



Einführung Git

VERSIONSVERWALTUNG LEICHT
GEMACHT

Was ist Git?

- Ein verteiltes **Versionskontrollsystem** (VCS)
- Speichert den Verlauf von Änderungen in Projekten
- Ideal für die Softwareentwicklung und Zusammenarbeit

Warum Git?

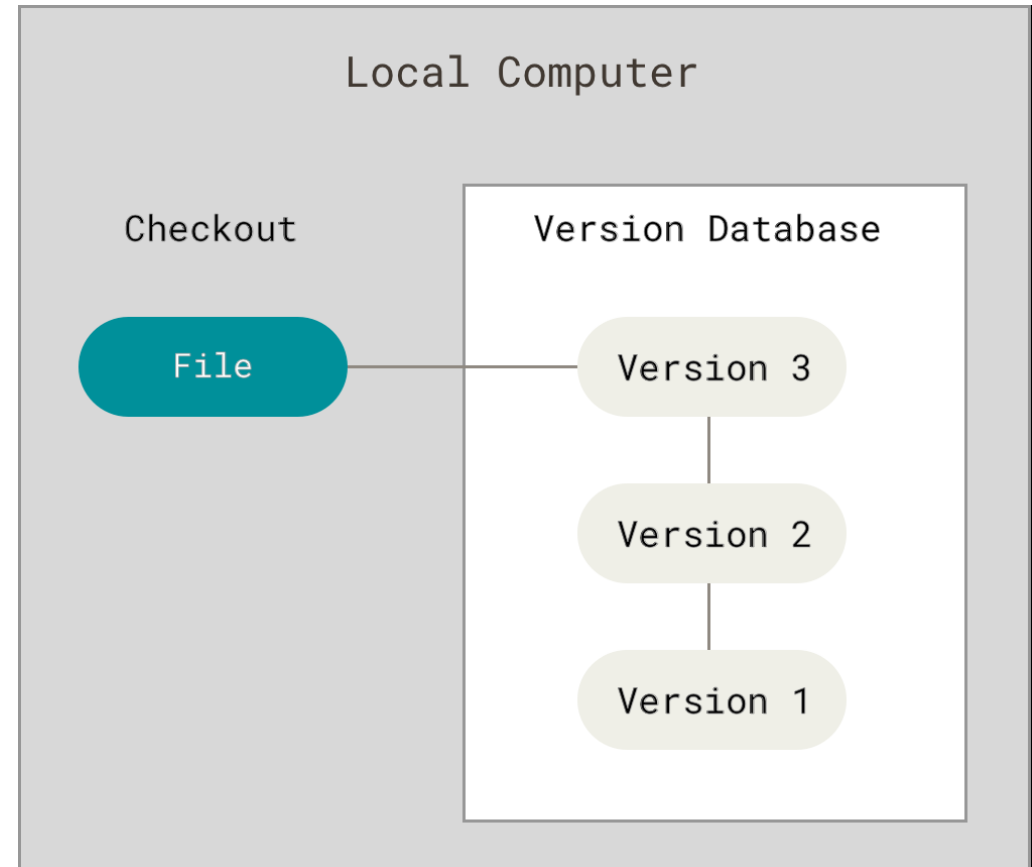
- Verfolgt jede Änderung im Code
- Unterstützt paralleles Arbeiten im Team
- Bietet eine Sicherheitskopie des gesamten Projektverlaufs

Was bedeutet Versionskontrolle?

