



Hochschule Osnabrück
University of Applied Sciences

Fallstudie Entwicklungswerkzeuge

Ausarbeitung

über das Thema

GIT Versionsverwaltungssystem

Autor: Vedad Hamamdžić
email@email.de

Prüfer: Paul Layer

Abgabedatum: 18.11.2014

I Zusammenfassung

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Abstract

Das ganze auf Englisch.

II Inhaltsverzeichnis

I	Zusammenfassung	I
II	Inhaltsverzeichnis	II
III	Abbildungsverzeichnis	III
IV	Tabellenverzeichnis	IV
V	Listing-Verzeichnis	IV
VI	Abkürzungsverzeichnis	V
1	GIT	1
1.1	Was ist GIT	1
1.1.1	Lokale Versionskontrollsysteme	1
1.1.2	Zentralisierte Versionskontrollsysteme	2
1.1.3	Verteilte Versionskontrollsysteme	2
1.2	GIT Historie	3
2	GIT Grundlagen	4
2.1	Begriffe die man kennen sollte	4
2.2	Installation von GIT	4
2.3	Konfiguration von GIT	4
2.4	Hilfestellungen durch das System	4
3	Mit Git Arbeiten	6
3.1	Ein Git Repository anlegen	6
3.2	Änderungen nachverfolgen	6
3.3	Mit Git commiten	7
3.4	Commithistorie anzeigen	7
3.5	Änderungen rückgängig machen	7
4	Branching mit Git	8
4.1	Was ist ein Branch?	8
4.2	Einfaches Branching	8
5	Merging mit Git	9
5.1	Einfaches und Merging	9
6	Git in Netzwerken	10
6.1	Welche Protokolle unterstützt Git	10
7	Quellenverzeichnis	11
	Anhang	I
A	GUI	I

III Abbildungsverzeichnis

Abb. 1	Lokale Architektur	1
Abb. 2	Zentralisierte Architektur	2
Abb. 3	Verteilte Architektur	2

IV Tabellenverzeichnis

Tab. 1	Beispieltabelle	6
--------	---------------------------	---

V Listing-Verzeichnis

Lst. 1	Arduino Beispielprogramm	6
--------	------------------------------------	---

VI Abkürzungsverzeichnis

OSGi Open Service Gateway initiative

1 GIT

1.1 Was ist GIT

GIT ist ein Versionsverwaltungssystem. Doch was ist ein Versionsverwaltungssystem? Ein Versionsverwaltungssystem ist ein System, welches Änderungen an einer Datei oder eine Reihe von Dateien protokolliert, so dass bestimmte Versionen später wieder aufrufbar sind.¹ Um Problemen entgegenzuwirken die eine Amateurhafte Methoden der Versionsverwaltung mit sich bringen wie z.B das ständige kopieren neuer Versionen in ein Verzeichnis. Wurden diese Systeme entwickelt. Dabei unterscheidet man 3 Arten von Systemen. Der wesentlichste Unterschied besteht darin wie und wo die Daten gehalten werden.

1.1.1 Lokale Versionskontrollsysteme

Von Lokalen Versionskontrollsystemen spricht man wenn die Daten auf dem Lokalen System vorliegen (siehe Abbildung 1). Dabei werden die Dateien in einer Version Database (Repository) gehalten. Nach jedem Checkout wird automatisch eine neue Version im Repository erstellt. Somit entgeht man der Gefahr durch das oben erwähnte Kopieren in andere Verzeichnisse eine der Versionen zu überschreiben, da man vergessen hat die Datei umzubenennen. Natürlich ist diese Variante der Versionskontrolle für Große Projekte die im Team bearbeitet werden eher destruktiv. Ein Beispiel für Lokale Systeme ist RCS(Revision Control System). Eher geeignet für Teamwork sind die beiden andern Architekturen.

