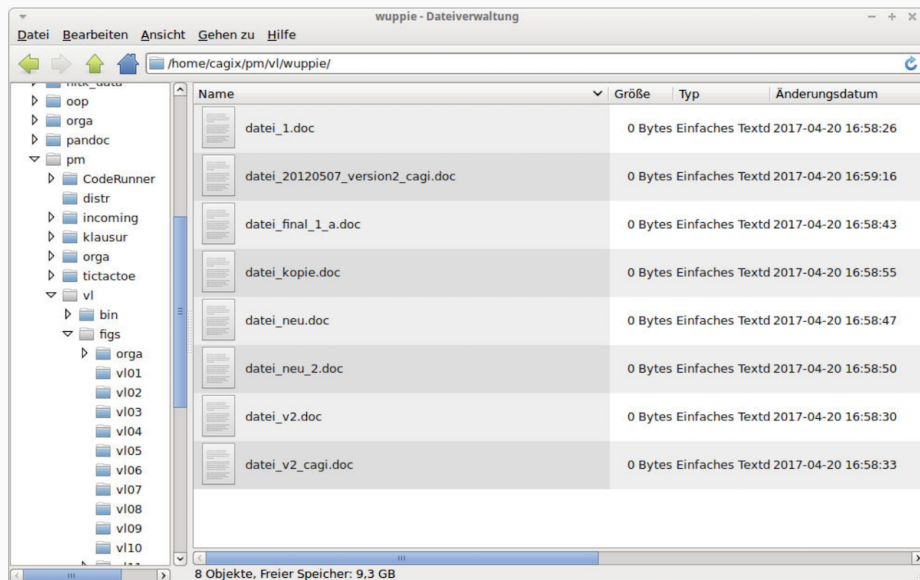
Carsten Gips (FH Bielefeld)

Unless otherwise noted, this work is licensed under CC BY-SA 4.0.

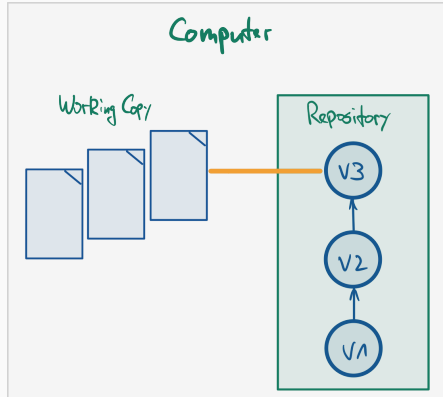
# Typische Probleme bei SW-Entwicklung

- Was hat wer wann (und wo) geändert? Und warum?
- Ich brauche den Stand von gestern/letzter Woche/...
- Ich will schnell mal eine neue Idee ausprobieren ...
- Ich arbeite an mehreren Rechnern (Synchronisation)
- Wir müssen gemeinsam an der gleichen Codebasis arbeiten.
- Wir arbeiten am Release v42, aber Kunde braucht schnell einen Fix für v40

