Riemenschneider-Gymnasium Oberstufenjahrgang 2017/19

Würzburg

S E M I N A R A R B E I T

aus dem Fach

Informatik

Thema:

Anteilnahme an Open-Source Projekten mit Hilfe des Versionsverwaltungssystem Git

Verfasser: Timo Feuerbach

W – Seminar: Google

Kursleiter: Michael Müller

Abgabetermin: 07.11.2017

Ergebnis der Seminararbeit:

Erzielte Note: …………………………. in Worten: ………………………

Erzielte Punkte: …………………………. in Worten: ………………………

|  |
| --- |
|  |

(Unterschrift des Kursleiters)

Inhalt

[1. Einleitung 3](#_Toc497751969)

[2. Was ist Open Source 4](#_Toc497751970)

[3. Versionsverwaltung 7](#_Toc497751971)

[4. GitHub und Git an Hand eines praktischen Beispiels 10](#_Toc497751972)

[Quellenverzeichnis 11](#_Toc497751973)

## **Einleitung**

## **Was ist Open Source**

Open Source kommt aus dem Englischen und „heißt offener Quellcode und meint gemeinhin Software, die jeder nach Belieben studieren, benutzen, verändern und kopieren darf.“ [[1]](#footnote-1) Das bedeutet also, dass eine bestimmte Software nur als Open Source bezeichnet werden darf, wenn man ohne Probleme den Quelltext der Software lesen, bearbeiten und nutzen kann. Der Quelltext muss also dem Nutzer entweder mit dem Programm bereitgestellt werden oder für alle frei und einfach zu finden im Internet öffentlich gemacht werden. Das bedeutet allerdings nicht „wie häufig angenommen, *alles-ist erlaubt,* es sind Bedingungen an die Nutzung geknüpft.“ [[2]](#footnote-2) So ist es also nicht Bedingungslos erlaubt, den Quelltext eines anderen einfach herunterzuladen, nur gering oder gar nicht zu verändern und dann als sein eigenes weiter zu vervielfältigen. Es ist jedem erlaubt, die Software bedingungslos zu nutzen, jedoch wird die Verbreitung und die Vervielfältigungen „vielfach von bestimmten Voraussetzungen abhängig gemacht.“ [[3]](#footnote-3) Somit kann von sogenannter Public Domain Software, welche vom Nutzer uneingeschränkt und bedingungslos vervielfältigt, verbreitet und verändert werden darf, unterschieden werden. [[4]](#footnote-4) Der Quelltext, oder auch „Programmcode“ [[5]](#footnote-5) ist der Text einer Software, welcher in einer bestimmten Programmiersprache formuliert ist. Die frei verfügbare Softwareinfrastruktur bei Open Source Software soll der Allgemeinheit von praktischem Nutzen sein. [[6]](#footnote-6) Außerdem wird der Begriff Open Source häufig mit dem Begriff *Freie Software* in Verbindung gebracht oder gar verwechselt. Durch die Definition von Freier Software wird allerdings klar, dass meistens nicht der Preis der Software gemeint ist, sondern die Übersetzung von „frei“ aus dem Englischen „free“, welche in diesem Fall für Freiheit steht. Auch wenn Freie Software meist kostenlos ist, bezieht sich das „free“ darauf, dass einem Programmierer der Verwendungszweck freisteht. Durch die „Veröffentlichung der Freie-Software-Definition und den Worten *Denk‘ an Redefreiheit nicht Freibier* [[7]](#footnote-7) wurde versucht, diese Verwechslungen zu klären und zu verhindern, damit man ab diesem Zeitpunkt nicht mehr Freie Software mit Open Source Software verwechseln konnte. „Open Source-Programme (…) machen den Größen der Softwareindustrie ernste Konkurrenz. [[8]](#footnote-8) Durchaus praktisch, wenn einem das entsprechende Geld für bestimmte notwendige Software dann fehlt, so kostet das Software Paket „Microsoft Office“ je nach Verwendungszweck zwischen 69,00€ und 99,00€ pro Jahr [[9]](#footnote-9), natürlich existieren hierfür nicht nur eine, sondern gleich mehrere Open Source Alternativen, die zwei bekanntesten sind „LibreOffice“ und „Apache OpenOffice“ [[10]](#footnote-10), beide sind vollwertige Alternativen zu Microsoft Office, sie haben ähnliche oder gleiche Funktionen, kosten aber im Gegensatz zu Microsoft Office nichts und sind Open Source, das wohl bekannteste, größte und vom Marktwert teuerste Open Source Projekt ist das von der Firma „Red Hat“ geleitete Projekt „Linux“ [[11]](#footnote-11), dass seit dem Jahr „1991“ durchgehend weiterentwickelt wird. [[12]](#footnote-12) Damit die Verfügbarkeit des Quelltextes einer Open Source Software garantiert werden kann, muss diese logischerweise irgendwo veröffentlicht sein, auch hierfür gibt es einige unterschiedliche Varianten, eine davon ist SourceForge. „SourceForge (…) ist ein Filehosting-Dienst für Softwareprojekte. Programmierer können dort quelloffene Softwareprojekte erstellen und verwalten.“ [[13]](#footnote-13) Das ist aber nur eine der möglichen Varianten. Die Variante, mit der sich diese Arbeit beschäftigen soll nennt sich GitHub. „(…) GitHub hostet nicht nur kostenlos die Open Source Projekte der Entwickler (…).“ [[14]](#footnote-14) Heißt, dass GitHub ein Online File-Hosting Dienst ist. Das bedeutet, dass ein Programmierer, der sich dazu entschieden hat seine Software Open Source zu machen, die Möglichkeit hat, sein Projekt direkt auf die GitHub-Server hochzuladen und dieses im Anschluss öffentlich zu machen. Damit auf den Servern nicht das Problem auftritt, dass Dateien aus Projekten mit Dateien aus anderen Projekten unerwünscht vermischt werden und man auch die Möglichkeit hat, verschiedene Versionen seiner Software zu veröffentlichen, ohne dass andere Versionen der Software gelöscht oder verändert werden, wird ein gewisses Ordnungssystem benötigt. Dieses Ordnungssystem nennt man auch Versionsverwaltung, welches bei GitHub auch namensgebend war, „Git ist nämlich auch der Name einer Software zur Versionsverwaltung.“ [[15]](#footnote-15)

