

► Server-Hardware

► Server-Software

► Storage

► Virtualisierung

► Netzwerk+Zubehör

► Themenschwerpunkte

► Archiv

▼ Werkzeuge

Links auf diese Seite

Änderungen an verlinkten Seiten

Spezialseiten

Druckversion

Permanenter Link

Seiteninformationen

▼ In anderen Sprachen

English

Git Grundbegriffe

Hauptseite > Themenschwerpunkte > Git

Git ist ein **verteiltes Versionierungssystem** welches frei als Open-Source zur Verfügung gestellt wird. Einige bekannte Namen wie der Linux-Kernel, Eclipse oder Android verwenden Git für die Verwaltung ihrer Projekte.^[1] Ähnlich zu **Subversion (SVN)** [☞] wird Git für die **Versionskontrolle** (stetige Protokollierung von Änderungen) von Dateien eingesetzt. Im Gegensatz zu **SVN** gibt es jedoch einige grundlegende Unterschiede, andere Begriffe und Schemata können aber auch direkt umgelegt werden.^[2] Dieser Artikel erklärt **grundlegende Begriffe** die immer wieder in Zusammenhang mit Git auftauchen, außerdem werden die Prinzipien eines verteilten Versionierungssystems erläutert.

Inhaltsverzeichnis ^[Verbergen]

1

Was ist Git?

2

Wichtige Begriffe

3

Einzelnachweise

Was ist Git?

Git wird für die Versionierung von Dateien eingesetzt. Vor allem im Programmierbereich dient es dazu die eigenen Änderungen zu überwachen, sie rückgängig zu machen, sie anderen über sogenannte "Repositories" (Repos) zur Verfügung zu stellen oder Aktualisierungen von anderen einzuholen. Im Gegensatz zu SVN besitzt Git kein zentrales Repository sondern tritt als verteiltes System auf - aus der Sicht eines Bearbeiters ist die eigene Arbeitskopie (Working Copy) ein eigenes Repo für sich. Bei Bedarf können Änderungen von öffentlich zugänglichen Repos in die eigene Working Copy eingespielt werden oder die eigenen Änderungen z.B. über einen Patch den anderen zugänglich gemacht werden.

Dadurch ergeben sich am Repo für Bearbeiter folgende Möglichkeiten:^[3]

- Die eigene Arbeit kann einfach wieder in die zentrale Basis integriert werden.
- Es kann zeitgleich weiterentwickelt werden, z.B. jeder an verschiedenen Features.
- Die Versionierung verhindert, dass bereits getätigte Arbeiten verloren gehen bzw. überschrieben werden.
- Bei Bedarf kann zu früheren Versionen zurückgekehrt werden oder simultan an verschiedenen Versionen gearbeitet werden (auch "Branches" genannt).

