

GNU General Public License 3.0 Vs. Apache License 2.0

Seminararbeit 2022

eingereicht bei

Uni. Prof. Dr. Rony G. Flatscher

**Institut für Wirtschaftsinformatik und Gesellschaft
Wirtschaftsuniversität Wien**

von

Barbara Hruza

E-Mail: barbara.hruza@s.wu.ac.at

Studienrichtung: Wirtschaftsinformatik

Matrikelnummer: 1610380

Eigenständigkeitserklärung für Seminararbeit

Lehrveranstaltungsnummer: 4142

Semester: SS 2022

Lehrveranstaltung: Seminar aus BIS

Lehrveranstaltungsleiter: Uni. Prof. Dr. Rony G. Flatscher

Titel der Arbeit: GNU General Public License 3.0 Vs. Apache License 2.0

Verfasserin: Hruza Barbara

Matrikelnummer: h01610380

Ich versichere / stimme zu:

1. dass ich/wir die Arbeit selbstständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich/uns auch sonst keiner unerlaubten Hilfe bedient habe/n.
2. dass ich/wir dieses Thema bisher weder im In- noch im Ausland (einer Beurteilerin/einem Beurteiler zur Begutachtung) in irgendeiner Form als Prüfungsarbeit vorgelegt habe/n.

Mit der Unterschrift nehme ich/nehmen wir zur Kenntnis, dass falsche Angaben studien- und strafrechtliche Konsequenzen haben können.

30.05.2022

Barbara Hruza

Abstract

Der Inhalt dieser Seminararbeit handelt über die beiden Software Lizenzen GNU General Public License und Apache License. Zu Beginn findet sich ein theoretischer Überblick über Free Software und Open Source Software, sowie auch die Free Software Foundation die maßgeblich in der Entstehung der GNU General Public License ist. Anschließend werden die beiden Lizenzen getrennt näher beschrieben. Im nächsten Abschnitt werden beide Lizenzen auf ihre Kompatibilität analysiert und Anwendungsbeispiele aus der Praxis beschrieben. Abschließend ist eine Zusammenfassung der wichtigsten Eigenschaften zu finden und es werden potenzielle zukünftige Veränderungen diskutiert.

Inhaltsverzeichnis

Abbildungsverzeichnis.....	VI
1 Einleitung	1
2 Freie Software.....	3
2.1 Definition.....	3
2.2 Vorteile von Freier Software	5
2.3 Nachteile von Freier Software.....	6
2.4 Unterschied zu Open Source Software	7
3 Open Source.....	9
3.1 Grundkriterien Open Source Lizenzen.....	9
3.2 Copyleft.....	11
3.2.1 Lizenzen ohne Copyleft Effekt	12
3.2.2 Lizenzen mit strengem Copyleft Effekt.....	12
4 Free Software Foundation	13
4.1 Das GNU Projekt.....	14
5 GNU General Public License.....	15
5.1 Allgemeine Nutzung	15
5.2 Version 3.....	17
5.2.1 Wichtigsten Änderungen GPLv3	19
5.2.2 Ziele	20
5.3 Digital Rights Management (DRM).....	21
6 Apache Software Foundation.....	22
7 Apache License 2.0.....	23
7.1 Definition und Eigenschaften	23
7.2 Limitierung und Haftungen.....	24
7.2.1 Haftungsausschluss	24
7.2.2 Haftungsbeschränkung.....	24
7.2.3 Übernahme von Garantien oder Haftungen	24
8 GNU Affero General Public License	25
9 Vergleich GNU GPL und Apache License	26
9.1 Kompatibilität.....	27

10	Zusammenfassung und Outlook.....	28
	Literaturverzeichnis	30

Abbildungsverzeichnis

Abbildung 1: Prinzip von freier Software [vgl. PFS]	4
Abbildung 2: GNU Logo [vgl. GNU].....	14
Abbildung 3: Logo von GPLv3 [vgl. Gnu].....	17
Abbildung 4: Beziehungen zwischen freien Software Lizenzen und Stärke des Copyleft [vgl. Gpl]	18
Abbildung 5: Logo ASF [vgl. ASF]	22
Abbildung 6: Logo AGPLv3 [vgl. GnuA]	25

1 Einleitung

Was sind Software Lizenzen? Darf ich Software einfach ändern? Darf ich Software einfach kopieren und teilen? Was unterscheidet Software Lizenzen? Was kostet die Verwendung von Software?

All diese Dinge fragt man sich in der heutigen Zeit immer öfter, denn in den letzten Jahren werden wir im täglichen Leben von immer mehr digitalen Funktionen, Programmen und Abläufen begleitet. Sei es im privaten Bereich oder auch in geschäftlicher Hinsicht. Infolgedessen werden die Menschen von dieser weiter fortschreitenden digitalen Entwicklung immer stärker abhängig. Ein Leben ohne digitale Unterstützer wäre heutzutage kaum denkbar, darum werden auch täglich neue oder verbesserte Programme von Software Entwicklern auf dem Markt veröffentlicht. Für diese Vielzahl an unterschiedlichen Software Programmen gilt es in erster Linie für die Nutzer verfügbar zu werden. Diese können entweder frei verfügbar sein oder gegen Entgelt erwerblich sein. Für diese Unterscheidungen sind Regeln festgelegt, die in Form von Lizenzen umgesetzt werden.

Der Fokus dieser Arbeit liegt auf den beiden Software Lizenzen „General Public License“ (GPL) und „Apache License“ (AL). Bevor diese beiden Lizenzen näher erläutert werden, findet sich zu Beginn der Arbeit ein theoretischer Input über Free Software, Open Software und die Free Software Foundation. Diese Begriffe sind sehr wichtig für die oben genannten Lizenzen und unter anderem maßgeblich für das GNU Projekt von Richard Stallman.

Im zweiten Teil der Arbeit wird zuerst auf die einzelnen Software Lizenzen GNU, GPL und AL eingegangen. Es werden die jeweiligen Bestandteile und Eigenschaften näher beschrieben und ebenso auf rechtlichen Aspekte eingegangen. Anschließend wird die Kompatibilität der beiden

Lizenzen diskutiert und jeweils einige praktische Anwendungsbeispiele präsentiert.

Als Abschluss in dieser Arbeit findet sich eine Zusammenfassung der wichtigsten Aspekte beider Software Lizenzen und ein kleiner Ausblick auf potentielle zukünftige Veränderungen.

2 Freie Software

2.1 Definition

Software wird nicht als Freie Software bezeichnet, weil sie kostenlos ist, sondern weil sie Nutzern eine Freiheit bietet. Nach der Definition der Free Software Foundation bedeutet es, dass Nutzer die Freiheit haben die Software auszuführen, zu kopieren, zu verbreiten, zu untersuchen, zu ändern und zu verbessern [vgl. FSF]. Somit liegt die Macht über die Programme bei den Nutzern selbst, wenn dies nicht der Fall ist und Nutzer nicht die Möglichkeit dieser Freiheit haben handelt es sich um unfreie Programme. Um dies genauer zu definieren, gilt ein Programm als Freie Software, wenn es folgende vier Freiheiten gleich erfüllt [vgl. FSF]:

- „Freiheit 0: Die Freiheit das Programm so auszuführen wie man möchte und auch für jeden Zweck.
- Freiheit 1: Die Freiheit, die Funktionsweise des Programmes zu untersuchen und den eigenen Bedürfnissen anzupassen. Zugang zum Quellcode ist dafür Voraussetzung.
- Freiheit 2: Die Freiheit, das Programm neu umzuverteilen, um damit Mitmenschen helfen zu können.
- Freiheit 3: Die Freiheit, das Programm zu verbessern und diese Verbesserungen der Öffentlichkeit freizugeben, damit die gesamte Gesellschaft davon profitiert. Zugang zum Quellcode ist dafür Voraussetzung“ [vgl. FSF].