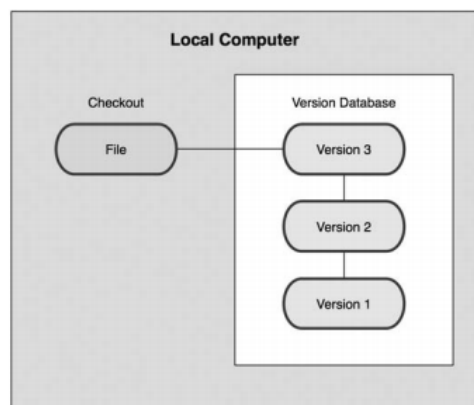


Abbildung 1: Lokale Architektur [Cha09]²

¹[Cha09] Seite 1 Zeile 1

1.1.2 Zentralisierte Versionskontrollsysteme

Bei zentralisierte Versionskontrollsystemen wird die Versionierung nicht lokal vorgenommen. Die Entwickler haben einen Zentralen Punkt (Abbildung 2), einen Server und dort befindet sich der Quellcode des Projektes in einem Repository zu deutsch Lager. Der unterschied zu einfachen Lokalen Systemen ist nun offensichtlich. Man braucht zumindest ein Netzwerk um solche Systeme zu nutzen. Ein sehr beliebtes zentralisiertes System ist Subversion. Ein weiterer Vorteil gegenüber der lokalen Versionnierung besteht darin das gemeinsames Arbeiten an einem Projekt möglich ist und bei Verwendung eines Servers der Online erreichbar ist kann das Arbeiten auch ohne Ortsbindung ablaufen. Doch dieser Vorteil der Ortsungebundenheit bietet einen enormen „Single Point of Failure“ denn wenn der Server ausfällt ist man nicht in der Lage seiner Arbeit nachzugehen.

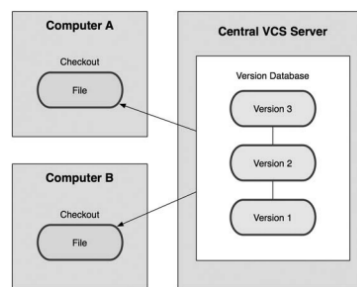


Abbildung 2: Zentralisierte Architektur³

1.1.3 Verteilte Versionskontrollsysteme

GIT gehört zu den Verteilten Systemen, der Unterschied zu den Varianten davor ist das sie beides können. Einer Art hybride Lösung. Man ist in der Lage Lokal zu Versionieren aber auch im Netzwerk Versionen, anderen zur Verfügung zu stellen (Abbildung 3). Jeder kann als Server fungieren und somit wird der „Single Point of Failure“ eliminiert den Zentralisierten Systeme haben. In der Praxis ist aber eher üblich das man einen Server nutzt vor allem bei Teamarbeiten. Wenn dieser jedoch ausfällt ist man in der Lage weiter seine Arbeit zu verrichten.

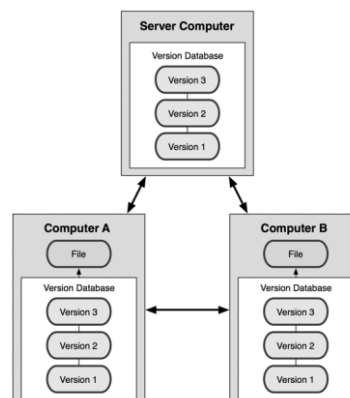


Abbildung 3: Verteilte Architektur⁴

1.2 GIT Historie

Im Jahre 2005 ist es zu unstimmigkeiten zwischen der Entwicklercommunity von Linux und dem Anbieter des proprietären BitKeeper-Systems das vorher kostenfrei genutzt wurde. Die Linux-Kernel-Entwickler mussten sich was einfallen lassen. Deswegen begann Linus Torvalds im April 2005 mit der Entwicklung von GIT und präsentierte auch sehr schnell die erste Version. Git baute auf den Erfahrungen mit BitKeeper auf doch die Hauptziele des neuen Systems waren⁵:

- Geschwindigkeit
- Einfaches Design
- Gute Unterstützung von nicht-linearer Entwicklung (tausende paralleler verschiedener Verzweigungen der Versionen)
- Vollständig verteilt
- Fähig, große Projekte wie den Linux Kernel effektiv zu verwalten

Durch die kontinuierliche Weiterentwicklung des Systems und die Benutzerfreundlichkeit wurde Git zu einem sehr beliebten Tool. Ein großer Einfluss auf den Erfolg von Git hat auch die Social Coding Plattform GitHub auf der man viele Open Source Projekte findet wie z.B :

- Der Linux Kernel⁶
- Ruby on Rails⁷
- Die Javascript Bibliothek JQuery⁸
- Das CMS Joomla⁹

Das sind natürlich nicht alle Open Source Projekte die GIT in Verbindung mit GitHub nutzen, aber einige bekannte die sich für Git entschieden haben. Der Dienst, den GitHub bereitstellt ist kostenfrei, doch nur unter der Bedingung dass die Projekte öffentlich zugänglich sind. Des Weiteren gibt es Optional wählbare Services die gegen Bezahlung verfügbar sind, aber auch eine Enterprise version die für Firmen interessant sein kann wird bereitgestellt.

