

Open Source und freie Software
-
Gemeinsamkeiten und Unterschiede im historischen Kontext

Rico Warnke

Seminararbeit zum Seminar „Kreativität im digitalen Wandel“

Leitung: Prof. Hans-Gert Gräbe, Ken Pierre Kleemann

Leipzig, 30.09.2020

Gliederung

1. Einleitung.....	3
2. Begriffserklärung	4
2.1. Open Source	4
2.2. Freie Software	6
2.3. Proprietäre Software.....	7
3. Historische Entwicklung	7
3.1. Entstehung proprietärer und freier Software.....	7
3.2. Das GNU-Projekt und die FSF	10
3.3. Die GNU-Lizenzen.....	11
3.4. Linus Torvalds - Linux	12
3.5. Eric S Raymond - Die Kathedrale und der Basar	13
3.6. Abspaltung der Open Source Bewegung	13
3.7. Mozilla / Android	14
4. Gemeinsamkeiten und Unterschiede	15
4.1. Gemeinsamkeiten.....	15
4.2. Unterschiede	16
5. Fazit.....	18

1. Einleitung

Linux ist vielen Softwarenutzern ein Begriff, doch worauf basiert es? Korrekter Weise trägt das freie Betriebssystem den Namen "GNU/Linux", wobei Linux lediglich den Kernel bezeichnet. Es stellt sich die Frage ob Linux sich in all seinen Derivaten eindeutig zu freier Software oder zu Open Source Software zuordnen lässt? Daraus ergibt sich die Fragestellung dieser Arbeit, ob und wie sich freie Software und Open Source Software unterscheiden lassen. Hier soll die historische Entwicklung beider Softwarearten dargestellt werden, um anschließend Gemeinsamkeiten und Unterschiede zu betrachten. Zunächst werden die Begriffe "Open Source" und "Freie Software" definiert, sowie auch der Begriff der "proprietäre Software", der beiden Prinzipien entgegensteht. Es folgt ein historischer Abriss über die Entstehungsgeschichte von freier Software, mit Richard Stallman als zentraler Figur. Es werden Stallmans Beweggründe, Software zu revolutionieren und sie als freies Gut zu betrachten dargelegt. Das GNU-Projekts sowie die Gründung der "Free Software Foundation" stehen hierbei im Vordergrund und verschiedene GNU-Lizenzen werden erläutert.

Anschließend wird die Vervollständigung des Betriebssystems GNU/Linux, durch Linus Torvalds Beitrag des Linux-Kernels beleuchtet. Unterschiedliche Ansichten zu Software führten zur Abspaltung einer Gruppierung, der Open Source Bewegung, mit dem Ziel Software nicht unter ethischen Aspekten sondern unter wirtschaftlichen Aspekten zu betrachten. Hierbei wird insbesondere das Mozilla-Projekt sowieso die Entwicklung von Android als Betriebssystem für mobile Endgeräte fokussiert.

Abschließend wird dargelegt, wie sich freie Software und Open Source Software aufgrund ihrer historischen Entwicklung unterscheiden und welche Eigenschaften sie gemeinsam haben.

2. Begriffserklärung

2.1. Open Source

Bei der Open-Source-Definition¹ handelt es sich um eine überarbeitete Version der Debian-Richtlinien für freie Software² (DFSG), die im Jahr 1998 durch die Open Source Initiative (OSI) erstellt wurde.

Open Source beinhaltet nicht nur den Zugriff auf den Quellcode, auch die Weitergabe von Open-Source-Programmen muss die folgenden Anforderungen erfüllen:

Freie Weiterverteilung

Die Lizenz darf keine Person daran hindern, die Software als Bestandteil einer aggregierten Software-Distribution, die Programme aus mehreren verschiedenen Quellen enthält, zu verkaufen oder zu verschenken. Die Lizenz darf keine Lizenzgebühr oder sonstige Gebühr für einen solchen Verkauf verlangen.

Offener Quellcode

Das Programm muss im Quellcode verfügbar sein und die Verbreitung sowohl im Quellcode als auch in kompilierter Form erlauben. Absichtlich verschleierter Quellcode ist nicht erlaubt. Zwischenformen wie z.B. die Ausgabe eines Präprozessors sind nicht erlaubt.

Abgeleitete Werke

Die Lizenz muss Modifikationen und abgeleitete Werke zu lassen und deren Verbreitung unter den gleichen Bedingungen erlauben wie die Lizenz der Originalsoftware.

¹ <https://opensource.org/docs/osd> (Letzter Zugriff: 12.05.2020).

² https://www.debian.org/social_contract.html#guidelines (Letzter Zugriff: 12.05.2020).