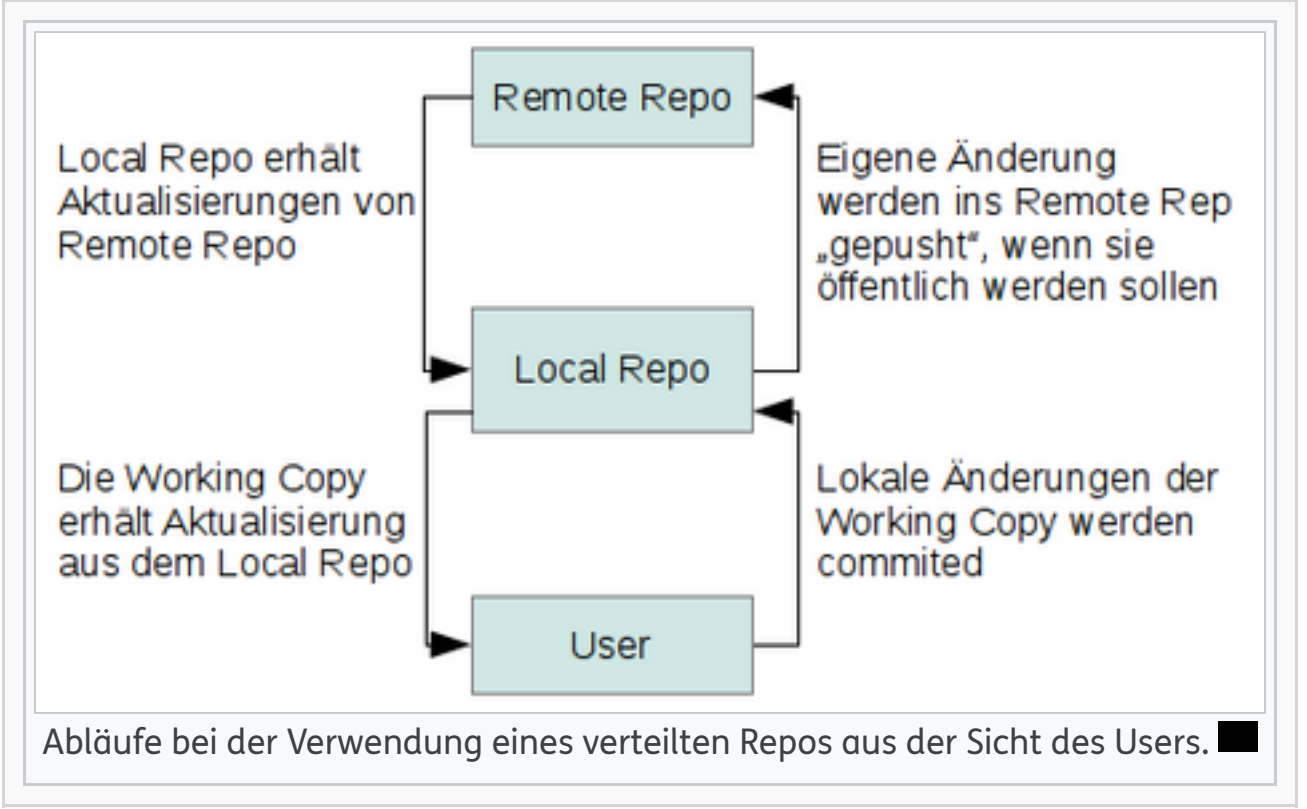
Git bietet aber noch viel mehr, welche Eigenschaften es anderen Systemen überlegen macht kann z.B. unter **Why Git is better than X** [☞] nachgelesen werden.

Wichtige Begriffe

- Repository und Working Copy:** In ein Repository bzw. einem Repo befinden sich alle Dateien inklusive derer vorangegangenen Versionen. Dadurch stehen stets alle Änderungen zur Verfügung, die von einer Datei ins Repo gespielt wurden und es kann nachvollzogen werden Wer, Wann, welche Änderungen durchgeführt hat. Das besondere an Git ist, dass jede lokale Working Copy eines Users (ein "Klon" - via "git clone") wieder ein vollständiges, eigenes, lokales Repo darstellt. Es existieren somit mehrere Kopien der Repos, der, der einen Klon besitzt, kann daran arbeiten - inklusive kompletter History, auch offline und ohne Abhängigkeit von einem zentralen Server.^[4] Die Änderungen aus dem eigenen Repo/der Working Copy können dann auf einen Ruck oder Schritt für Schritt, wenn sie als public angesehen werden, wieder in das Remote-Repo "gepusht" werden (git push).
- Branches:** Beim Einsatz von Git dienen Branches (engl.: to branch - sich verzweigen) dazu, um einen separaten Arbeitszweig zu erstellen. Dieser kann dann auch als neuer Kontext gesehen werden, in dem gearbeitet wird. So kann z.B. die Programmierung eines Sicherheits-Patches in einem eigenen Branch erfolgen (im Kontext des Patches), der bei Fertigstellung und nach dem Testen zurück in den Master-Zweig eingearbeitet wird.
- Versionierung:** Sobald an einer Working Copy gearbeitet wird protokolliert Git alle getätigten Änderungen mit. Mittels commit können die Änderungen zu dem Repository hinzugefügt werden, eine neue Version der Datei(n) befinden sich dann im Repo. Anschließend können verschiedene Versionen miteinander verglichen, Änderungen rückgängig oder zu einer früheren Version zurückgekehrt werden. Die Log-Informationen, die von Git mit aufgezeichnet werden können mit "git log" ausgegeben werden, "git status" listet die noch nicht ins Repo gespielten Änderungen der Working Copy auf.

Einzelnachweise

- ↑ **Git-Übersicht** [☞] (Git Projekt-Webseite)
- ↑ **Git Crash Course for SVN users** [☞](kernel.org)
- ↑ **Git for designers** [☞] (hoth.entp.com)
- ↑ **Why GIT?** [☞] (issaris.blogspot.com)



Druckversion

Das könnte Sie auch interessieren

Git Commits mit rebase zusammenführen

Zum Artikel

Git diff mit odt-Dateien

Zum Artikel

Git Grundbefehle

Zum Artikel

Kategorie: Git