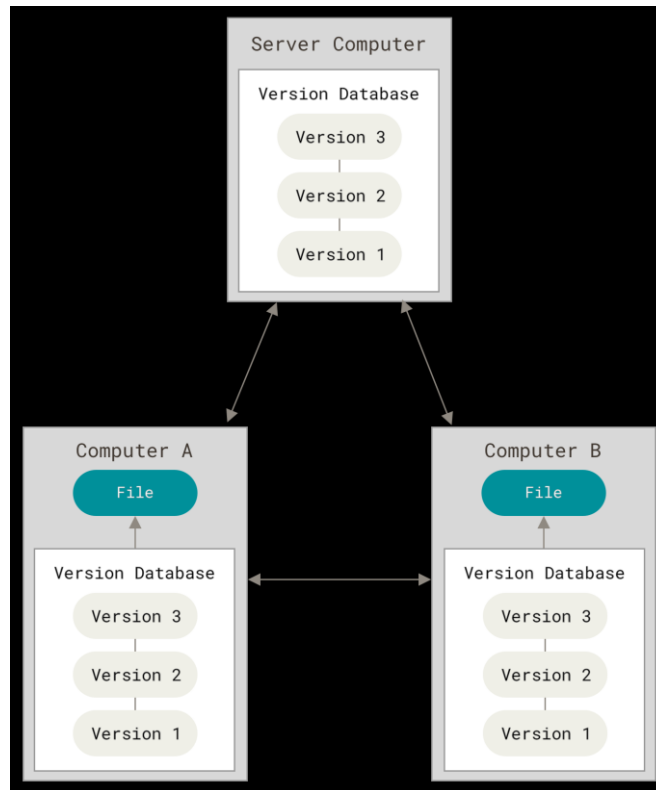
- **Verwaltung** und **Nachverfolgung** von Änderungen in Dateien und Projekten

- **Vorteile:**

- **Historie:** Speichert alle Änderungen, erlaubt das Zurücksetzen auf frühere Versionen.
- **Zusammenarbeit:** Mehrere Personen können gleichzeitig an einem Projekt arbeiten.
- **Sicherheit:** Jede Version ist gesichert, keine Datenverluste bei Fehlern.



Was bedeutet Versionskontrolle?



- Verteiltes System:

Jeder Nutzer hat eine vollständige Kopie des Projekts

Geschichte

- Entwickelt von: Linus Torvalds (2005), der auch den Linux-Kernel ins Leben rief
- Warum wurde Git entwickelt?
 - Torvalds und das Linux-Entwicklerteam benötigten ein robustes und leistungsfähiges Versionskontrollsystem
 - Die bestehenden Systeme waren nicht flexibel und leistungsstark genug für ein globales Open-Source-Projekt wie den Linux-Kernel

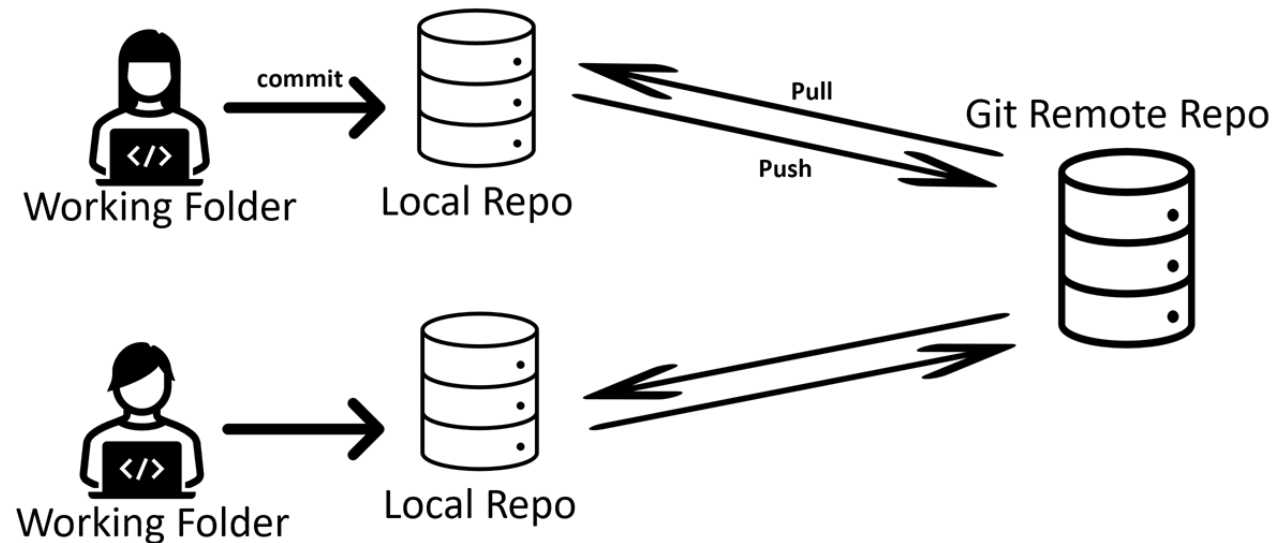
Verbreitung: Git ist heute eines der am meisten genutzten Versionskontrollsysteme und wird in zahlreichen Projekten weltweit eingesetzt