Integrität des Urheber-Quellcodes

Die Lizenz muss die Verbreitung von Software, die aus modifiziertem Quellcode erstellt wurde, ausdrücklich erlauben. Die Lizenz kann verlangen, dass abgeleitete Werke einen anderen Namen oder eine andere Versionsnummer als die Originalsoftware tragen müssen.

Keine Benachteiligung von Personen oder Gruppen

Die Lizenz muss allen Personen oder Gruppen die Verwendung der Software zugänglich machen.

Keine Einschränkung der Verwendung

Die Lizenz darf niemanden daran hindern, das Programm in einem bestimmten Bereich zu nutzen z.B. darf die Nutzung des Programms in einem Unternehmen nicht eingeschränkt werden.

Verteilung der Lizenz

Die mit dem Programm verbundenen Rechte müssen für alle gelten, an die das Programm weiterverbreitet wird, ohne dass eine zusätzliche Lizenz von diesen Beteiligten erworben werden muss.

Die Lizenz darf nicht produktspezifisch sein

Die mit dem Programm verbundenen Rechte dürfen nicht davon abhängen, dass das Programm Teil einer bestimmten Software-Distribution, also eine Zusammenstellung von Software-Komponenten, ist. Wenn das Programm aus dieser Distribution extrahiert wurde und innerhalb der Lizenzbedingungen des Programms verwendet oder weitergegeben wird, sollten alle Beteiligten, an die das Programm weiterverbreitet wird, die gleichen Rechte haben, die in Verbindung mit der ursprünglichen Software-Distribution gewährt werden.

Die Lizenz darf andere Software nicht einschränken

Die Lizenz darf keine Einschränkungen für andere Software enthalten, die zusammen mit der lizenzierten Software vertrieben wird. Beispielsweise darf die Lizenz nicht darauf bestehen, dass alle anderen Programme, die auf dem gleichen Medium verbreitet werden, Open-Source-Software sein müssen.

Technologieneutralität

Keine Bestimmung der Lizenz darf auf einer individuellen Technologie oder Art der Schnittstelle basieren.

2.2. Freie Software

Freie Software bedeutet, dass Nutzer die Freiheit haben Software auszuführen, zu kopieren, zu verbreiten, zu untersuchen, zu ändern und zu verbessern.³

Sie ist daher nicht eine Frage des Preises, sondern der Freiheit sie zu nutzen. Eine weitere Bezeichnung ist Libre Software. Dieser Name ist dem spanischen bzw. französischen Wort für frei entnommen und macht deutlich, dass es sich nicht nur um gratis Software handelt. Die Nutzer kontrollieren also das Programm und was es für sie ausführt.

Als Freie Software wird Software angesehen, wenn sie dem Nutzer folgenden vier Freiheiten gewährt:

Freiheit 0

Die Freiheit, die Software auszuführen wie man möchte, für jeden Zweck.

Freiheit 1

Die Freiheit, die Programmfunktionen zu analysieren und den individuellen Bedürfnissen anzupassen. Dabei ist der Zugang zum Quellcode eine Voraussetzung.

³ <http://www.gnu.org/philosophy/free-sw> (letzter Zugriff: 12.05.2020).

Freiheit 2

Die Freiheit, die Software weiterzugeben um seinen Mitmenschen damit zu helfen.

Freiheit 3

Die Freiheit, die Software zu modifizieren und seine Verbesserung zu veröffentlichen.

Ein freies Programm muss für die kommerzielle Nutzung, Vertrieb und Entwicklung zugänglich sein. Unabhängig von der Art und Weise, wie man eine Kopie freier Software erhalten hat, hat man immer die Freiheit diese zu ändern, zu kopieren oder gar zu verkaufen.

2.3. Proprietäre Software

Eine Software gilt als proprietär, wenn sie das Recht sowie die Möglichkeiten für Anpassung, Änderung, Weiter- und Wiederverwendung des Nutzers stark einschränkt.⁴ Als Ursprung proprietärer Software wird ihre Abhängigkeit von der Hardware angesehen.

Mit der zunehmenden Verbreitung von Computern mit gleichen Mikroprozessoren in den 1980er Jahren, nahm auch die Methode, Quelltexte unter Verschluss zu halten zu. Als Möglichkeiten der Proprietärisierung zählen Softwarepatente, das Urheberrecht, Lizenzbedingungen, herstellerspezifischer Softwareaufbau und die Behandlung des Quelltextes als Betriebsgeheimnis.

3. Historische Entwicklung

3.1. Entstehung proprietärer und freier Software

Die Ursprünge der Freien Software sind zurückzuführen auf die "Hacker-Kultur", welche sich an den akademischen US-Einrichtungen "Massachusetts institute of technology (MIT)", "Leland Stanford Junior University", "University of California, Berkeley" und der "Carnegie Mellon University (CMU)" in den 1960er- und 1970er-Jahre bildete.