## **Versionsverwaltung**

GitHub ist im Prinzip nichts anderes, als ein GUI für das Versionverwaltungssystem Git. „Als Graphical User Interface (GUI) bezeichnet man die grafische Benutzeroberfläche, die das Bedienen der Computer erleichtert.“ [[16]](#footnote-16) Das heißt, dass die Benutzeroberfläche auf grafischer und visueller Elemente basiert, also stellt das elektronische Gerät mit Hilfe eines visuellen Wiedergabegeräts die Benutzeroberfläche einer Software dar. Auch die Bedienung dieser Software ist grafisch, also werden z.B. Knöpfe etc. auf dem visuellen Wiedergabegerät angezeigt, um per Klick oder Auswahl angesteuert und benutzt werden zu können. „Eine Versionsverwaltung ist ein System, das zu Erfassung von Änderungen an Dokumenten oder Dateien verwendet wird. Alle Versionen werden in einem Archiv mit Zeitstempel und Benutzerkennung gesichert und können später wiederhergestellt werden.“[[17]](#footnote-17) Das heißt, dass sobald ein Benutzer des Systems eine neue Datei anlegt, oder eine bereits vorhandene Datei modifiziert wird für diese Änderungen eine sogenannte neue Version angelegt, mit dem entsprechenden Änderungsdatum und der Benutzerkennung versehen und abgelegt, sodass man jeder Zeit zu dieser Version zurückkehren kann. Das dient dazu, dass bei versehentlichen oder absichtlich falschen Änderungen ohne Probleme zu einer älteren Version zurückgekehrt werden kann, so zum Beispiel funktioniert auch Wikipedia, da dort jeder einen Eintrag modifizieren kann, kann es durchaus passieren, dass jemand versehentlich oder absichtlich falsche Informationen über ein gewisses Thema verbreitet. [[18]](#footnote-18) Sollte das passieren, so kann dieser versehentlich oder mutwillig erzeugte Fehler schnell und problemlos durch das wiederherstellen einer älteren Version beseitigt werden. „Versionsverwaltungssysteme haben das Ziel, den Entwicklungsprozess von Software zu unterstützen.“ [[19]](#footnote-19)