Grundkonzepte

Repository

Speichert alle Dateien und den gesamten Verlauf eines Projekts

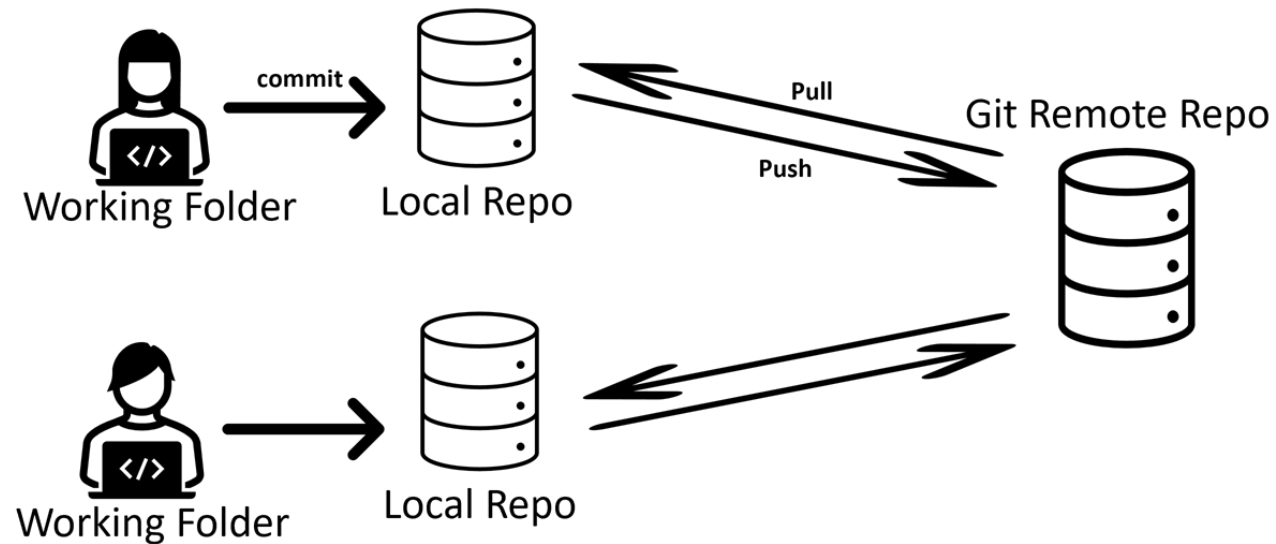
Remote-Repository: Ein Online-Repo für Backup und Zusammenarbeit, z. B. auf GitHub



Commit

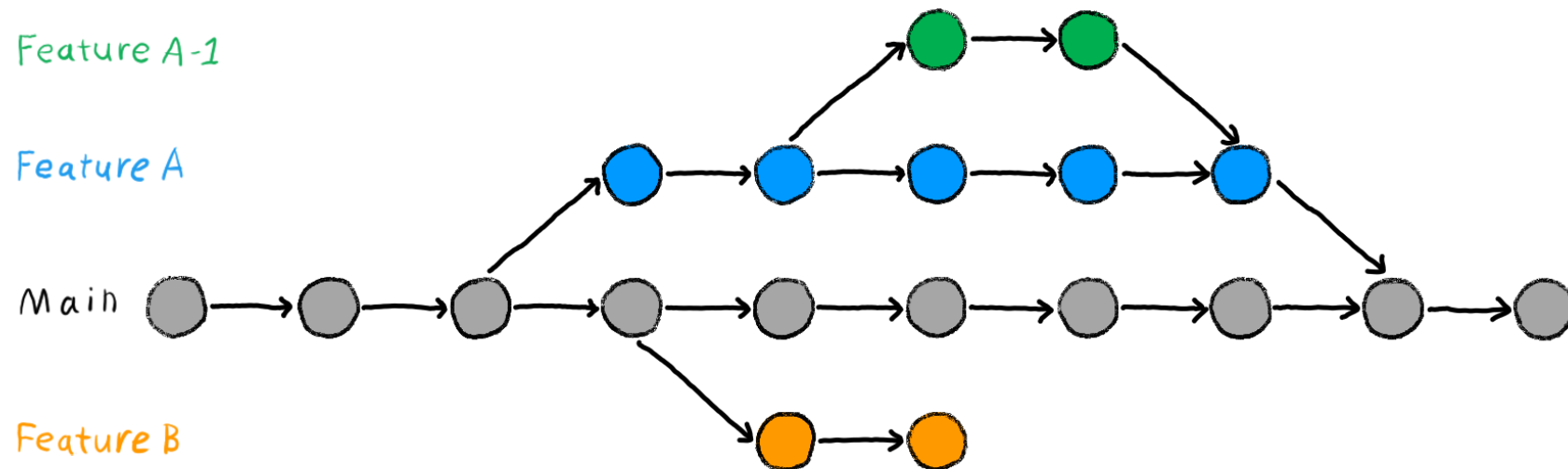
Eine Momentaufnahme des Codes zu einem bestimmten Zeitpunkt

