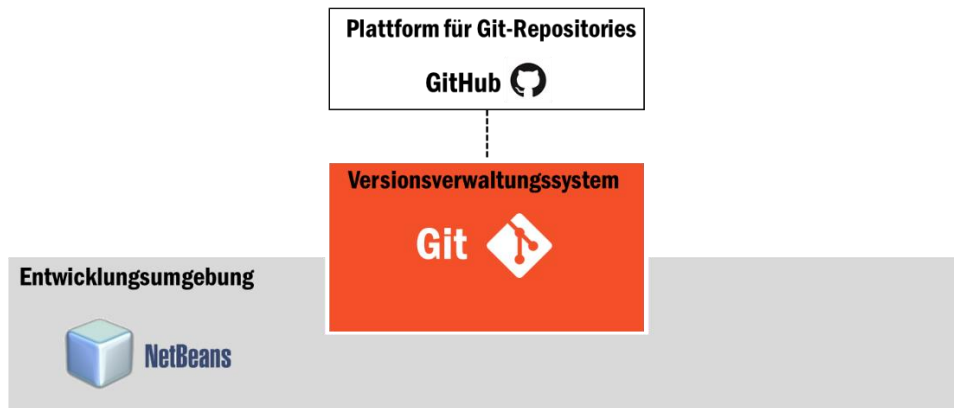


# Einführung in Git & Github

18. Januar 2018

## 1. Überblick



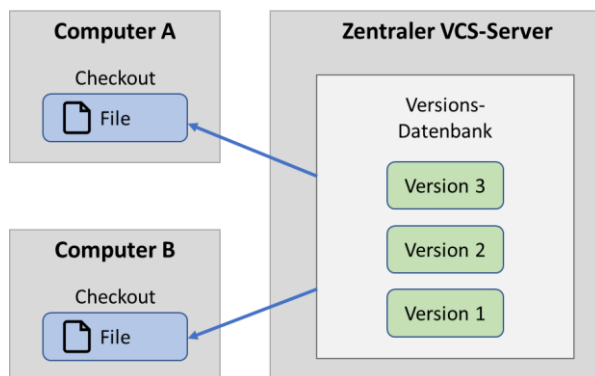
## 2. Versionsverwaltungssysteme

### Funktionen:

- Archivierung von Versionen und Historie der Änderungen
- „Zeitmaschine“: Bei Fehlern kann auf ältere Versionen zurückgegriffen werden
- Zusammenarbeit von mehreren Personen an einem Projekt ermöglichen

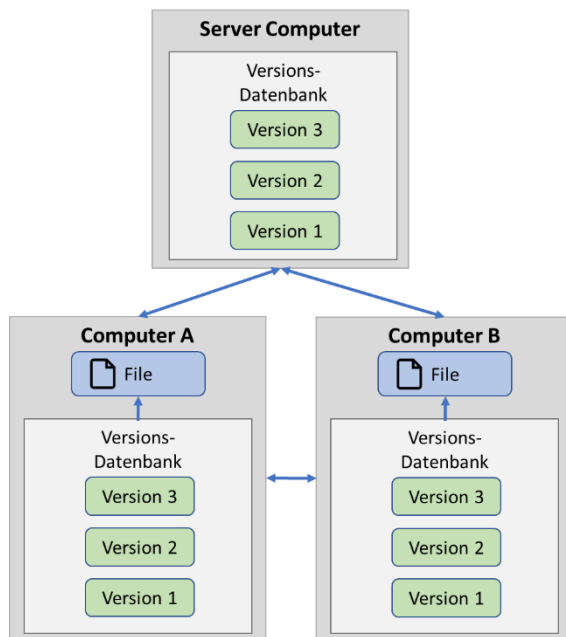
### Aufbau:

- Zentralisiert (z.B. Subversion)



- Auf dem zentralen Server liegen die "offiziellen Versionen" des Codes
- Server speichert den einzigen Versionsverlauf
- "Checkout" auf lokalen Computer  
→ Modifikationen (nicht versioniert)
- "Checkin" zurück auf den Server  
→ Neue Version

- Denzentral (z.B Git)

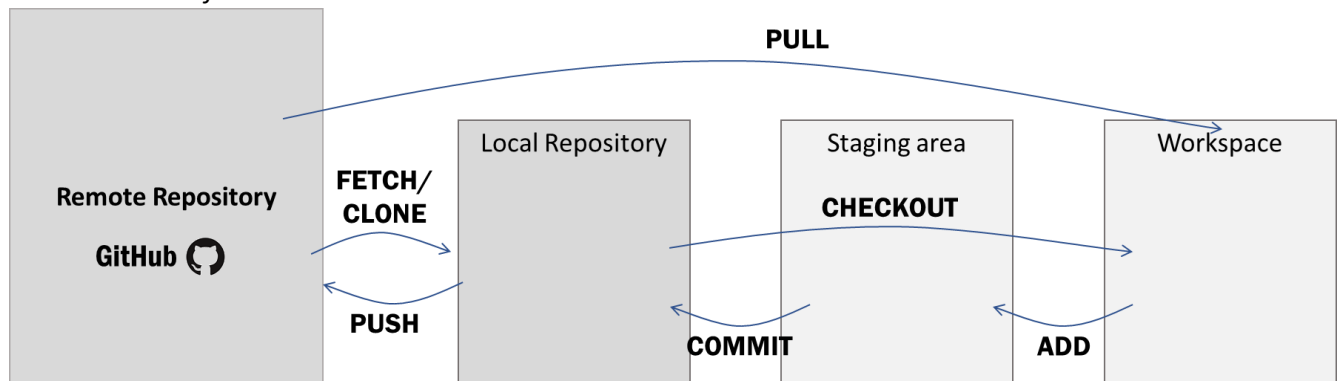


- Kein "Checkout", sondern ein "Klon" vom gesamten Repository wird vom Server kopiert
- "Pull" - Änderungen werden vom Original gezogen
- Lokal ist ein Checkin/Checkout von der LOKALEN Datei möglich
- Lokaler Versionsverlauf wird gespeichert
- "Push" - Änderungen werden zum Server übertragen

### 3. Funktionsweise von Git

Git speichert "Snapshots" vom gesamten Projekt  
 → Jede Version hat den gesamten Code gespeichert

Übersicht der Befehle:



Benefits von Git:

- Offline Versionskontrolle: Jedes Local Repository ist ein vollständiger Klon des Remote Repositories
- Unterstützung von nicht-linearen Entwicklungsprojekten (Merging von Branches)
- Sehr hohe Verbreitung
- Sehr schnell (durch den dezentralen Aufbau)

### 4. GitHub-Plattform

- Hosting von Git-Repositories (open-source)
- Kollaboration auf Basis des Codes
- Steuerung von Teams
- Syntax-Highlighting für zahlreiche Programmiersprachen