„Zu den wichtigsten Aufgaben einer Versionsverwaltung gehören die Protokollierung der Entwicklung, die Möglichkeit der Archivierung und Wiederherstellung von alten Versionen und die Koordination des Zugriffs mehrerer Entwickler auf die selbe Datei.“ [[20]](#footnote-20)

Bei der Protokollierung hält die Versionsverwaltung sämtliche Änderungen fest, dass nachvollzogen werden kann, wer etwas verändert hat und was verändert wurde. Falls beim Ändern einer Datei ein Fehler gemacht wurde, so kann eine zuvor archivierte Version der Datei wiederhergestellt werden. Durch die Kontrollierung des Zugriffs wird das Überschreibung einer Datei, an der mehrere Personen arbeiten verhindert. „Historisch haben sich dabei drei Genres von Versionsverwaltungssystemen entwickelt: die lokale, die zentrale und die verteilte Versionsverwaltung.“ [[21]](#footnote-21) Die Lokale Versionsverwaltung funktioniert so, dass die Dateien eine Version erhalten und lokal auf einem Rechner gesichert werden.

„Die Versionierung erfolgt nur dateiwiese, nicht projektweise. Bei großen Projekten ist das problematisch. Da jede Datei ihre eigene Versionshistorie hat, ist es nicht ohne weiteres möglich, mehrere Dateien auf einmal auf die Version zurückzusetzen, die sie zu einem bestimmten Zeitpunkt hatte.“ [[22]](#footnote-22)

Der zweiten Variante, die Zentrale „Versionsverwaltung liegt ein „Client-Server-System zugrunde.“ [[23]](#footnote-23) „Das Client-Server Prinzip ist das Standardkonzept für die Verteilung von Aufgaben innerhalb eines Netzwerkes.“ [[24]](#footnote-24) Durch diese Verteilung kann ein Server eine Aufgabe nicht nur an einen, sondern gleich an mehrere Computer in einem Netzwerk geben, die Computer arbeiten dann gemeinsam an der Aufgabe und können diese dadurch schneller erledigen. Bei dieser Variante erfolgt die Versionierung projektweise, das bedeutet, dass nicht nur die veränderte Datei bei der Erstellung einer neuen Version neu versioniert wird, sondern das ganze Projekt. [[25]](#footnote-25) „Populär wurde die zentrale Versionsverwaltung durch Concurrent Versions System (CVS).“ [[26]](#footnote-26) Die letzte Art der Versionsverwaltung, verwendet kein zentrales Repository, so wie die zentrale, eine zentrale Versionsgeschichte existiert auch nicht mehr. [[27]](#footnote-27) „Ein Repository ist eine spezielle Datenbank zur systematischen Ablage von Modellen und deren Bestandteilen.“ [[28]](#footnote-28) Also werden bei einer Versionsverwaltung die einzelnen Versionen in so einer Repository abgespeichert. Dadurch, dass bei dieser Variante der Verwaltung jeder Entwickler sein eigenes Repository besitzt kann jeder Entwickler ohne Bedenken alle Dateien nach seinem Belieben verändern und speichern. Die Repositories können jederzeit unter den Entwicklern abgeglichen werden, dabei werden durch sogenanntes Merging identische Dateien zusammengefügt. So wird das gleichzeitige Bearbeiten der gleichen Datei durch mehrere Entwickler möglich gemacht. [[29]](#footnote-29) Ein paar Beispiele für verteilte Versionsverwaltungen sind: „Mercurial“, „GNU arch“, „Fossil“, „BitKeeper“, und „Git“. [[30]](#footnote-30)

