Enes Kurnaz, Oliver Nagel
Institut für Mathematik und Informatik

Freie Universität Berlin

Versionsverwaltung mit Git

Inhalt

- Einführung - Was bedeutet Versionsverwaltung?
- Git
  - Geschichte
  - Funktionsweise
  - Terminologie
- erste Schritte mit Git
- Git zur Zusammenarbeit benutzen
  - Syncing
  - Branches

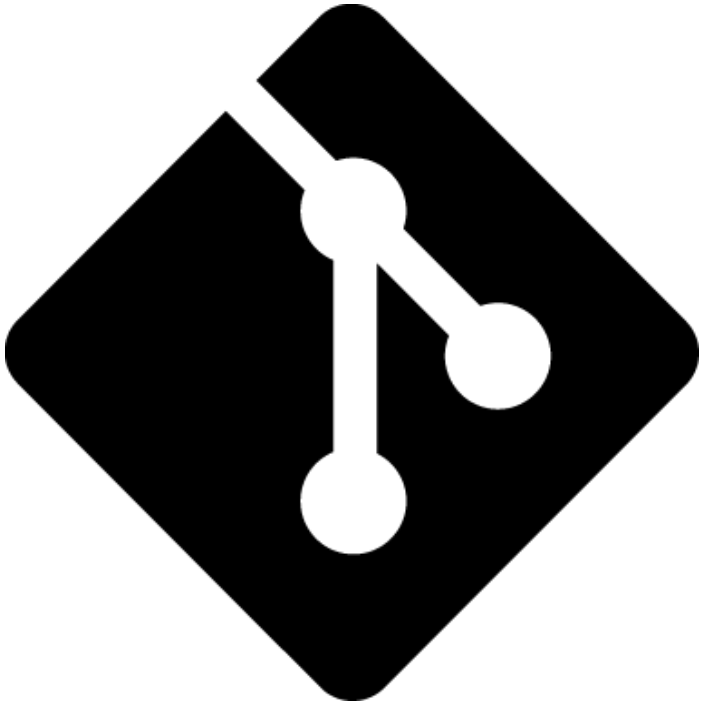
Was bedeutet Versionsverwaltung?

- System zum Management der Änderung von Dokumenten, Dateien und anderen Sammlungen von Daten

- Einsatz vor allem in der Softwareentwicklung

- Hauptaufgaben einer Versionsverwaltung:
    - Dokumentationsfunktion
    - Wiederherstellungsfunktion
    - Koordination des Zugriffs auf Dokumente und der Arbeit an einem Projekt

Git



- Beginn der Entwicklung von Linus Torvalds im April 2005
- Ersatz für BitKeeper in der Linux-Kernel-Entwicklung
- verteilte Versionkontrolle
- geschrieben in C
- unter GNU GPL Lizenz
- verfügbar für fast alle unixoiden Betriebsysteme
- Windowsportierung existiert

Terminologie

- Branch
  - Abspaltung von einer ursprünglichen Version der Entwicklung
  - hat eigene Versionsgeschichte

- Merging
  - Zusammenführen von zwei Branches
  - Fast-Forward-Merging und 3-Way-Merging

- Fork
  - neuer Branch in unabhängiger Versionsverwaltung

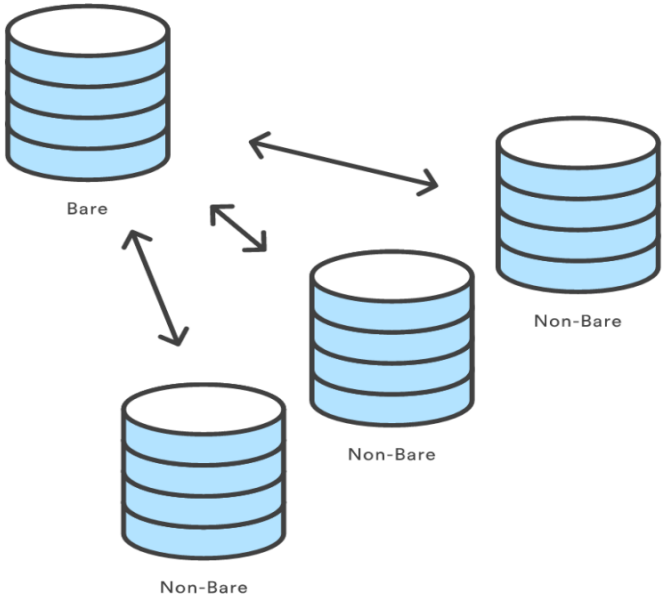
- Master
  - Hauptentwicklungszweig
  - unterscheidet sich technisch nicht von anderen Branches