Damit diese vier Freiheiten tatsächlich erfüllt sind, müssen sie laut Richard Stallman¹ unwiderruflich sein, solange man als Nutzer keinen

¹ Gründete 1983 die Freie-Software-Bewegung & 1985 die Free Software Foundation [vgl. STA]

Fehler macht, andernfalls ist die Software nicht frei. Es gibt jedoch gewisse Regeln, wie Copyleft, die mit den Freiheiten vereinbar sind und diese sogar schützen. Copyleft (siehe genaueres in Kapitel 3.2) sorgt durch seine Einschränkungen dafür, dass man als Nutzer keine Beschränkungen hinzufügen kann, die anderen Nutzern die Verwendung verweigern könnten [vgl. Stall, 2002, S.44].

Abbildung 1 zeigt das Prinzip von freier Software. Es wird dargestellt welche einzelnen Komponenten benötigt werden und wie diese miteinander arbeiten und verbunden sind.

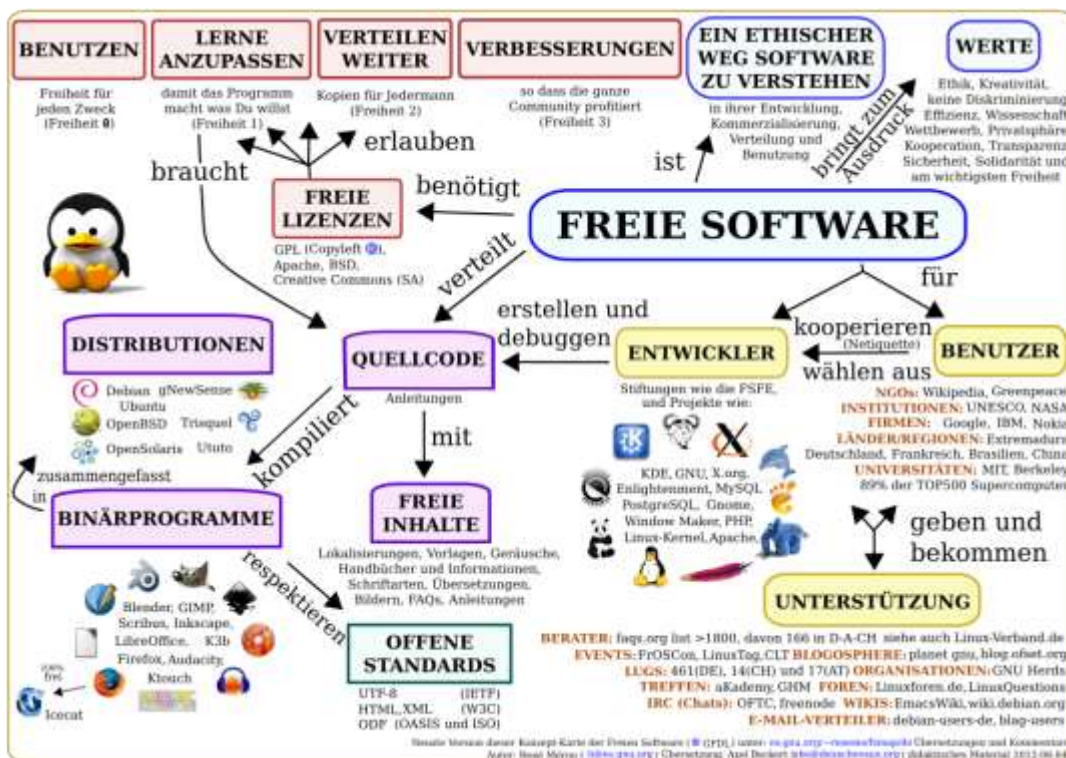


Abbildung 1: Prinzip von freier Software [vgl. PFS]

Weiters ist noch zu erwähnen, dass Freie Software nicht in den Bereich der nicht kommerziellen Nutzung² fällt. Denn ein Ziel von freien Pro-

² CC German Chapter definiert: „Eine nicht kommerzielle Nutzung ist eine Nutzung die „nicht vorrangig auf einen geschäftlichen Vorteil oder eine geldwerte Vergütung“ gerichtet ist.“ [vgl. CCG]

grammen ist es für die Nutzung, die Entwicklung und den Vertrieb verfügbar zu sein. Der Grund ist, dass Nutzer möglicherweise für Kopien einer freier Software etwas gezahlt haben oder unentgeltlich erworben haben. Aber auch auf der andere Seite die Möglichkeit besteht Kopien selber zu verkaufen [vgl. FSF].

2.2 Vorteile von Freier Software

Ganz allgemein ist festzustellen, dass Freie Software kostengünstiger ist oder sogar vollständig frei verfügbar, aber vor allem gibt es keine Lizenzgebühr die bei der Nutzung anfällt. Durch diese Aspekte wird es Nutzern ermöglicht erstens, legal Kopien zu erstellen und zweitens die Software auf unterschiedlich vielen Computern zu verwenden und das jederzeit. Dieser leichte Erwerb kann insbesondere bei einer großen Anzahl von Nutzern zu einer schnellen Weiterentwicklung beitragen, denn bei regelmäßigem Feedback, können Fehler schneller korrigiert werden und so die Sicherstellung einer reibungslosen Funktionalität sichergestellt werden [vgl. MRD].

Im Vergleich zu proprietärer Software spricht vor allem die Freiheit für die Nutzer, die sie bei der Verwendung haben. Die Software ist auf keinen Hersteller beschränkt und jeder kann diese im vollen Umfang nutzen [vgl. FOS]. Durch die uneingeschränkte Nutzung ist es den Nutzern auch möglich die Software zu ändern und auf das jeweilige Bedürfnis anzupassen. Somit können eigene und vielfältige Ideen schnell und einfach umgesetzt werden. Ein weiterer Vorteil ist die Sicherheit und die Stabilität des Quellcodes, denn durch die freie Verfügbarkeit agieren Nutzer auch als Co-Entwickler, die die Software verbessern und Fehler beheben können. Aus dieser Motivation resultiert eine ständige Weiterentwicklung meist durch die Nutzer selbst [vgl. Vij, S.17, 2014].

Einen weiteren Vorteil der durch die freie Verfügbarkeit entsteht ist ein synergetischer Effekt, durch diesen ähnliche Unternehmen die dieselbe Software benutzen voneinander profitieren können. Damit wird gemeint, dass einzelne Unternehmen von der Weiterentwicklung der Software anderer Unternehmen profitieren können und so können die wechselseitigen Verbesserungen allen Beteiligten zugutekommen [vgl. FOS].

2.3 Nachteile von Freier Software

Einen großen Nachteil in der freien Software sehen die Entwickler in der möglichen Zerteilung von Programmcodes. Es kann dazu kommen, dass mehrere Entwickler Verbesserungen am Programmcode vornehmen, aber nicht jede Änderung ist mit jeder verbesserten Version kompatibel. Folge dessen sind mehrere Versionen von demselben Programmcode verfügbar, was schnell unübersichtlich werden kann, gerade für Anfänger oder Amateure [vgl. Webb, 2008, S.30]. Auf der anderen Seite ist Freie Software abhängig von einer großen und vor allem aktiven Community, die für funktionierende Programme und den notwendigen Fortschritt sorgen. Denn wenn keine Programmierer an der Software arbeiten, ist auch die Entwicklung und die Weiterentwicklung unsicher [vgl. MRD].

Ein weiterer Nachteil ist die mangelnde Verfügbarkeit für Unterstützung, denn oft steht hinter einer freien Software nur ein kleines Entwicklerteam. Dadurch steigt das Risiko, dass Nutzer keine oder nur schlechte Möglichkeiten haben die jeweiligen Personen oder Organisationen nach Hilfe zu fragen [vgl. Mar, 2013, S.17].

