IN DIESEM KAPITEL

Kapitel 1
Was ist Git?

Git ist bei Weitem die am häufigsten eingesetzte Versionsverwaltung weltweit, und der Abstand zu den anderen Produkten wird mit jedem Jahr größer. Die Zahl von Softwarelösungen steigt und steigt: Smart-Cities, Smart-Products, Autos, Straßenlaternen und Küchenmaschinen … Neben dem klassischen Personal Computer und dem Smartphone werden immer mehr digitale Produkte auf den Markt gebracht, und jedes dieser Produkte basiert auf Software, deren Quellcode mit großer Wahrscheinlichkeit in einem Git-Repository liegt!

Git ist also für Entwickler das, was der Hammer für die Zimmerleute ist: ein alltägliches Werkzeug, bei dem ein effizienter und sicherer Umgang eine Selbstverständlichkeit ist.

Aber was ist denn nun dieses Git?

Git – das aktuelle Logo ist in Abbildung 1.1 zu sehen – ist eine Software zur verteilten Versionsverwaltung.

Abbildung 1.1: Das heutige Git-Logo seit 2012

Quelle: `https://git-scm.com/downloads/logos`

Versionsverwaltung – zentral oder verteilt?

Eine *Versionsverwaltung* ist ein System zur Erfassung von Änderungen an Dateien. Alle Versionen werden gespeichert und können später wiederhergestellt werden. Im Unterschied zu Dokumentenmanagementsystemen, in denen Versionen nur für einzelne Dateien gelten, können sich Versionen bei Versionsverwaltungen auf mehrere Dateien beziehen (siehe Abbildung 1.2). Änderungen an mehreren Dateien werden also zu Versionen zusammengefasst und gespeichert. Die Versionen können einzeln wiederhergestellt werden.

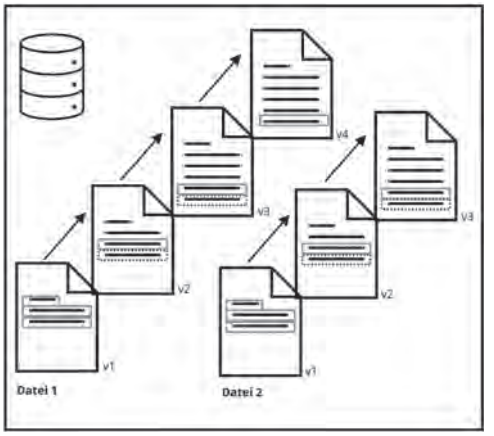

Abbildung 1.2: Versionsverwaltung von Dateien

Systeme zur Versionsverwaltung lassen sich in zwei Kategorien einteilen: zentral und dezentral.

Bei einer *zentralen Versionsverwaltung* werden die Änderungen in einem zentralen Speicher – dem sogenannten *Repository* – abgelegt und von dort von allen Clients abgerufen. Das Abrufen der Dateien nennt man auch *Checkout*. Nach dem Bearbeiten werden die Änderungen dann wieder in das zentrale Repository übertragen. Dies nennt man auch Checkin (siehe Abbildung 1.3).

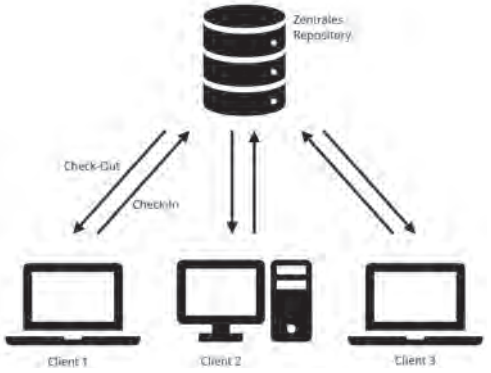

Abbildung 1.3: Zentrale Versionsverwaltung

Bei zentralen Repositories kann die Datei beim Checkout für andere Benutzer gesperrt werden. Dies verhindert, dass in der Zwischenzeit ein anderer Benutzer dieselbe Datei ändert. Werden Dateien nicht gesperrt, dann kann in der Zeit, in der ein Benutzer eine Datei lokal bearbeitet, ein anderer Benutzer seine Änderungen an derselben Datei im zentralen Repository speichern. In diesem Falle entsteht ein Konflikt beim späterem Checkin der Datei des ersten Benutzers. Dieser muss dann vom Benutzer vor dem Checkin gelöst werden, indem er die Änderungen erst zusammenführt – das sogenannte *Mergen*. Egal, für welche der beiden Optionen man sich entscheidet – sperren oder nicht sperren –, man benötigt zum Arbeiten mit der Versionsverwaltung immer eine Verbindung zum zentralen Repository. Es ist nicht möglich, offline mit Versionen zu arbeiten.

Im Gegensatz dazu werden bei einer *verteilten Versionsverwaltung* alle Versionen in einem lokalen Repository gespeichert. Ein Benutzer kann also komplett autark mit der Versionsverwaltung arbeiten. Die Änderungen können dann mit anderen Repositories synchronisiert werden (siehe Abbildung 1.4).