## **GitHub und Git an Hand eines praktischen Beispiels [[31]](#footnote-31)**

„Prinzipiell ließe sich Git (…) auch komplett ohne ein Web-Interface oder zentralen Server verwenden.“ [[32]](#footnote-32) Git funktioniert auch mit den Konsolen der Betriebssysteme Windows, macOS und Linux, dafür müssen die Konsolen mit der Git-Software erweitert werden, welche im Internet zu finden ist, dadurch werden den Konsolen einige Befehle für die Verwendung von Git hinzugefügt, die hinzugefügten Befehle sind für alle Betriebssysteme gleich. Als erstes wird ein Repository benötigt, dieses wird mit den folgenden Befehlen [[33]](#footnote-33) erstellt. Ein neues Verzeichnis, welches als Grundlage für das Repository benötigt wird, wird mit dem Befehl „mkdir W-Seminar“ erzeugt. Mit dem Befehl „cd W-Seminar“ gelangt man in das neu erstellte Verzeichnis mit dem Namen „W-Seminar“. Von dort aus startet erst der eigentliche Teil der Repository-Erstellung, mit Hilfe des Befehls „git init“ initialisiert, also es wird festgelegt, dass alle Dateien in diesem noch leeren Verzeichnis zu dem zukünftigen Repository gehören sollen, die Konsole gibt die Antwort „Leeres Git-Repository in /home/timo/Schreibtisch/W-Seminar/.git/ initialisiert“.

## **Quellenverzeichnis**

Internetquellen

1. Bundeszentrale für politische Bildung, „Open Source“, unter: http://www.bpb.de/gesellschaft/medien/opensource/ (aufgerufen: 01.11.2017)
2. Wikipedia, „Open Source“, unter: https://de.wikipedia.org/wiki/Open\_Source (aufgerufen: 01.11.2017)
3. „Public Domain“, unter: https://www.roru.de/software/publicdomain.htm (aufgerufen: 01.11.2017)
4. Sebastian Brockmeyer, „Versionsverwaltung“, unter: https://github.com/UOS-Open-Source-Softwareentwicklung/oss1314/wiki/Versionsverwaltung (aufgerufen: 03.11.2017)
5. ITWissen, „Quelltext“, unter: http://www.itwissen.info/Quelltext-source-code.html (aufgerufen: 01.11.2017)
6. ITWissen, „GUI (graphical user interface)“, unter: http://www.itwissen.info/GUI-graphical-user-interface-Grafische-Benutzeroberflaeche.html (aufgerufen: 03.11.2017)
7. Fachadmin, „Client-Server Prinzip“, unter: https://www.fachadmin.de/index.php/Client-Server\_Prinzip (aufgerufen: 03.11.2017)
8. Richard Stallman, „Warum Open Source das Ziel Freie Software verfehlt“, unter: https://www.gnu.org/philosophy/open-source-misses-the-point (aufgerufen: 01.11.2017)
9. „Microsoft Office 2016 kaufen“, unter: https://www.microsoft.com/de-de/store/b/office (aufgerufen: 01.11.2017)
10. Springer Gabler Verlag, Gabler Wirtschaftslexikon, unter. http://wirtschaftslexikon.gabler.de/Archiv/569801/repository-v4.html (aufgerufen: 03.11.2017)
11. Moritz Stückler, „Was ist eigentlich GitHub?“, unter: http://t3n.de/news/eigentlich-github-472886/ (aufgerufen:01.11.2017)
12. Alternativeto, „Microsoft Office Suite“, unter: https://alternativeto.net/software/microsoft-office-suite/?license=opensource (aufgerufen: 01.11.2017)
13. Wikipedia, „Linux“, unter: https://de.wikipedia.org/wiki/Linux (aufgerufen: 01.11.2017)
14. Wikipedia, „SourceForge“, unter: https://de.wikipedia.org/wiki/SourceForge (aufgerufen 01.11.2017)
15. Wikipedia, „Versionsverwaltung“, unter: https://de.wikipedia.org/wiki/Versionsverwaltung (aufgerufen: 27.10.2017)