Die meisten Probleme bilden sich bei freier und Open Source Software durch das Fehlen einer Garantie oder Haftung. Wenn es an der Software zu Beschädigungen kommt oder diese einen Schaden anrichtet, gibt es keine Person die für die fehlerhafte Software legal verantwortlich ge-

macht werden kann. Durch gesetzliche Regelungen kommt es in einigen Ländern dazu, dass Open Source Software nur als Geschenk betrachtet wird, um so den vorherrschenden Gesetzen bezüglich Gewährleistung zu entsprechen [vgl. Pen, 2005, S.23].

2.4 Unterschied zu Open Source Software

„Die Begriffe Freie Software und Open Source stehen für fast die gleiche Programmvielfalt. Sie sagen jedoch grundlegend unterschiedliche Dinge über diese Programme, die auf unterschiedlichen Werten basieren.“

[Richard Stallman, FSF1]

Beide Software Konzepte ermöglichen ihren Nutzern einen freien Zugang zu dem Quellcode und weiters ordnen sie dieselben Copyleft und freizügigen Softwarelizenzen als frei beziehungsweise offen ein. Der ausschlaggebende Unterschied besteht in der Sichtweise und der Werte. Auf der einen Seite wird der Fokus bei freier Software auf die Nutzerkontrolle gelegt und Software als sozial und politisch angesehen, somit liegt der Fokus auf dem Benutzer selbst [vgl. FS1]. Auf der anderen Seite bezieht sich Open Source Software auf den Allgemeinnutzen einer freien Softwareinfrastruktur, eines freien Softwaremarktes und gemeinsamen Entwicklungsprozess, was wiederum die Entwickler in den Mittelpunkt stellt [vgl. OSI][vgl. FS1]. Freie Software sieht alle Programme die nicht frei sind als ein soziales Problem und dieses kann nur gelöst werden indem man unfreie Software nicht mehr verwendet. Hingegen beschäftigt sich Open Source mit der Frage der Weiterentwicklung der Software und sieht unfreie Software als unterlegene Lösung im praktischen Sinn [vgl. FSF1].

Im Gegensatz zu den vier Freiheiten für Freie Software, müssen Lizenzverträge unter der Open Source Initiative (OSI) die Open Source Defini-

tion (siehe genaueres Punkt 3.1.) erfüllen und den folgenden drei Kriterien entsprechen:

- Der Quelltext liegt den Nutzern in einer lesbaren Programmiersprache offen vor
- Open Source Software hat keine Beschränkungen hinsichtlich der Anzahl der Nutzer und der Reproduktion der Software
- Open Source Software darf ohne Lizenzgebühren verändert weitergegeben werden. Durch offene Quelltexte ist die Weiterentwicklung erwünscht [vgl. OSI]

Aus diesen unterschiedlichen Werten der beiden Konzepte zeigt sich, dass Open Source Kriterien lockerer sind und die meisten die Voraussetzungen für Freie Software erfüllen, doch es gibt praktische Unterschiede. Einer ist die zu starke Einschränkung der Open Source Lizenz, denn hier werden durch die nicht mögliche Umgestaltung einer Version nicht alle Freiheiten erfüllt. Eine weitere Ausnahme sind zu schwache Lizenzen ohne Copyleft beim Quellcode, wodurch ebenso unfreie Kriterien enthalten sind. In der Praxis wird das für den Nutzer durch Prüfsignaturen in den Programmen sichtbar, die dazu führen das es Nutzern nicht möglich ist abgewandelte Version zu verwenden. Somit wird nach der Definition der Stiftung für Freie Software eine Freiheit nicht erfüllt und ist somit unfrei [vgl. FSF1].

3 Open Source

Open Source wurde 1998 als neuer Begriff von den Gründern der Open Source Initiative (OSI) Eric S. Raymond, Bruce Perens und Tim O´Reilly implementiert, um den Begriff „Freie Software“ zu ersetzen und einen besseren Marketingeffekt zu erreichen. Der Begriff sollte geschäftsfreundlicher und attraktiver auf Unternehmen wirken [vgl. FS1]. Das Ziel war den Fokus weg von dem vermeintlichen Preis zu legen und hin zu einem klaren Begriff, der die freie Verfügbarkeit des Quellcodes äußert. Die Gründer wollten so deutlicher machen, dass Open Source durch den freien Zugang und die offene Verwendung flexibler ist und nicht an die Entwickler gebunden ist. Somit wurde durch die OSI Open Source als Oberbegriff für Freie Software festgelegt, der bis heute als Standard betrachtet wird [vgl. Red].

Um als Open Source zu gelten, müssen immer alle 10 Grundkriterien aus dem nächsten Punkt erfüllt sein. Sie müssen sowohl für das Originalprogramm erfüllt sein, als auch für alle abgeleiteten Versionen [vgl. Per, 1999] .

3.1 Grundkriterien Open Source Lizenzen

Erstes Kriterium ist die freie Weiterverbreitung von Softwarekomponenten, in Form eines Verkaufs oder einer Schenkung, ohne dafür eine entgeltliche Gebühren leisten zu müssen. Nutzer haben die Freiheit die Software zu vervielfachen und zu verkaufen ohne etwas dafür zu bezahlen [vgl. Per, 1999].

Zweitens, muss der Quellcode des Produkts in einer modifizierbaren Form enthalten sein, wenn nicht muss er zumindest durch einen gebüh-

renlosen Download verfügbar sein. Der Quellcode muss für den Nutzer verständlich sein und leicht zu bearbeiten [vgl. Nrf].

Drittens, muss es erlaubt sein jede abgeänderte Version unter denselben Voraussetzungen wie in der Originallizenz zu verteilen [vgl. Per, 1999].

Viertes Kriterium ist die Sicherstellung der Integrität des Quellcodes des Entwicklers. Das bedeutet es muss speziell genehmigt sein geänderte Versionen weiterzugeben. Entweder unter einem neuen Namen oder in Form eines Patch neben dem originalen Quellcode [vgl. Per, 1999].

Fünftens, darf keine Diskriminierung von bestimmten Personen oder Gruppen existieren. Die Software muss von jedem in der gleichen Weise verwendet werden können [vgl. Per, 1999].

Das siebte Kriterium besagt, dass jeder dieselbe Lizenz erhält. Sie soll automatisch mit der Verwendung der Software erteilt werden, ohne dass zusätzliche Anmeldungen erforderlich sind [vgl. Per, 1999].

Achtens, darf die Lizenz nicht produktspezifisch sein. Verteilte Programme sollen weiterhin allen Nutzern dieselben Rechte geben, unabhängig von der Verbindung zur Originalen Softwareverteilung [vgl. Per, 1999].

Neuntes Kriterium ist die freie Verwendung anderer Software die gemeinsam mit der lizenzierten Software verwendet wird. Es dürfen keine Beschränkungen auferlegt werden die die Verwendung einschränken würde [vgl. Per, 1999].

Das letzte Kriterium stellt sicher, dass die Lizenz auf keiner bestimmten Technologie beruht, sondern unterschiedliche Arten der Verteilung erlaubt [vgl. OSI1].

3.2 Copyleft

Copyleft wird verwendet, um Softwareprogrammen die Freiheiten zu geben auch alle abgeänderten Versionen des Programms frei zu machen. Die einfachste Methode, um Programme als frei zu bezeichnen ist sie in die Public Domain – ohne Urheberrecht - zu lassen. Dadurch darf die Software mit allen Änderungen geteilt werden, aber Nutzer bekommen so auch die Möglichkeit das Programm zu proprietärer Software zu ändern. Nutzer dieser Version haben dann nicht mehr die Freiheit, die der Entwickler beabsichtigte [vgl. FSF2]. Darum wurde Copyleft von Richard Stallman und der Free Software Foundation im Zuge des GNU Projekts entwickelt, um so allen Nutzern die Freiheiten zu gewähren [vgl. Mus, 2002, S.102].

