



Eine Software zur Versionsverwaltung dient dazu, den Veränderungsverlauf von Dateien und Verzeichnissen aufzuzeichnen und diese gegebenenfalls bei Problemen auf einen früheren Arbeitsstand zurücksetzen zu können. Hierbei werden oft noch weitere Werkzeuge angeboten, um Arbeitsschritte an derselben Datei aufzuteilen und verschiedene Versionen dieser Datei wieder zusammen zu führen. Dadurch kann kooperativ mit mehreren Personen an einem Projekt gearbeitet werden ohne versehentlich wichtige Daten zu löschen. Im Folgenden soll darauf eingegangen werden, was die Vorzüge des Git-Systems sind und warum *GitLab* in Vergleich zu anderen Softwarelösungen hervorsteicht, um Quellcode zu organisieren und versionieren.

Software zur Versionsverwaltung am Beispiel von Git

Git ist ein verteiltes Versionsverwaltungssystem, welches die Veränderungsabläufe von Dateien aufzeichnet. Es besteht durch seine dezentrale Arbeit an Projekten, bei der ein offizielles Repository mit der gemeinsamen Version der Dateien auf einem Server existiert und auf vielen weiteren Rechnern mit einer Kopie gearbeitet werden kann. Hierzu wird keine konstante Internetverbindung benötigt. Mit den lokalen Arbeitskopien können *Branches* (dt. Zweige), von einem Projekt erstellt und auf Bedarf wieder mit dem offiziellen Repository zusammengeführt werden. Dies bietet viele Vorteile im kooperativen Arbeiten in Projekten und gilt als besonders schnell und effizient bei der Veränderung von nur kleinen textbasierten Dateien.



Alternativ zu Git existieren noch viele andere Systeme, um Dateien und Verzeichnisse zu versionieren. Dabei sei SVN (*Apache Subversion*) zu nennen, welches ebenfalls weit verbreitet ist. Es gilt als leichter zu bedienendes zentrales System und kann besser mit großen und komplexen Dateien umgehen. In der Praxis werden deshalb oft Quellcode und Dokumente mit Git verwaltet, während Bilder, Videos und andere große binäre Formate, die nicht mit einem Texteditor lesbar sind, auf einem SVN System liegen.

GitLab – Eine Webanwendung zur Verwaltung von Softwareprojekten

GitLab ist eine Webanwendung zur Versionsverwaltung, die auf Git basiert. Die Open Source Software hat den Vorteil, dass sie als System kostenlos auf einem eigenen Server aufgesetzt werden kann. Hierbei können Projekte selber organisiert und Zugänge zum System konfiguriert werden. Viele automatisierte Funktionen helfen im Softwareentwicklungsprozess, angefangen bei der Planung, der Erstellung und dem Aufbau, bis hin zu dem Prüfen, den Testen, der Implementierung und der Überwachung. Alternativ gibt es auch verwandte Softwarelösungen wie *Gogs* oder *Gitolite*, die es ebenfalls erlauben, lokal gehostete Repositorien aufzustellen, aber die nicht den gleichen Funktionsumfang wie GitLab bieten.



Des Weiteren gibt es auch bekannte öffentliche Plattformen wie *GitHub* und *Bitbucket*, die ebenfalls Softwareprojekte hosten und als Grundlage Git verwenden. Auch wenn diese einen großen Beitrag zur *Open Source* Bewegung beisteuern, haben sie doch zwei große Nachteile: 1) Die Softwareprojekte müssen meist öffentlich gestellt werden, wenn man nicht einen gewissen Aufpreis zahlt, und 2) die entsprechenden Daten liegen auf fremden Servern. Da während der Projektphase oft mit empfindlichen Daten umgegangen wird, wie zum Beispiel mit urheberrechtlich geschützten Bildern oder Videos oder mit personenbezogenen Ergebnisdaten aus einer Umfrage, dürfen diese nicht einfach öffentlich gestellt werden oder auf externen Speichern liegen. *GitLab* erlaubt die lokale Bearbeitung von Dateien, bereitet diese aber auch so vor, dass die Überführung zum Beispiel zu *GitHub* am Ende des Projekts kein Problem darstellt. Dies wird dadurch unterstützt, dass *GitLab* und *GitHub* eine vergleichbare Infrastruktur haben. Die unterste zu verwaltende Ebene stellen die Repositorien dar, also die einzelnen Projekte. Diese können in unbestimmt vielen Gruppen und Untergruppen organisiert werden. Das hat den Vorteil, dass nicht nur eine kleine Arbeitsgruppe *GitLab* verwenden kann, um ihre Projekte in wenigen Teilbereichen zu gliedern, sondern auch eine große Institution, um Projekte auf mehreren Organisationsebenen zu trennen.