Funktionsweise

- Inhalte eines Projektes liegen in Verzeichnissen (Repositorys)
- Änderungen werden in Arbeitskopie durchgeführt
- Übertragen einer Version aus einem Repository in die Arbeitskopie wird Checkout genannt
- das Übertragen der Änderungen in der Arbeitskopie in das Repository wird Commit genannt
- Git verwendet verteilte Versionsverwaltung, d.h. jeder Bearbeitende verfügt über eine eigene Kopie des Repositorys incl. lokaler Versionsgeschichte und Branchstruktur
- der Austausch von Daten zwischen Repositorys ist möglich
- zudem ist Branching und Forking ein integraler Bestandteil von Git

Setting up a repository

- git init - Erstellt Git Repository
- bare Repositories - Erstellt schreibgeschützte Git Repository mit der --bare flag
- git clone - Kopiert vorhandenes Repository
- git config - Konfiguration der Git Installation



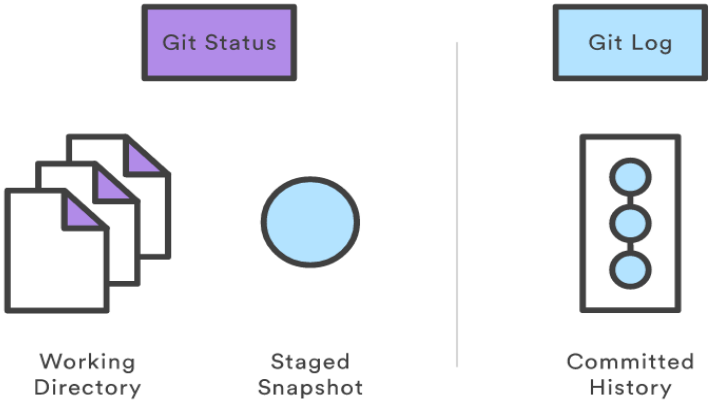
Saving changes

- git add - Fügt die Änderungen zum Sammelpunkt(staging area) hinzu
- git commit - Führt die Änderungen durch, die zu der staging area hinzugefügt wurden

Inspecting a repository

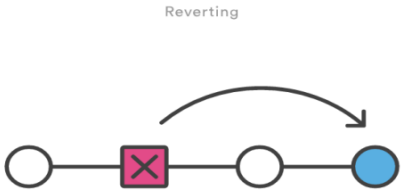
- git status - Zeigt an, welche Änderungen in der staging area/Arbeitsordner durchgeführt wurden und welche Dateien nicht rückverfolgt werden

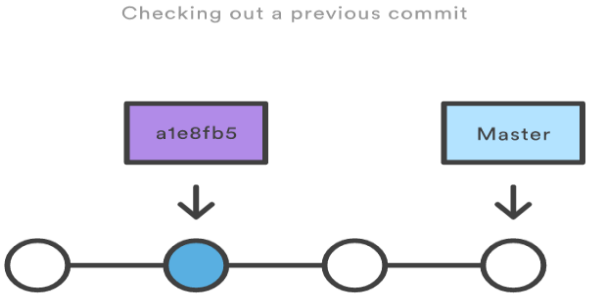
- git log - Zeigt die verbindlichen Änderungen an

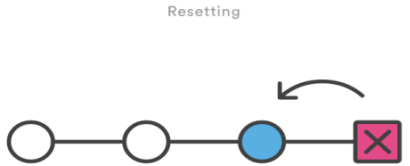


Undoing changes

- git checkout - Prüft Dateien & commits
- git revert - Hebt einzige Änderung(commit) auf
- git reset - Geht zu einem früheren Zustand
- git clean - Entfernt Dateien die nicht rückverfolgt werden können