Um ein Programm unter Copyleft zu stellen, muss der Entwickler zunächst selbst das jeweilige Programm urheberrechtlich durch Copyright schützen und dann die General Public Lizenz unterzeichnen. Die sorgt dafür, dass jeder weitere Nutzer die Software unter den gleichen Bedingungen verwenden, bearbeiten und weitergeben darf, einschließlich der von ihm vorgenommenen Modifizierungen. Durch dieses Verfahren ist die Software für alle frei verfügbar, aber es ist Nutzern nicht möglich sie unter privates Eigentum zu stellen [vgl. Mus, 2002, S. 101].

Da die unterschiedlichen Bedürfnisse der Lizenzgeber erfüllt werden sollen, wurden diverse Open Source Lizenzen entwickelt, die diese Freiheiten unterschiedlich streng einschränken. Die Einschränkung nach der Strenge wird als Copyleft Effekt bezeichnet. Durch diesen Copyleft Effekt wird geregelt, dass Software die bearbeitet wurde nur unter der Originallizenz geteilt und veröffentlicht werden darf [vgl. Nrf]. Das soll bewirken, dass alle Nutzer von den Veränderungen und Verbesserungen der Software profitieren [vgl. OSL].

In den nächsten zwei Punkten wird näher auf zwei spezifische Copyleft Effekte mit unterschiedlicher Strenge eingegangen die für die GNU GPL und Apache Lizenz relevant sind.

3.2.1 Lizenzen ohne Copyleft Effekt

Wenn eine Lizenz keinen Copyleft Effekt vorweist, dann bedeutet dies, dass der Nutzer alle Freiheiten der Lizenz hat und somit keine Beschränkungen hinsichtlich der Verwendung vorgeschrieben bekommt. Es besteht die Wahl, wie man individuell veränderte Software weitergeben möchte. Damit ist es dem Nutzer möglich für Versionen verschiedene Lizenzbedingungen zur Weitergabe zu nutzen, sogar die Änderung auf proprietäre Software ist zulässig [vgl. Ifr]. Dies bedeutet auch, dass Nutzer für die abgeänderte Version beziehungsweise den abgeänderten Teil keinen Quellcode mehr offen legen müssen.

Unter diese Kategorie fällt unter anderem die Berkeley Software Distribution-Lizenz und die Apache License 2.0 [vgl. Nrf].

3.2.2 Lizenzen mit strengem Copyleft Effekt

Bei Lizenzen mit strengem Copyleft Effekt wird der Nutzer dazu verpflichtet sämtliche Änderungen oder abgeänderte Versionen nur unter die Lizenz der Original Software inklusiver derer Bedingungen weiterzugeben [vgl. Ifr]. Daraus resultierend können andere Nutzer den Quellcode weiter abändern und die Software bleibt weiterhin für den Nutzer frei verfügbar, trotz Veränderungen [vgl. Nrf].

Unter diese Kategorie fallen sämtliche GPL Lizenzen, wie beispielweise die Affero General Public License und die GNU General Public License Version 1 bis 3 [vgl. Ifr].

4 Free Software Foundation

Die Free Software Foundation wurde 1985 von Richard Stallman gegründet, um die Freiheit von Softwareprogrammen zu fördern und Gelder für die Entwicklung zu sammeln [vgl. Fsf][vgl. FSF3]. Ziel dieser Stiftung ist es die Freiheit von Softwareprogrammen für alle Nutzer sicherzustellen, indem sie nur Freie Software, in Form des GNU Betriebssystems nutzen. Bis heute zählt die Stiftung zu einer der erfolgreichsten in der Computerwelt, die von einer weltweiten Community von Programmierern angetrieben wird [vgl. Fsf1].

Die Hauptaufgaben der Free Software Foundation sind einerseits die Pflege der Philosophie von Freier Software, damit klar ersichtlich ist was unter Freie Software fällt. Andererseits wird das GNU Projekt (siehe Punkt 4.1) von der Stiftung gesponsort, um sicherzustellen das ein funktionierendes Betriebssystem für Freie Software verfügbar ist. Ein weitere Aufgabe der Free Software Foundation ist die Sammlung und Haltung von Urheberrechtsübertragungen von Entwicklern und Unternehmen, um so Freie Software davor zu schützen proprietär zu werden [vgl. Fsf1].

Zusammenfassend kann gesagt werden, dass sich die Stiftung für die Einführung neuer Freier Software einsetzt und gegen den Ausbau von proprietärer Software arbeiten. Gefahren mit denen sich die Free Software Foundation auseinandersetzen muss sind Digital Restrictions Management (DRM), Softwarepatente und tückische Datenverarbeitung [vgl. Fsf1].

4.1 Das GNU Projekt

Im Zuge des GNU Projekt wurde unter Richard Stallman ein unixähnliches freies Betriebssystem, namens GNU (Gnu's Not Unix) entwickelt. Da es mit Unix³ kompatibel ist, ist es für Nutzer leichter von Unix zu GNU zu wechseln [vgl. Stall, 2002, S. 19]. Anstoß gaben Stallman die Veränderungen in den 1970er/1980er Jahren, wo Software



Abbildung 2: GNU Logo [vgl. GNU]

nur unter stark eingeschränkten Lizenzen von Unternehmen veröffentlicht wurden und der Quelltext nicht mehr freigegeben wurde [vgl. GNU]. Sein Ziel war es den Geist der Zusammenarbeit in der Computwelt wieder zurückzubringen und alle Hindernisse von proprietärer Software aus dem Weg zu räumen [vgl. FSF3]. Obwohl kaum technische Vorteile zwischen GNU und dem Unix System zu finden sind, bietet GNU einen sozialen und ethischen Vorteil, in dem es Nutzern erstens erlaubt ist zusammenzuarbeiten und zweitens jeder die Freiheiten des anderen respektiert [vgl. Stall, 2002, S. 24].

Um mit der Entwicklung der GNU Software zu starten, kündigte Stallman 1984 seinen Job am MIT [vgl. Stall, 2002, S. 20]. Stallman startete mit dem Schreiben eines freien Betriebssystem wo folgende einzelne Programme enthalten sein mussten: Betriebssystemkern, Compiler, Editoren, Textverarbeitung, E-Mail Software, grafische Oberflächen und noch einige mehr. In den darauffolgenden Jahren waren alle notwendigen Programme geschrieben worden oder von bereits existierenden

³ Nähere Informationen siehe:

<http://www.netzmafia.de/skripten/unix/unix1.html>

Programmen übernommen. Bis 1991 dann schließlich die letzte Komponente, der Betriebssystemkern, von Linux entwickelt wurde. Die Verbindung vom restlichen GNU System mit der Linux Komponente führte zu einem vollständigen System GNU/Linux [vgl. FSF3]. Doch auch schon während dem gesamten Entwicklungsprozess wurden die einzelnen Systeme, die auf dem Unix System umgesetzt wurden, immer beliebter und von Nutzer weiterentwickelt. Wodurch das gesamte Programm viel Stärker wurde als anfänglich erwartet [vgl. Stall, 2002, S.27].

Durch die Entwicklung von GNU ist es Nutzern möglich Softwaresysteme frei zu erhalten. Jeder kann sich somit Kosten sparen, aber vor allem Zeit. Denn anstatt Kopien von System zu erstellen, kann dieser Aufwand in die Weiterentwicklung gesteckt werden. Folglich können Nutzer auch individuelle Änderungen selbst durchführen oder von professionellen Programmierern ändern lassen [vgl. Stall, 2002, S.36].

5 GNU General Public License

5.1 Allgemeine Nutzung

Die GNU General Public License (GNU GPL) ist die mit am Abstand am häufigsten verbreitetste Lizenz von freier Software und gleichzeitig repräsentiert sie Lizenzen mit einem strengen Copyleft Effekt [vgl. Qua]. Das strenge Copyleft wird bei GNU GPL genutzt, damit alle Freiheitsrechte erhalten bleiben, auch bei veränderten Versionen [vgl. Gnu].