Der Umstieg von CVS auf Git – im Gespräch mit der Organisation Webis

Die Webis-Gruppe (kurz für Webtechnologien und Informationssysteme; Webseite: <https://webis.de>) ist eine Forschungsgruppe mit Sitz an der Bauhaus-Universität Weimar. Ihre Forschungsschwerpunkte liegen in den Bereichen Web Mining und Retrieval, Machine Learning, Computational Linguistics und Symbolic Artificial Intelligence. Über mehrere Jahre hinweg wurde hier die Software CVS (Concurrent Versions System) verwendet, um die eigenen Softwareprojekte und Dateien mit einer Versionskontrolle zu organisieren. 2015 wurde beschlossen, die Softwareprojekte auf Git zu transferieren und diese mit der Weboberfläche von GitLab leichter zugänglich zu machen.



Digital Bauhaus Lab, Foto: Tina Meinhardt

Grund für die Umstellung war unter anderem die Tatsache, dass CVS aktuell nicht mehr weiterentwickelt wird und bestimmte Mechaniken für die Softwareentwicklung hier ungünstig ausfallen. So kann es zum Beispiel passieren, dass es zu Inkonsistenzen bei der Aktualisierung eines Projekts kommt, weil CVS Veränderungen nur auf der Datei-Ebene aufzeichnet, während Git auf der Commit-Ebene das ganze Projekt aktualisiert. Das Vorgehen ist damit näher an der eigentlichen Prozedur der Softwareentwicklung, bei der ein Commit einem Releasestand der Software entsprechen kann. Ein anderes Problem ist zum Beispiel die fehlende Verfolgung von Namensänderungen, die zu einer Teilung der Historie eines Projekts führen kann. Git benutzt heuristische Regeln, um Inhalte zu vergleichen und Namensänderungen einer Datei zu erkennen.

Die eigentliche Webanwendung von *GitLab* bringt andere diverse Vorteile mit sich. Durch sie bekommt man leicht einen Überblick zu den Softwareprojekten und die Zugänge können je nach Benutzergruppe festgelegt werden. Features wie das Rendern der Readme-Dateien und das Anlegen von Wikiseiten erlauben es, die Projekte gut zu dokumentieren und durch einen *Issue-Tracker* können Probleme gemeldet, verfolgt und abgehandelt werden.

Webis baut weiterhin auf die Konfigurierbarkeit der Schnittstellen von *GitLab* indem es für die Anmeldung die lokale Authentifizierungsschnittstelle LDAP (*Lightweight Directory Access Protocol*) verwendet. Dadurch können sich sowohl Mitarbeiter als auch Studierende mit ihren bekannten Anmeldedaten der Universität in das System einloggen, ohne sich neue Anmeldedaten einfallen lassen und merken zu müssen.

Open Source und Community

GitLab ist eine Open Source Software, die mit den Programmiersprachen Ruby und Go entwickelt wurde. Sie ist damit frei verfügbar und kann komplett nach den eigenen Bedürfnissen konfiguriert, ggf. aber auch erweitert oder verändert werden, wenn die entsprechenden Programmierkenntnisse vorliegen. Sollten einem Fehler in der Software auffallen, so kann man diese über ein Forum oder über einen Bug Tracker melden. Dank des Entwicklers GitLab Inc. und großen Community kann man sich in der Regel auf eine schnelle Lösung verlassen.

Um GitLab zu nutzen, können verschiedene Wege eingeschlagen werden. Die primäre und bisher vorgestellte Variante ist das Aufsetzen der Software auf einem eigenen Server. Da dies mit einem erhöhten technischen Aufwand verbunden ist und die Software zunächst sicher eingerichtet und konfiguriert werden muss, ist es ratsam hier das Rechenzentrum bzw. den nächstliegenden IT-Support anzufragen. Oft existieren bereits Instanzen von GitLab an zentralen Einrichtungen, sodass das Anlegen einer neuen Instanz kein größeres Problem darstellt. Weiterhin bietet aber auch GitLab Inc. selbst seit 2012 als SAAS (Software as a Service) das Angebot, private und öffentliche Repositorien zu hosten. Hierzu kann ein kostenpflichtiger Support gebucht werden.

Bei Fragen zum Thema GitLab oder anderen Softwarerepositorien wenden Sie sich bitte an Herrn Kevin Lang, Kontaktstelle Forschungsdatenmanagement an der Bauhaus-Universität Weimar (kevin.lang@uni-weimar.de).



Haben Sie Fragen zum vorgestellten Best Practice oder wollen Sie ein Best Practice vorschlagen?

Kontaktieren Sie uns: info@forschungsdaten-thuringen.de