-> Hiermit können Veränderungen am Code erzeugt werden



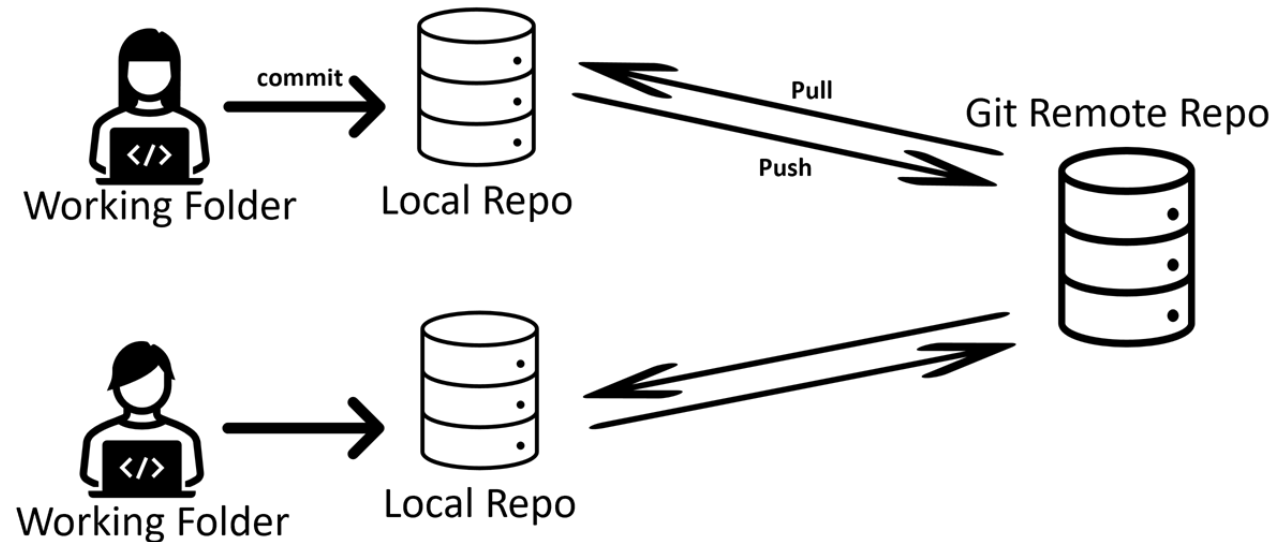
Branch

Ein paralleler Arbeitszweig für Änderungen oder neue Features



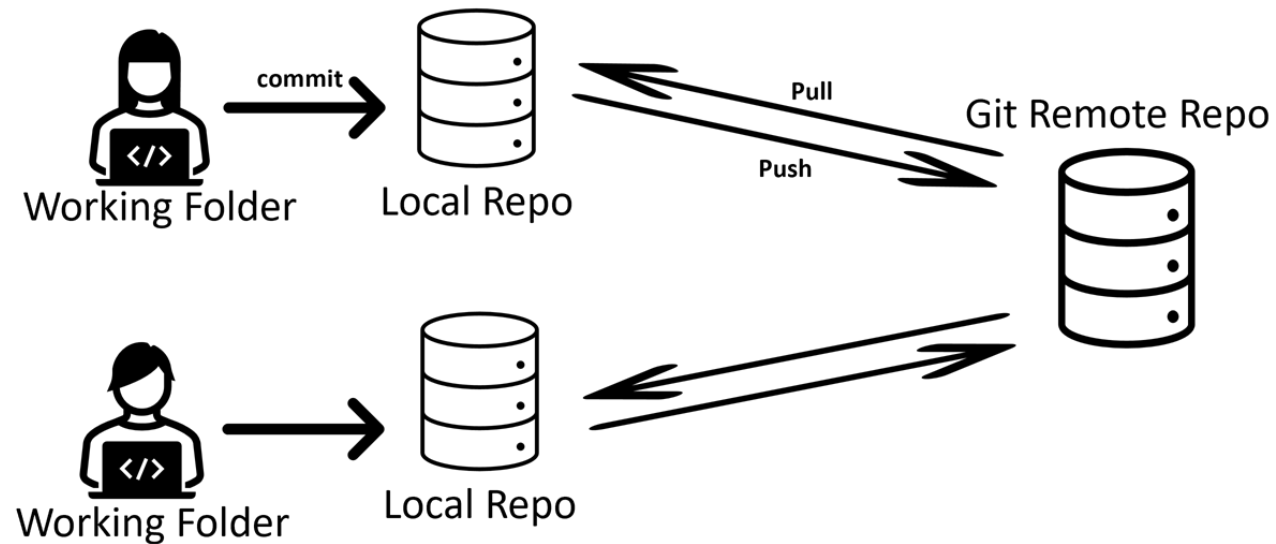
Pull und Push

- **Pull:** Holt Änderungen vom Remote-Repository **und** integriert sie direkt in den aktuellen Branch. (hier auch **Update**)
- **Push:** Überträgt lokale Commits (Änderungen) ins Remote-Repository, damit sie für andere verfügbar sind.