1989 veröffentlichte Richard Stallman die erste Version der GNU GPL mit dem Ziel diese bei unterschiedlichen Projekten anwenden zu können. Sie ist somit die erste Lizenz mit Copyleft für die allgemeine Verwendung [vgl. Gnu].

Die GNU GPL ist dafür ausgelegt dem Nutzer die Freiheit zu garantieren die Freie Software zu ändern und zu teilen. Sie stellt sicher das Nutzer den Quellcode erhalten, um diesen individuell bearbeiten zu können oder Teil aus diesem für andere freie Programme zu verwenden. Damit diese Freiheiten auch für jeden gewährleistet werden, schützt GNU GPL diese in zwei Schritten: die Originalsoftware wird urheberrechtlich geschützt und die Lizenz erlaubt Nutzern eine legale Bearbeitung dieser Software. Wichtig ist jedoch das GNU GPL keine Garantie für die Freie Software bietet, denn Nutzern soll bewusst sein, dass sie veränderte Versionen verwenden können die nicht vom Entwickler selbst stammen [vgl. Stall, 2002, S.199].

Zu unterscheiden ist auch, ob die GPL lizenzierte Software privat oder kommerziell verwendet wird. Wird die Software nur für den Privatgebrauch verwendet, ohne diese weiterzugeben, dann muss der Quellcode nicht offengelegt werden – es ist nur verpflichtend wenn diese mit anderen Nutzern geteilt wird und der Copyleft in Kraft tritt. Bei einem kommerziellen Vertrieb darf die Software entweder gegen ein Entgelt oder kostenlos vertrieben werden. Hier liegt auch der größte Unterschied zu anderen Software Lizenzen. In der General Public License ist festgelegt, dass alle GPL Softwares zu jedem Preis vertrieblich sind. Wo hingegen normale Software Lizenzen einen Vertrieb dieser Art untersagen [vgl. Gnu].

5.2 Version 3

Gründer des GNU Projekts Richard Stallman veröffentlichte 1989 die erste Version von GNU GPL. Knappe zwei Jahre später folgte die zweite Version. Die wesentlichen Änderungen liegen in der „Liberty or Death“ Klausel die besagt, dass es bei Nichterfüllung von einigen GNU GPL Kriterien verboten ist die Software zu verbreiten und in der Klausel 8 die eine geografische Abgrenzung für die Gültigkeit der Lizenz regelt. Die dritte Version GPLv3 folgte dann im Jahr 2007 [vgl. Gnu]. Grund für die Weiterentwicklung waren der technische und rechtliche Fortschritt und das Bestreben nach einer stärkeren Berücksichtigung von anderen Urheberrechtssystemen [vgl. Nrf].



Abbildung 3: Logo von GPLv3 [vgl. Gnu]

In Abbildung 4 sind die unterschiedlichen Kategorien von Software Lizenzen im Zusammenhang mit der Stärke des Copyleft Effekts zu sehen. Es wird dargestellt mit welchen Lizenzen die dritte Version von GPL kompatibel ist. Die Pfeile zeigen mit welcher Lizenz die jeweilige Lizenz kompatibel ist, was auch durch mehrere aufeinanderfolgende möglich ist. So ist beispielweise ISC-Lizenz mit der Version 3 von GPL kompatibel. Bedeutet fast alle Lizenzen die mit GPLv2 kompatibel sind, sind es auch mit der dritten Version. Aber auch neue Kompatibilitäten haben sich gebildet, wie die Apache License 2.0. [vgl. Gpl]. Durch den strengen Copyleft Effekt bei GPL ist jedoch zu beachten, dass eine Kombination von GPL lizenzierter Software und Software die gewisse Einschränkungen in der Freiheit hat nicht zulässig sind [vgl. Gnu].

Die GNU Affero GPL (AGPL) wurde durch die Einführung der dritten Version mit der GPLv3 kompatibel. Durch die dritte Version wurde die AGPL auf die Anwendung aller netzwerkinteraktiven Software erweitert und ist

somit nicht nur flexibler, sondern der Quellcode muss auch nur noch wie in der GPL verlangt offen gelegt werden, wenn keine Netzwerkschnittstelle verwendet wird. Durch diese Zusammenarbeit ist es Nutzern möglich ihren Copyleft zu stärken und gleichzeitig noch am GPL Code aufzubauen [vgl. Gpl].

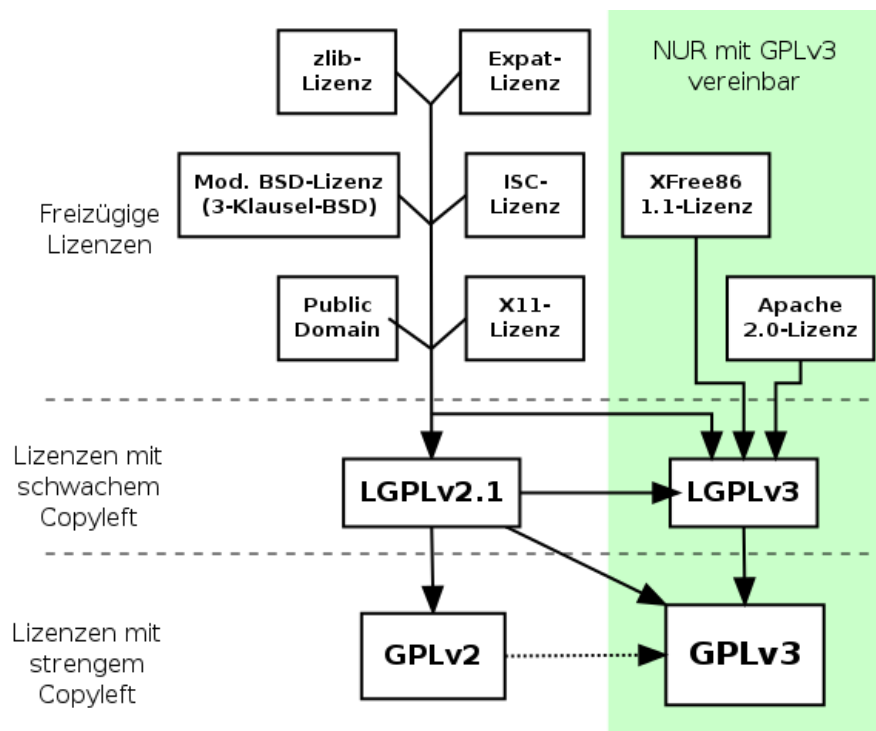


Abbildung 4: Beziehungen zwischen freien Software Lizenzen und Stärke des Copyleft [vgl. Gpl]

5.2.1 Wichtigsten Änderungen GPLv3

Von der zweiten auf die dritte Version gab es sechs wichtige Änderungen. Diese werden in den nächsten Absätzen näher erklärt.

In der dritten Version ist nun eine geregelte Patent Lizenz enthalten, die verlangt, dass alle Personen die eine GPL Software mit Patent weitergeben auch den Empfängern die volle Nutzung dieser Patente ermöglichen. Durch eine zusätzliche Regelung wurden auch Fälle der Lizenzverletzung erweitert. Die Rechte des Lizenznehmers fallen nicht wie bisher komplett weg wenn er die Rechte verletzt, sondern er kann seine Rechte wiederherstellen indem er sich wieder rechtskonform verhält [vgl. Nrf].

Durch Ziffer 14 kann der Entwickler entscheiden unter welcher Version von GPL er seine Programm freigeben möchte. Wenn der Entwickler sich nicht äußert, hat der Nutzer das Wahlrecht sich für eine GPL Version zu entscheiden [vgl. Nrf]. Laut Black Duck⁴ ist seit der Veröffentlichung der dritten Version, trotzdem die zweite Version mit 42,5% die meist verwendete Lizenz [vgl. Gnu].