1. Bundeszentrale für politische Bildung, „Open Source“, unter: http://www.bpb.de/gesellschaft/medien/opensource/ [↑](#footnote-ref-1)
2. Wikipedia, Definition, „Open Source“, unter: https://de.wikipedia.org/wiki/Open\_Source [↑](#footnote-ref-2)
3. Ebd. [↑](#footnote-ref-3)
4. Vgl. „Public Domain“, unter: https://www.roru.de/software/publicdomain.htm [↑](#footnote-ref-4)
5. ITWissen, „Quelltext“, unter: http://www.itwissen.info/Quelltext-source-code.html [↑](#footnote-ref-5)
6. Vgl. Definition, „Open Source“ unter: https://de.wikipedia.org/wiki/Open\_Source [↑](#footnote-ref-6)
7. Richard Stallman, „Warum Open Source das Ziel Freie Software verfehlt“, unter: https://www.gnu.org/philosophy/open-source-misses-the-point [↑](#footnote-ref-7)
8. Bundeszentrale für politische Bildung, „Open Source“, unter: http://www.bpb.de/gesellschaft/medien/opensource/ [↑](#footnote-ref-8)
9. Microsoft, „Microsoft Office 2016 kaufen“, unter: https://www.microsoft.com/de-de/store/b/office [↑](#footnote-ref-9)
10. Alternativeto, „Microsoft Office Suite“, unter: https://alternativeto.net/software/microsoft-office-suite/?license=opensource [↑](#footnote-ref-10)
11. Wirtschaftliche Bedeutung, „Open Source“, unter: https://de.wikipedia.org/wiki/Open\_Source [↑](#footnote-ref-11)
12. Historische Entwicklung, „Linux“, unter: https://de.wikipedia.org/wiki/Linux [↑](#footnote-ref-12)
13. Wikipedia, „SourceForge“, unter: https://de.wikipedia.org/wiki/SourceForge [↑](#footnote-ref-13)
14. Moritz Stückler, „Was ist eigentlich GitHub?“, unter: http://t3n.de/news/eigentlich-github-472886/ [↑](#footnote-ref-14)
15. Ebd. [↑](#footnote-ref-15)
16. ITWissen, „GUI (graphical user interface)“, unter: http://www.itwissen.info/GUI-graphical-user-interface-Grafische-Benutzeroberflaeche.html [↑](#footnote-ref-16)
17. Wikipedia, „Versionsverwaltung“, unter: https://de.wikipedia.org/wiki/Versionsverwaltung [↑](#footnote-ref-17)
18. Wikipedia, „Versionsverwaltung“, unter: https://de.wikipedia.org/wiki/Versionsverwaltung [↑](#footnote-ref-18)
19. Sebastian Brockmeyer, „Versionsverwaltung“, unter: https://github.com/UOS-Open-Source-Softwareentwicklung/oss1314/wiki/Versionsverwaltung [↑](#footnote-ref-19)
20. Sebastian Brockmeyer, „Versionsverwaltung“, unter: https://github.com/UOS-Open-Source-Softwareentwicklung/oss1314/wiki/Versionsverwaltung [↑](#footnote-ref-20)
21. Ebd. [↑](#footnote-ref-21)
22. Ebd. [↑](#footnote-ref-22)
23. Ebd. [↑](#footnote-ref-23)
24. Fachadmin, „Client-Server Prinzip“, unter: https://www.fachadmin.de/index.php/Client-Server\_Prinzip [↑](#footnote-ref-24)
25. Vgl. Sebastian Brockmeyer, „Versionsverwaltung“, unter: https://github.com/UOS-Open-Source-Softwareentwicklung/oss1314/wiki/Versionsverwaltung [↑](#footnote-ref-25)
26. Ebd. [↑](#footnote-ref-26)
27. Vgl. Sebastian Brockmeyer, „Versionsverwaltung“, unter: https://github.com/UOS-Open-Source-Softwareentwicklung/oss1314/wiki/Versionsverwaltung [↑](#footnote-ref-27)
28. Springer Gabler Verlag, Gabler Wirtschaftslexikon, unter. http://wirtschaftslexikon.gabler.de/Archiv/569801/repository-v4.html [↑](#footnote-ref-28)
29. Vgl. Sebastian Brockmeyer, „Versionsverwaltung“, unter: https://github.com/UOS-Open-Source-Softwareentwicklung/oss1314/wiki/Versionsverwaltung [↑](#footnote-ref-29)
30. Vgl. Ebd. [↑](#footnote-ref-30)
31. Anmerkung: Die folgenden Ausführungen entstammen meiner eigenen Erfahrung mit GitHub und Git zur Veranschaulichung wurden Screenshots eines Linux (Ubuntu) Betriebssystems benutzt [↑](#footnote-ref-31)
32. Moritz Stückler, „Was ist eigentlich GitHub?“, unter: http://t3n.de/news/eigentlich-github-472886/ [↑](#footnote-ref-32)
33. Anmerkung: Nach jedem Befehl erfolgt die Bestätigung durch die „Enter“-Taste [↑](#footnote-ref-33)