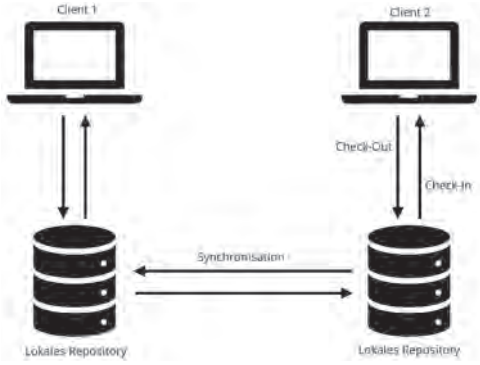

Abbildung 1.4: Verteilte Versionsverwaltung

Natürlich heißt das nicht, dass jeder Client sein Repository mit allen anderen Clients synchronisiert. Auch bei einer verteilten Versionsverwaltung sollte man ein zentrales Repository haben, mit dem alle Clients regelmäßig synchronisieren. Das zentrale Repository ist auch das, von dem sich neue Clients eine Kopie laden, um zu arbeiten, oder von dem Release-Stände für Software erstellt werden.

Um bei einem Hardwarefehler keinen Datenverlust zu erleiden, sollte möglichst regelmäßig mit diesem zentralen Repository synchronisiert werden. Alle lokalen Änderungen gingen sonst verloren.

Das zentrale Repository ist aber bei einer verteilten Versionsverwaltung nur eine Konvention. Prinzipiell kann ein Repository mit jedem beliebigen Repository synchronisiert werden.

Die Geschichte von Git

Git wurde 2005 von *Linus Torvalds* – dem Initiator des Linux-Kernels – ins Leben gerufen. Dies geschah nach einer Lizenzänderung von *BitKeeper*, einem proprietären Programm zur

Versionsverwaltung, das bis dahin für die Entwicklung des Linux-Kernels genutzt wurde. Durch die Änderung konnte BitKeeper nicht weiter kostenlos für die Entwicklung verwendet werden, und so entschied sich Linus Torvalds im April 2005 sein eigenes Programm zur Versionsverwaltung zu entwickeln.

Die drei Hauptanforderungen, die das Programm erfüllen sollte, waren folgende:

- ✔ verteilte Arbeitsabläufe,
- ✔ Sicherheit,
- ✔ Effizienz.

Es gab bereits ein existierendes Projekt namens *Monotone*, das die ersten beiden Kriterien erfüllte – das dritte jedoch nicht. Deshalb entschied sich Linus Torvalds, ein eigenes, neues Projekt zu erstellen. Er verwendete Ideen und Konzepte aus BitKeeper und Monotone, jedoch keinen Quellcode.

Das erste Logo von Git (siehe Abbildung 1.5) wurde im April 2007 veröffentlicht und erst 2012 durch das heutige Logo ersetzt.

Abbildung 1.5: Das originale Git-Logo von 2007

Quelle: `https://commons.wikimedia.org/wiki/File:Git-logo-2007.svg`
Copyright: Git / GPL `http://www.gnu.org/licenses/gpl.html`

Was bedeutet der Name »Git«?

Das Wort »Git« stammt aus dem britischen Englisch und bedeutet umgangssprachlich »Blödmann«, »Depp« oder »Idiot«. Linus Torvalds erklärte die Wahl des Namens folgendermaßen:

> *Ich bin ein egoistischer Mistkerl, und ich benenne all meine Projekte nach mir: zuerst »Linux«, jetzt eben »Git«.*

Das war aber nur ein Witz. In Wirklichkeit fiel die Wahl auf das Wort, da es folgende Eigenschaften aufwies:

- ✔ Es ist kurz.
- ✔ Es ist einfach auszusprechen.
- ✔ Es ist einfach auf einer Standardtastatur zu schreiben.
- ✔ Es wurde bis dahin noch nicht verwendet.

Der Name *Linux* wurde anfangs nicht von Torvalds selbst propagiert und nur sehr widerwillig von ihm akzeptiert!

Lizenz und Betriebssysteme

Git ist eine freie Software und unter der GNU General Public License, Version 2, verfügbar. Der Quellcode liegt auf GitHub und jeder, der möchte, kann sich an dem Projekt beteiligen. Git ist also auch kostenlos verfügbar.

Es gibt eigentlich kein Betriebssystem, auf dem Git nicht eingesetzt werden kann. Es gibt eine Version für Linux, Windows, macOS, Solaris und andere Unix-Betriebssysteme.

Ausblick

Git hat sich am Markt der Versionsverwaltungen durchgesetzt. Sein Einsatzgebiet reicht weit über das einer reinen Quellcode-Verwaltung hinaus. Zwar gibt es noch einige Nischenprodukte, doch zeigt sich bei den Tools zur Versionsverwaltung eine klare Konsolidierung. Git ist gekommen, um zu bleiben. Gerade dass keine Firma mit kommerziellem Interesse dahintersteckt, ist ein großer Vorteil von Git. Firmen müssen keine Angst davor haben, sich von einer anderen Firma abhängig zu machen.

Wenn es also eine Quellcode-Verwaltung gibt, mit der Sie sich heute in der Tiefe auseinandersetzen sollten, dann ist das Git. Die aufgewendete Zeit ist eine sichere Investition in die Zukunft.

Die Welt basiert zunehmend auf Software, und Software wird mit Hilfe von Git hergestellt.

Kurz und knackig

- ✔ Git ist eine freie Software zur verteilten Versionsverwaltung.
- ✔ Das Projekt wurde 2005 von Linus Torvalds ins Leben gerufen.
- ✔ Git bedeutet so viel wie »Blödmann« oder »Idiot«.
- ✔ Git ist eine freie Software und kostenlos für fast alle Betriebssysteme verfügbar.
- ✔ Git ist heute die am häufigsten eingesetzte Versionsverwaltung weltweit.