In Ziffer 7 wird der strenge Copyleft aufgelockert, wodurch Nutzer mehr Freiheiten bekommen, da nun auch Lizenzen mit abgeschwächtem Copyleft Effekt anerkannt werden. Auf der anderen Seite werden dadurch aber auch Lizenzen mit zusätzlichen Pflichten erlaubt. Diese weiteren Pflichten können beispielweise Gewährleistungsregelungen sein und müssen sich im Pflichtenkatalog in Ziffer 7 wiederfinden. Unternehmen können beim Handel von eigener und GPL Software strengere Gewährleistungs- und Haftungsbestimmungen dazunehmen. Besonders ist jedoch, dass die strengeren Bestimmungen nur für die neuen

⁴ Nähere Informationen siehe: <https://www.ccpsoft.de/black-duck-hub/>

Softwareteile gelten, da die alten Bestandteile noch den Regelungen der GPLv3 folgen [vgl. Nrf].

Eine neue Regelung ist die Einführung einer Digital-Rights-Management (DRM) Regelung. Diese wird in Ziffer 3 festgehalten und besagt, dass GPL-Software nicht für DRM genutzt werden darf. Grund dafür sind die Einschränkungen der Nutzungsrechte durch DRM.

Weiters wird in Ziffer 13 die Kompatibilität mit der AGPL festgelegt, wodurch es möglich ist, dass bei Verbindung mit GPL-Software die Lizenzbedingungen der AGPL unverändert bleiben [vgl. Nrf].

5.2.2 Ziele

Durch die Entwicklung der Version 3 soll der Copyleft weiter geschützt werden und Nutzer sollen vor allem vor folgenden Bedrohungen geschützt werden.

Zum einen ist das die Tivoisierung. Hier werden Geräte so manipuliert, dass Freie Software von Unternehmen bearbeitet werden können, aber Nutzer keine Änderungen mehr vornehmen können [vgl. Gpl].

Zum anderen existieren Gesetze die das Schreiben und Teilen von Software illegal machen. Darunter fallen zum Beispiel der US-Digital Millennium Copyright Act und das digitale Restriktionsmanagement [vgl. Gpl].

Eine weitere Bedrohung sind diskriminierende Patentgeschäfte. So versucht beispielweise Microsoft Lizenzgebühren für die Nutzung von freier Software zu erhalten, indem sie den Nutzern von freier Software sagen, dass sie nicht verklagt werden solange sie Software verwenden deren Anbieter dafür eine Gebühr an Microsoft zahlt [vgl. Gpl].

All diese Maßnahmen führen zu einem besseren Copyleft und schützen die Freiheiten der Nutzer. Durch die Änderungen will GPv3 den Nutzern

von freier Software eine leichte und interaktive Zusammenarbeit gewährleisten [vgl. Gpl].

5.3 Digital Rights Management (DRM)

Hierbei handelt es sich im weiten Sinne um einen Kopierschutz, der eine Verbreitung an dritte verhindern soll. Dadurch wird die Verwendung auf den jeweiligen Käufer beschränkt, denn nur auf dem Gerät des Käufers kann das entsprechende Programm ausgeführt werden [vgl. Mum].

Somit stellt DRM eine große Herausforderung für Nutzer von digitalen Medien dar. Jedoch bietet es einen gewissen Schutz des Urheberrechts, denn mit der einfachen Übertragung und Duplizierung von digitalen Daten ist die Gefahr von rechtlichen Verstößen deutlich gewachsen [vgl. Ian].

Durch DRM kann die Nutzung auf verschiedene Arten für die Nutzer eingeschränkt werden. Es kann verhindert werden, dass Programme bearbeitet, geteilt, mitgeschnitten oder gespeichert werden, auch die Einschränkung auf bestimmte IP-Adressen oder Standorte ist möglich. Doch am häufigsten wird ein Computercode in das Programm eingebaut, der den Zugriff beschränkt. Nur ausgewählte Systeme können diesen Code lesen und die Einschränkungen nach Bedarf festlegen [vgl. Byn].

In der Praxis werden meistens virtuelle private Netzwerke, Softwarelizenzen, IP-Authentifizierungsprotokolle, Proxy-Server, etc. verwendet [vgl. Byn].

In Bezug auf Freie Software erklärt der Gründer der Free Software Foundation Richard Stallman, dass DRM das Funktionieren von Software verweigert und um diese Freiheiten zu schützen müsste man DRM aufhalten. Somit können Programmen mit DRM nur ausgeführt werden,

wenn keine Freiheit zur Veränderung gegeben ist. Umgangen wird das von Herstellern, die lizenzierte Freie Software für DRM Systeme verwenden, indem Funktionen verweigert werden, wenn Veränderungen an der Software vorgenommen werden. Daraus lässt sich schließen, dass es für Nutzer kaum möglich ist Veränderungen tatsächlich ausführen zu können [vgl. DRM].

6 Apache Software Foundation

Die Apache Software Foundation wurde 1999 der Nachfolger der Apache Group. Sie ist eine ehrenamtliche Organisation die sämtliche Apache Softwareprojekte unterstützt.



Abbildung 5: Logo ASF [vgl. ASF]

Ein Merkmal für alle Apache Projekte ist einerseits der gemeinsame Entwicklungsprozess und andererseits die offene Apache Lizenz. Für die Stiftung arbeiten mehrere ausgewählte Entwicklerteams zusammen, um aktiv neue Projekte zu bearbeiten [vgl. ASF].

Damit ein Projekt unter Apache fällt muss zuerst ein Grant- oder Contributorvertrag von der Apache Software Foundation lizenziert werden. Im Zuge dessen werden die erforderlichen Eigentumsrechte für die Entwicklung und die folgende Verbreitung sichergestellt [vgl. ASF].

Wenn ein Projekt dann von der ASF angenommen wurde, kommt es weiter in den sogenannten Brutkasten. Bei Erfolg wird das Projekt zu einem Toplevel Projekt aufgestuft. Bekannte Apache Toplevel Projekte sind Hadoop, Cassandra oder Tomcat [vgl. ASF1].

7 Apache License 2.0

7.1 Definition und Eigenschaften

Die Apache License 2.0 wurde 2004 von der Apache Software Foundation veröffentlicht und ist eine anerkannte Freie Software Lizenz [vgl. Apa].

Bei dieser Lizenz handelt es sich um freizügige Lizenz, die nur sehr wenige Beschränkungen hat und keinen Copyleft Effekt vorweist. Daraus folgt, dass es hinsichtlich der Nutzung und Verteilung der lizenzierten Apache Software kaum Verpflichtungen gibt. Nutzer können selbst entscheiden unter welche Lizenz sie ihre veränderten Softwareteile stellen wollen und es ist ihnen sogar erlaubt sie als proprietäre Software zu lizenzieren. Der originale Teil der Software bleibt weiterhin unter Apache 2.0. [vgl. OSA].

Eine Besonderheit im Vergleich zu anderen Non Copyleft Lizenzen ist eine vollständige Auflistung aller Änderungen die an der ursprünglichen Software gemacht wurden. Eine weitere Unterscheidung ist eine konkrete Regelung für den Umgang mit Patenten. Lizenzgeber dürfen keine zusätzlichen Lizenzzahlungen auf patentierte Software verlangen die unter Apache 2.0 läuft [vgl. OSA].

Eine Software die unter Apache 2.0 lizenziert ist, darf nur verbreitet, verändert und benutzt werden wenn folgende Kriterien erfüllt sind:

- Nutzer müssen eine Kopie des Lizenztextes erhalten
- Veränderte Software muss explizit gekennzeichnet sein
- Alle rechtlichen Vermerke müssen in der Originalform erhalten bleiben

- Wenn in der Software eine NOTICE Datei enthalten ist muss ein Copyright Vermerk enthalten sein. Dieser kann in der NOTICE Datei, im Quelltext oder innerhalb einer generierten Anzeige enthalten sein [vgl. OSA]

7.2 Limitierung und Haftungen

7.2.1 Haftungsausschluss

Das Programm wird vom Lizenzgeber als „AS IS“ bereitgestellt, sofern es das geltende Recht nicht vorschreibt, jedoch ohne jegliche Garantien oder Bedingungen. Es gibt weder ausdrückliche noch stillschweigende Vereinbarungen, denn Nutzer sind alleine für die korrekte Verwendung der Programme verantwortlich und somit auch für Risiken die mit der Verwendung einhergehen [vgl. Apa].