⁵[Cha09] Seite 5

⁶[Tor]

⁷[Rub]

⁸[Jqu]

⁹[joo]

2 GIT Grundlagen

Um grundlegende Funktionen von GIT zu nutzen, ist es unumgänglich gewisse Begriffe zu kennen. Elementar hingegen ist der Umgang mit der Konsole des jeweiligen Systems. Es existieren einige plugins für Entwicklungsumgebungen wie z.B. Eclipse die mit einem GUI ausgestattet sind. Jedoch sind diese Plugins meistens nicht soweit entwickelt um den Kompletten Funktionsumfang des Systems bedienbar zu machen. Weswegen meine Erläuterungen zum Git System sich auf Linux als Betriebssysteme beziehen werden und nur mit der Konsole zu bedienen sind.

2.1 Begriffe die man kennen sollte

Bevor man mit Versionierungssystemen arbeitet sollte man einige begriffe kennenlernen.

Repository clone branch merge

2.2 Installation von GIT

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

2.3 Konfiguration von GIT

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

2.4 Hilfestellungen durch das System

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata

sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

3 Mit Git Arbeiten

Für Auflistungen wird die *compactitem*-Umgebung genutzt, wodurch der Zeilenabstand zwischen den Punkten verringert wird.

- Nur
- ein
- Beispiel.

3.1 Ein Git Repository anlegen

Zuletzt ein Beispiel für ein Listing, in dem Quellcode eingebunden werden kann, siehe Listing 1.

```

1  int ledPin = 13;
2  void setup() {
3      pinMode(ledPin , OUTPUT);
4  }
5  void loop() {
6      digitalWrite(ledPin , HIGH);
7      delay(500);
8      digitalWrite(ledPin , LOW);
9      delay(500);
10 }
```

Listing 1: Arduino Beispielprogramm

In diesem Abschnitt wird eine Tabelle (siehe Tabelle 1) dargestellt.

Name	Name	Name
1	2	3
4	5	6
7	8	9

Tabelle 1: Beispieltabelle

3.2 Änderungen nachverfolgen

Die Quellen befinden sich in der Datei *bibo.bib*. Ein Buch- und eine Online-Quelle sind beispielhaft eingefügt. [Vgl. [?], [Bre]]

Abkürzungen lassen sich natürlich auch nutzen (Open Service Gateway initiative (OSGi)). Weiter oben im Latex-Code findet sich das Verzeichnis.

3.3 Mit Git commiten

3.4 Commithistorie anzeigen

3.5 Änderungen rückgängig machen

4 Branching mit Git

Lorem ipsum dolor sit amet.

4.1 Was ist ein Branch?

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

4.2 Einfaches Branching

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

5 Merging mit Git

Lorem ipsum dolor sit amet.

5.1 Einfaches und Merging

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

6 Git in Netzwerken

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

6.1 Welche Protokolle unterstützt Git

7 Quellenverzeichnis

- [Bre] BRETTSCHEIDER, Daniel: *Daniel Brettschneiders Blog*. <http://www.daniel-brettschneider.de>. – Zugriff: 15.02.2013, Archiviert mit WebCite®: <http://www.webcitation.org/6ESWiGbhW>
- [Cha09] CHACON, Scott: *Pro Git*. Berkeley, CA New York : Apress Distributed to the Book trade worldwide by Springer-Verlag, 2009. – ISBN 9781430218340
- [joo] JOOMLA: *Joomla CMS*. <https://github.com/joomla/joomla-cms>
- [Jqu] JQUERY: *Jquery, jquery*. <https://github.com/jquery/jquery>. – Zugriff: 15.02.2013, Archiviert mit WebCite®: <https://github.com/jquery/jquery>
- [Rub] RUBY: *Ruby on Rails*. <https://github.com/rails/rails>. – Zugriff: 15.02.2013, Archiviert mit WebCite®: <https://github.com/rails/rails>
- [Tor] TORVALDS: *Linux Kernel, Torvalds*. <https://github.com/torvalds/linux>. – Zugriff: 15.02.2013, Archiviert mit WebCite®: <https://github.com/torvalds/linux>

Anhang

A GUI

Ein toller Anhang.

Screenshot

Unterkategorie, die nicht im Inhaltsverzeichnis auftaucht.

Erklärung

Hiermit versichere ich, dass ich meine Abschlussarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Datum:

.....

(Unterschrift)