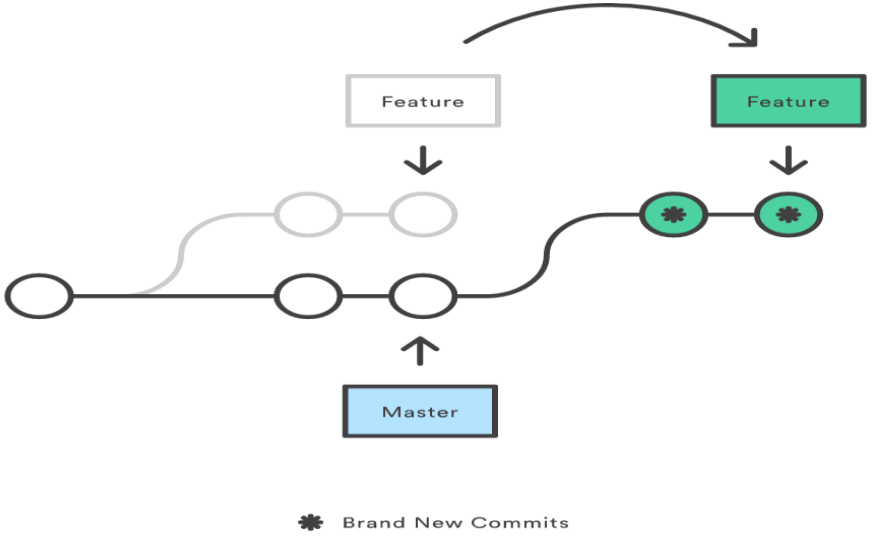




Rewriting history

- git commit --amend - editiert frühere Änderung(commit)

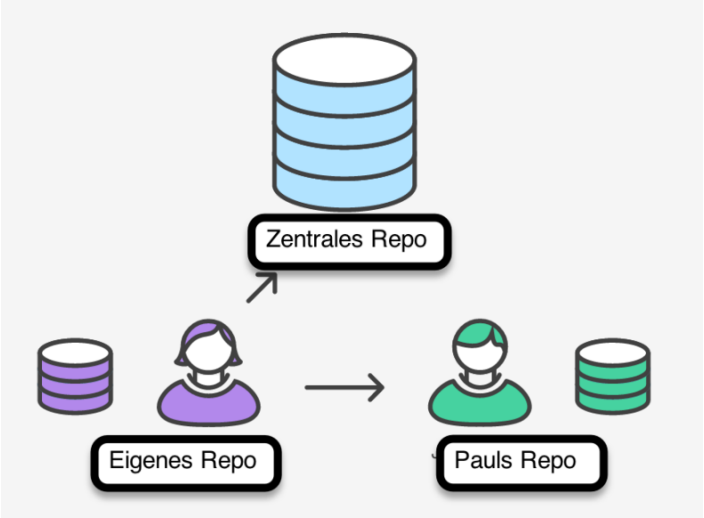
- git rebase - Ändert die Position von einem Zweig(Branch)
  - mit der -i flag können detaillierte Änderungen durchgeführt



Git zur Zusammenarbeit benutzen

- Syncing
  - git remote
  - git fetch/ git pull
  - git push
- Branches
  - git branch
  - git checkout
  - git merge

Syncing - git remote



- Austausch von Informationen zwischen Repositorys erfolgt nicht automatisch
- Verwaltung von Remote-Verbindungen zu anderen Repositorys
- bei Verwendung von git clone wird automatisch eine Remote-Verbindung zu geklonten Repo angelegt
- auch Verbindung zu Repositorys von anderen Bearbeitenden kann von Vorteil sein

Syncing - git fetch/git pull

- git fetch
  - Abrufen von Branches aus Remote-Repos
  - Commits aus Remote-Verbindungen können betrachtet und bewertet werden
  - Remote-Branches können in das lokale Arbeitsverzeichnis gemerged werden