7.2.2 Haftungsbeschränkung

In jedem Fall haftet man als Nutzer selber für seine Handlungen. Mitwirkende haften in keinem Fall für unerlaubte Handlungen, direkten oder indirekten Schäden oder Folgeschäden die bei der Nutzung des Programmes durch den Nutzer entstehen könnten [vgl. Apa].

7.2.3 Übernahme von Garantien oder Haftungen

Wenn man als Nutzer das Programm oder Teile davon weitergeben möchte kann man sich dazu entscheiden eine Gebühr für die Annahme von Support, für eine Garantie oder für eine Entschädigung zu erheben. Für diese Bereitstellungen darf der Nutzer nur im eigenen Namen handeln und es muss mit den Rechten der Lizenz kompatibel sein [vgl. Apa].

8 GNU Affero General Public License

Die Affero General Public License (AGPL) fällt auch unter die Freie Software Lizenzen und basiert auf der General Public License Version 3 (GPLv3). Die AGPL enthält als modifizierte Version eine Besonderheit, denn es wird für Entwickler möglich veränderte Programme von anderen, die auf einem Webserver ausgeführt werden, zu verwenden. Es ist also anzunehmen, dass ein Entwickler der ein Programm auf seinem eigenen Webservern bearbeitet und ausführt niemals als Kopie für andere freigibt. Durch die GNU AGPL wird dafür gesorgt, dass jeder die bearbeitete Version des Entwicklers auf einem Webserver der für alle verfügbar ist verwenden kann. Einzige Voraussetzung ist, dass alle Bedingungen der Lizenz eingehalten werden, insbesondere die zur Verfügungstellung eines Downloads des Quellcodes [vgl. FsA].

Somit stellt die AGPL eine Erweiterung des Abschnitt 13 im GPL Lizenztext dar, die dafür sorgt, dass das ASP-Schlupfloch beseitigt wird. Unter ASP Schlupfloch versteht man die Lücke die bei der Verwendung von GPL-Software im Hosting entsteht, denn hier wären Nutzer nicht zur Weitergabe des Quelltextes verpflichtet. Aufgrund dieser Klausel wird die Affero General Public License von der Free Software Foundation favorisiert [vgl.GnuA].

Das Service-as-a-Software-Substitute (SaaS) Problem wird durch die AGPL nicht gelöst. SaaS definiert die Nutzung von Webservern einer dritten Person, heißt die eigenen Daten werden beispielweise an eine IT-Firma gesendet, die diese verarbeitet und dann wieder zum Nutzer zurückschickt. Nachteil hier ist, man hat als Nutzer keine Kontrolle über die Datenverarbeitung die auf dem Webserver des Dienstleisters durch-



Abbildung 6: Logo AGPLv3 [vgl. GnuA]

geführt wird und kann nicht kontrollieren auf welche Art die eigenen Daten durchsucht werden [vgl. FsA].

Bis jetzt wurden laut Free Software Foundation noch keine Möglichkeiten gefunden, dieses Problem zu lösen. Es wird empfohlen auf die Dienste die SaaS sind zu verzichten, um diesem möglichen Unrecht zu entgehen [vgl. FsA].

9 Vergleich GNU GPL und Apache License

Wenn man beide Lizenzen gegenüberstellt ist ein Hauptunterschied in der Copyleft Stärke zu erkennen. Wo die GNU General Public License einen sehr strengen Copyleft Effekt hat, mit der Verpflichtung Änderungen und abgeänderte Versionen nur unter die Lizenz der Originalen Software zu stellen, hat der Nutzer bei der Apache License deutlich mehr Freiheiten, da diese unter die Non-Copyleft Lizenzen fällt. Hier hat der Nutzer alle Freiheiten und somit keine Verpflichtungen hinsichtlich der Nutzung und der Weitergabe der Software [vgl. Ifr].

Daraus lässt sich erkennen, dass es hier im Bezug auf die Freiheit der Software auch einen klaren Unterschied gibt. Durch die größeren Freiheiten bei der Apache License haben Nutzer sogar die Möglichkeit eine eigentlich Freie Software zu einer proprietären zu ändern und folglich müssen sie hier auch keinen Quellcode mehr offenlegen. Was zur Folge hat, dass der Zugang zu der betroffenen Software unterbunden wird. Im Vergleich dazu ist so eine Änderung bei GPL lizenzierter Software nicht möglich, da Nutzer erst gar nicht zu so einer Freiheit kommen durch die verpflichtende Lizenzierung der Software unter der Originallizenz.

9.1 Kompatibilität

Die Apache License 2.0 ist mit der dritten Version der GNU General Public License kompatibel. Jedoch ist die gemeinsame Verwendung mit vorherigen Versionen der GNU GPL nicht möglich.

Die Kompatibilität wird durch die erlaubte Nutzung von Codes der Apache License 2.0 in Projekten der GPLv3 ermöglicht. Das fertige Projekt muss dann aber als Ganzes unter der GPLv3 lizenziert werden. Möglich gemacht wird das durch gleiche Eigenschaften. Denn die Kriterien aus Punkt 7.1. sind als Untermenge in der GPLv3 enthalten [vgl. AL].

Es muss auch darauf geachtet werden, dass die Apache License keine Freiheiten entzieht, denn als freizügige Lizenz ist das Entziehen der Freiheiten bei Weitergabe möglich. Doch um mit GPL lizenzierter Software kompatibel zu sein, müssen die Freiheiten weiterhin aufrecht bleiben. Es dürfen ausschließlich Änderungen vorgenommen werden, wenn das Copyleft der GPL weiterhin erfüllt wird [vgl. Gnu].

Da unter der der Apache License viele kostenlose Softwareprogramme erhältlich sind ist sie demnach auch von einer sehr großen Anzahl an Nutzern und Entwicklern umgeben. Durch die Kompatibilität zwischen den beiden Softwarelizenzen erhofft man sich die Zusammenarbeit und den Austausch von Entwicklern von Freier Software weiter zu fördern [vgl. Gpl].

10 Zusammenfassung und Outlook

Freie Software ist über die letzten Jahre immer bekannter und auch beliebter bei Nutzern geworden. Nicht nur für den privaten Gebrauch ist es interessanter geworden auch Unternehmen fangen an diese Art von Software vermehrt zu nutzen. Nutzer erkennen die großen Vorteile von Freier Software, die von Richard Stallman in Form von vier Freiheiten bereits in den 80er Jahren festgelegt wurden.

Wenn man den Begriff „Freie Software“ hört assoziiert man sofort Wörter wie kostenlos oder gratis damit, doch die eigentliche Bezeichnung hat eine ganz andere Bedeutung. Ziel war es einen Begriff zu verwenden der die freie Meinungsäußerung und Redefreiheit widerspiegelt und Werte aufweist die jeden gleichberechtigten und allen einen Zugang zu der Software ermöglichen. Wie Richard Stallman sagte sollte man bei Frei an , „`free´ as in ´free speech´, not as in ´free beer´“ [vgl. Stall, 2002, S. 43] denken.

Freie Software bietet Nutzern viele Vorteile, von der freien Verfügbarkeit bis hin zur individuellen Bearbeitung. Doch die oftmals unzähligen Versionen eines Programmes können auch dazu führen, dass Fehler bei der Ausführung passieren. Dies kann dann unterschiedliche Gründe haben, entweder veränderte Versionen sind nicht mehr kompatibel mit bestimmten Programmen oder sie sind oft zu unübersichtlich, sodass sich Leien nicht auskennen und die Software nicht mehr sinnvoll verwenden können.