# Folgen SW-Entwicklung ohne Versionsverwaltung

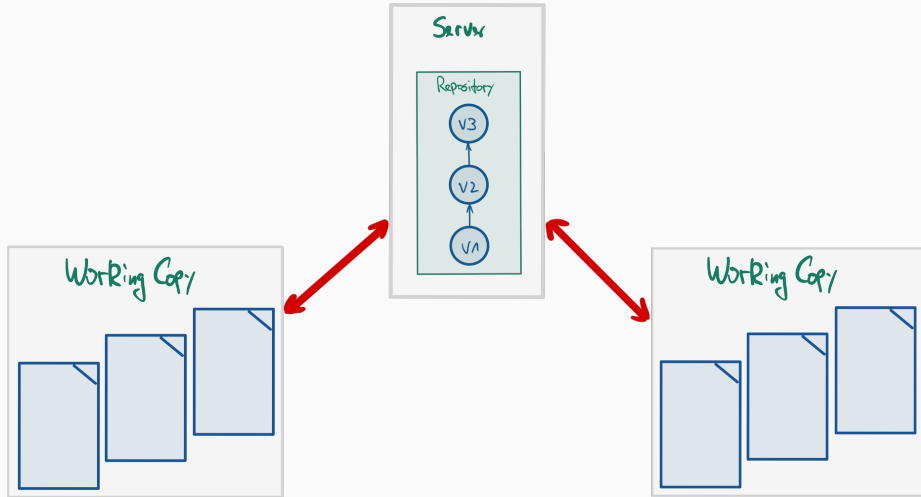


# Prinzip Versionsverwaltung

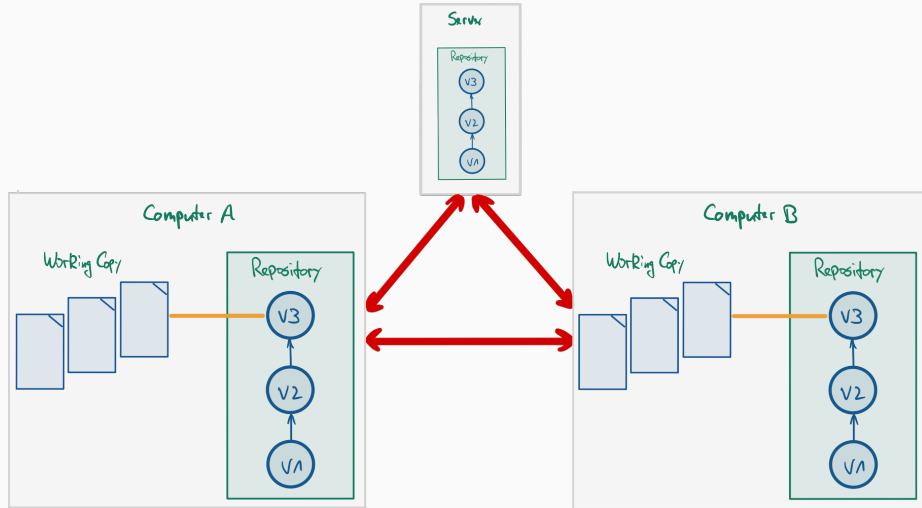


- **Repository:** Datenbank mit verschiedenen Versionsständen, Kommentaren, Tags etc.
- **Workingcopy:** Arbeitskopie eines bestimmten Versionsstandes

## Varianten: Zentrale Versionsverwaltung (Beispiel SVN)



# Varianten: Verteilte Versionsverwaltung (Beispiel Git)



# Versionsverwaltung mit Git: Typische Arbeitsschritte

1. Repository anlegen (oder clonen)
2. Dateien neu erstellen (und löschen, umbenennen, verschieben)
3. Änderungen einpflegen (“committen”)
4. Änderungen und Logs betrachten
5. Änderungen rückgängig machen
6. Projektstand markieren (“taggen”)
7. Entwicklungszweige anlegen (“branchen”)
8. Entwicklungszweige zusammenführen (“mergen”)
9. Änderungen verteilen (verteiltes Arbeiten, Workflows)

# (Globale) Konfiguration

## Minimum:

- `git config --global user.name <name>`
- `git config --global user.email <email>`

## Aliase:

- `git config --global alias.ci commit`
- `git config --global alias.co checkout`
- `git config --global alias.br branch`
- `git config --global alias.st status`
- `git config --global alias.ll 'log --all --graph --decorate --oneline'`



# Neues Repo anlegen

- `git init`

=> Erzeugt neues Repository im akt. Verzeichnis

- `git clone <url>`

=> Erzeugt (verlinkte) Kopie

- Git: Versionsmanagement mit dezentralen Repositories
- Anlegen eines lokalen Repos mit `git init`
- Clonen eines existierenden Repos mit `git clone <url>`

# LICENSE



Unless otherwise noted, this work is licensed under CC BY-SA 4.0.