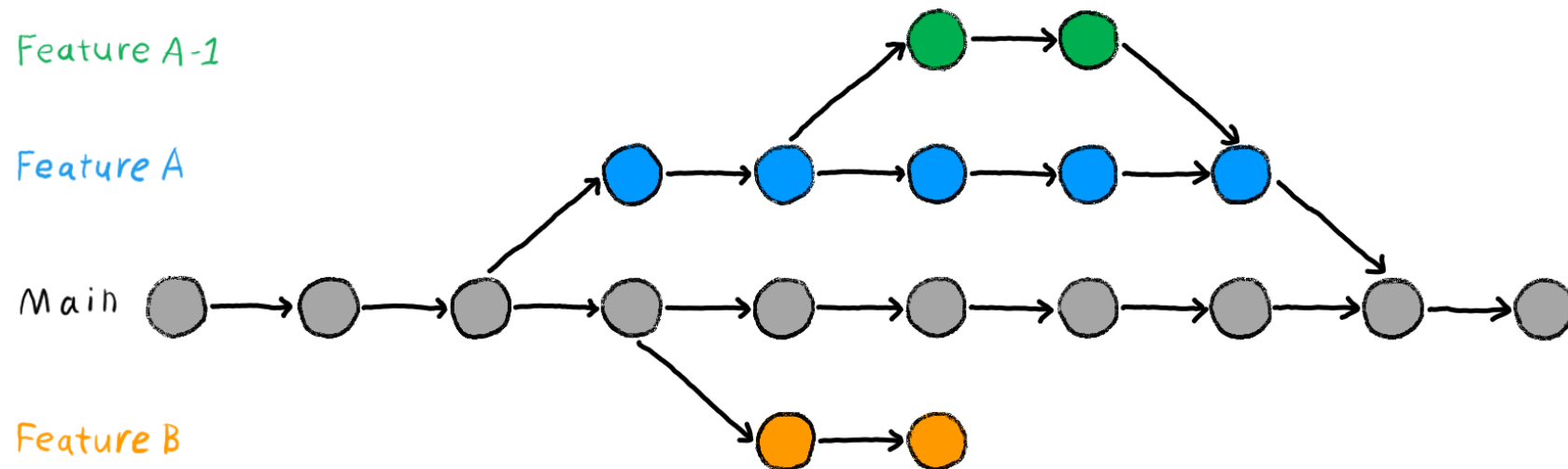
Fetch

- Holt Änderungen vom Remote-Repository, **ohne** sie in den lokalen Branch zu integrieren.



Merge

Zusammenführen von Branches und Übernahme der Änderungen in den Hauptzweig



Best Practices

- **Regelmäßig committen:** Kleine, häufige Commits erleichtern das Zurückverfolgen und Verstehen der Änderungen
- **Klare Commit-Botschaften:** Eine kurze und klare Beschreibung der Änderungen erleichtert die Zusammenarbeit