Richard Stallman hat mit der Freien Software Bewegung und der Einführung des Copyleft den Grundstein für die freie Verfügbarkeit von Software gelegt. Das GNU Projekt ist nur ein Teil eines großen Projekts, dass die Förderung von Freier Software durch die GNU General Public License unterstützt. Durch diese wird sichergestellt, dass jeder Nutzer

den Zugang zu Programmen und deren Quellcodes hat und diese auch individuell weiterverarbeiten kann. Gleiches gilt für die Apache License, nur hier in einer lockereren Variante. Denn diese fällt unter die Non Copyleft Lizenzen, wo es Nutzern zusätzlich erlaubt ist die Lizenz für die Software frei zu wählen oder diese auch proprietär zu machen.

Literaturverzeichnis

- [Ian] Iannella, R. (2001), "Digital Rights Management (DRM) Architectures", in: D-Lib Magazine, 7(6)
- [Mar] Martin, E.J. (2013), "Open Source: Is It Right for You?", in: EContent, 36(1), S.14-19
- [Mum] Mumenthaler, R. (2018), "E-Books: Grundlagen und Praxis", in: Creative Commons Nammensnennur 4.0 International
- [Mus] Mustonen, M. (2002), „Copyleft – the economics of Linux and other open source software“, in: Information Economics and Policy, 15(2003), S. 99-121
- [Pen] Pentas, P. (2005), "Evaluation of open source content management systems based on the specific requirements of the Intercultural Academic Network", in: Universität für angewandte Wissenschaft Darmstadt
- [Per] Perens, B. (1999), "The open source definition", in: Open source: voices from the open source revolution, 1, S. 171-188
- [Stall] Stallman, R.M. (2002), "Free Software, Free Society: Selected Essays of Richard M. Stallman", in: GNU Press
- [Vij] Vijay, N., Aruna, T. (2014), "Free Open Source Software in Electronics Engineering Education: A Survey", in I.J. Modern Education and Computer Science, 5, S. 15-25
- [Webb] Webb, W. (2008), "Free software encircles embedded design", in: EDN Cambridge, 53(16), S.29-35

Online Literatur:

- [AL] Apache Lizenz. <https://de.wikipedia.org/wiki/Apache-Lizenz>, besucht am 11.5.2022

- [ASF] Apache Software Foundation.
https://de.wikipedia.org/wiki/Apache_Software_Foundation, besucht am 10.5.2022
- [ASF1] Apache Software Foundation. <https://www.linux-magazin.de/ausgaben/2015/08/apache-projekte/>, besucht am 10.5.2022
- [Apa] Apache Software Foundation: Apache License, Version 2.0.
<https://www.apache.org/licenses/LICENSE-2.0>, besucht am 10.5.2022
- [Byn] Bynder: Definition: Digital Rights Management.
<https://www.bynder.com/de/glossar/digital-rights-management/>, besucht am 23.4.2022
- [CCG] CC German Chapter: FAQ. <https://oer-faq.de/faq/CC-4-3-2>, besucht am 18.03.2022
- [DRM] Digitale Rechteverwaltung: DRM gegenüber freier Software.
https://de.wikipedia.org/wiki/Digitale_Rechteverwaltung#DRM_gegen%C3%BCber_freier_Software, besucht am 23.4.2022
- [FOS] Do-FOSS Vertrauen braucht Freie Software: Vorteile Freier Software. <https://blog.do-foss.de/freie-software/vorteile/>, besucht am 18.03.2022
- [FsA] Free Software Foundation: Warum die GNU Affero GPL.
<https://www.gnu.org/licenses/why-affero-gpl>, besucht am 27.5.2022
- [Fsf] Free Software Foundation.
https://de.wikipedia.org/wiki/Free_Software_Foundation, besucht am 2.4.2022
- [Fsf1] Free Software Foundation: About. <https://www.fsf.org/about/>, besucht am 2.4.2022
- [FSF] Free Software Foundation: Freie-Software-Definition.
<https://www.gnu.org/philosophy/free-sw.html>, besucht am 18.03.2022

- [FSF1] Free Software Foundation: Warum "Open Source" das Ziel Freie Software verfehlt. <https://www.gnu.org/philosophy/open-source-misses-the-point>, besucht am 23.3.2022]
- [FSF2] Free Software Foundation: Copyleft. Was ist das?. <https://www.gnu.org/licenses/copyleft.html>, besucht am 27.3.2022
- [FSF3] Free Software Foundation: Geschichte des GNU Systems. <https://www.gnu.org/gnu/gnu-history.html>, besucht am 8.4.2022
- [FS1] Freie Software: Abgrenzung Open Source. https://de.wikipedia.org/wiki/Freie_Software#Open_Source, besucht am 23.3.2022
- [GnuA] GNU Affero General Public License. https://de.wikipedia.org/wiki/GNU_Affero_General_Public_License, besucht am 27.5.2022
- [Gnu] GNU General Public License https://de.wikipedia.org/wiki/GNU_General_Public_License, besucht am 13.4.2022
- [GNU] GNU-Projekt: Geschichte. <https://de.wikipedia.org/wiki/GNU-Projekt#Geschichte>, besucht am 8.4.2022
- [Gpl] Free Software Foundation: GNU GPLv3: Eine Kurzanleitung. <https://www.gnu.org/licenses/quick-guide-gplv3>, besucht am 13.4.2022
- [Ifrr] Institut für Rechtsfragen der Freien und Open Source Software: Lizenzcenter. <https://ifross.github.io/ifrOSS/Lizenzcenter#lizenzen-ohne-copyleft-effekt-permissive-licenses>, besucht am 27.3.2022
- [MRD] Mr. Datenschutz: Freie Software: Die Vorteile und Nachteile von Open Source. <https://mrdatenschutz.de/open-source-die-vorteile-und-nachteile-freier-software/>, besucht am 19.03.2022
- [Nrf] Open Source: Nutzungsrechte an Freier Software (Open Source). <https://www.it-recht-kanzlei.de/open-source-software-nutzungsrechte.html>, besucht am 27.3.2022
- [OSI] Open Source: Definition der Open Source Initiative. https://de.wikipedia.org/wiki/Open_Source, besucht am 23.3.2022

- [OSI1] Open Source Initiative: Definition. <https://opensource.org/osd>, besucht am 25.3.2022
- [OSA] WSS Redpoint: Open Source Lizenzen – Die Apache Lizenz. <https://wss-redpoint.com/open-source-lizenzen-die-apache-lizenz/>, besucht am 12.5.2022
- [OSL] Open-Source-Lizenzen – Grundlagen und Entscheidungshilfen. <https://www.bitfactory.io/de/blog/open-source-lizenzen/>, besucht am 27.3.2022
- [PFS] Freie Software. https://de.wikipedia.org/wiki/Freie_Software#/media/Datei:Konzept-karte_der_Freien_Software.svg, besucht am 18.03.2022
- [Red] Was ist Open Source?: Freie, geschlossene und quelloffene Software. <https://www.redhat.com/de/topics/open-source/what-is-open-source>, besucht am 25.3.2022
- [STA] Wikipedia: Richard Stallman https://de.wikipedia.org/w/index.php?title=Richard_Stallman&stable_id=222960786, besucht am 18.03.2022
- [Qua] Linux-Treiber entwickeln: Nicht vergessen: Auswahl einer geeigneten Lizenz. <https://ezs.kr.hsnr.de/TreiberBuch/html/sec.lizenz.html>, besucht am 13.4.2022

Weiterführende Literatur:

Unix Betriebssystem: <http://www.netzmafia.de/skripten/unix/unix1.html>

Black Duck Hub: <https://www.ccpsoft.de/black-duck-hub/>