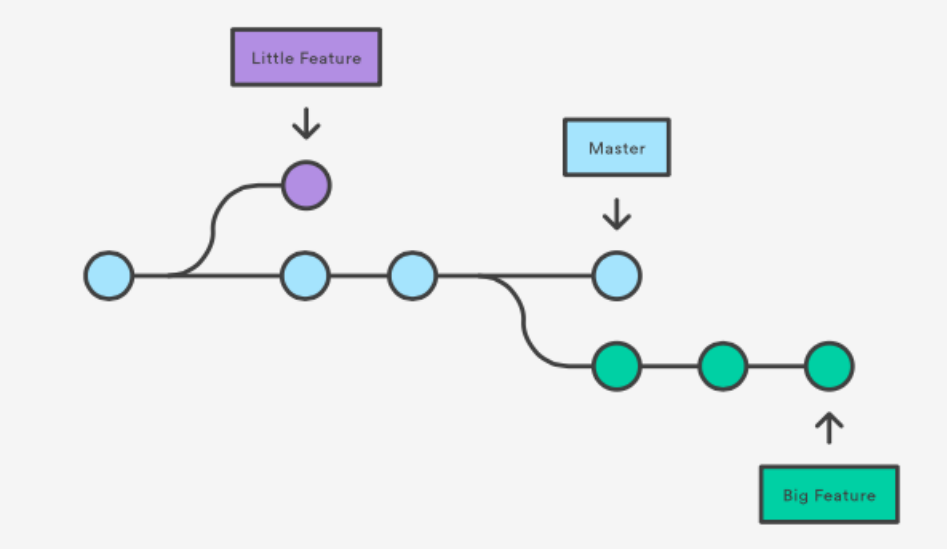
- git pull
  - Zusammenfassung von git fetch und git merge zu einem Befehl

Syncing - git push

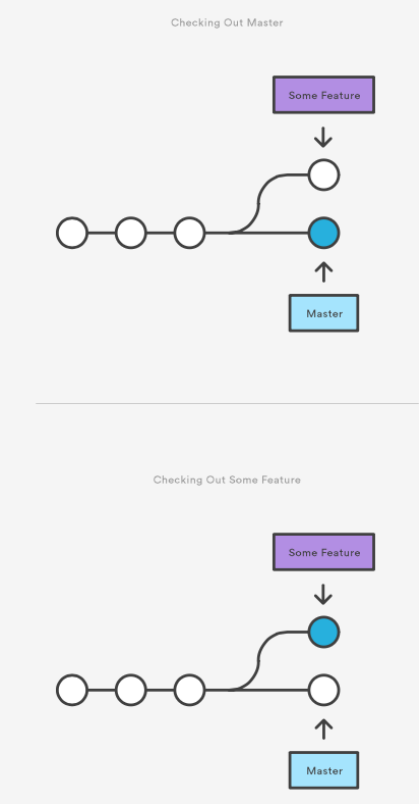
- Commits aus dem lokalen Repository in Remote-Repository exportieren
- häufigster Anwendungsfall ist das pushen in ein zentrales Projektrepository
- zwischenzeitliche Änderungen im Remote-branch könnten durch git push überschrieben werden
- git erlaubt daher nur pushes in Remote-Branches, wenn merge-Prozess “fast forward” für den Remote-branch ist

Branches - git branch

- mit git branch können Branches erstellt, umbenannt und gelöscht werden
- integraler Bestandteil der Arbeit mit Git
- Änderungen im Repo können abgekapselt behandelt werden
- Master-Branch kann vor instabilen oder fehlerhaftem Code bewahrt werden
- mit git branch -d [branch name] können Branches gelöscht werden



Branches - git checkout



- mit git checkout [existing_branch] kann zwischen den lokalen Branches gewechselt werden
- ausgewählter Branch wird in das Arbeitsverzeichnis übertragen
- mit git checkout [new_branch] wird ein neuer Branch erstellt und in das Arbeitsverzeicchnis übertragen
- erlaubt das parallele Bearbeiten von verschiedenen Features in einem Repository

Branches - git merge

- mit git merge [branch] können Branches zusammengefügt werden
- benannter Branch wird in das aktuelle Verzeichnis gemerged, der benannte Branch bleibt unverändert
- abhängig von der Revisionsnummer des Verzeichnisses wird zwischen Fast-Forward-Merge und 3-Way-Merge unterschieden
- Git wählt den Merge-Algorithmus selbstständig aus
- FF-Merging oft bei kleineren Änderungen angewendet
- 3-Way-Merging wird oft zur Integration von Features mit längerer Entwicklungszeit verwendet