⁴ https://wiki.ubuntuusers.de/unfreie_Software/?rev=231304 (Letzter Zugriff: 20.05.2020).

Bis zum Jahr 1969 galt es als gängige Praxis, Software inklusive Quelltext mit Computersystemen auszuliefern. Diese wurde als Ergänzung angesehen, um das System nutzbar zu machen. Durch das Prinzip der Offenlegung des Quelltextes, erhielten die Hersteller vielerlei Fehlerkorrekturen und Vorschläge für die Modifikation ihrer Programme.

Im Juni 1969 kündigte das Unternehmen International Business Machines Corporation (IBM) neue Regelungen für ihre Software an. Die Software wurde von den bisherigen Hardware-Nutzungsbedingungen entbunden und separat durch Lizenzverträge urheberrechtlich geschützt. Die Weiterentwicklung und Wartung ihrer Software war nun nicht länger eine freie Dienstleistung, sondern eine kommerzielle Dienstleistung des Unternehmens. Somit wurde ein eigener Wirtschaftsmarkt begründet.

Weitere Unternehmen begannen in den späten 1970er Jahren ebenfalls mit der Einführung von Softwarelizenzen, welche die Möglichkeiten der Modifikation und der Verteilung der Software beschränkten. Weiterhin erfolgte die Auslieferung der Programme im Binärcode und nicht länger im Quelltext, was Modifikationen nahezu unmöglich machte und zum Schutz der Software als Geschäftsgeheimnis diente. Da Software nun getrennt von Hardware verkauft wurde und der Quelltext vor der Konkurrenz verborgen wurde, galt sie als proprietäre Software.⁵

Der Programmierer Richard Matthew Stallman arbeitete in der Abteilung für künstliche Intelligenz des Massachusetts Institute of Technology, mit einer Gruppe Programmierern zusammen, welche sich selbst als "Hacker" bezeichneten. Diese Hackergemeinschaft lebte eine Philosophie des unbegrenzten Informationsflusses und es war ganz natürlich, eigene Software-Modifikationen und deren Quelltext mit anderen Programmierern zu teilen.

Mit der Einführung proprietärer Software in den Laboren des MIT, begann Stallman gegen eine Monopolstellung proprietärer Anbieter, durch Entwicklung alternativer Software, vorzugehen.

Das Unternehmen Bells Laboratories entwickelte Unix, ein Mehrbenutzer-Betriebssystem, zur Unterstützung der Softwareentwicklung, im Jahr 1969. Unix galt bis 1983 als quelloffenes Betriebssystem, dessen Entwicklung maßgeblich durch US-

⁵ <https://www.heise.de/developer/meldung/Vor-50-Jahren-Unbundling-Impuls-fuer-eine-unabhaengige-Softwareindustrie-4453306.html> (Letzter Zugriff: 18.05.2020).

Universitäten vorangetrieben wurde und erheblichen Einfluss auf die Hacker-Kultur hatte. American Telephone and Telegraph Company (AT&T) entschied sich im Jahr 1983 für die Vermarktung von UNIX System V. , eine proprietäre Version ihres Unix, was zahllose Informatiker die an der Mitentwicklung des Betriebssystems beteiligt waren, erzürnte.

Richard Stallman nennt als Schlüsselereignis, das ihn zum Start des GNU-Projekt veranlasste, ein Szene die er Anfang der 1980er Jahre am MIT erlebte.

Sie nutzten einen Xerox-Netzwerkdrucker in einem anderen Raum, an dem Druckaufträge eingingen. Wollte man seinen Druck einige Zeit später abholen, stellte man meist fest dass es zu einem Papierstau kam oder der Drucker ausgegangen war. Stallmans Vorhaben bestand in der Implementierung einer Funktion, die den Status des Druckers am Arbeitsplatz anzeigen sollte. Er wand sich an Xerox um den Quelltext des Druckertreibers zu erhalten, doch wurde ihm dieser verweigert, da sich der Mitarbeiter zur Nichtweitergabe des Quelltexts verpflichtet hatte.

Es stellt sich also ein praktisches Problem, dessen Lösung darin bestand, eine Software um eine Funktion zu erweitern. Die Mauer die man als "geistiges Eigentum" bezeichnet, macht eine Lösung des Problems unmöglich. Es bestünde zwar die Möglichkeit, die Firma um die Implementierung dieser Funktion zu bitten und diese zu verbreiten, was jedoch langwierig und unsicher ist. Des Weiteren wäre eine Dekompilierung sehr mühsam, zeitraubend und nur in begrenztem Maße legal.

Stallman stand vor der Frage, welche Voraussetzungen für eine neue Gemeinschaft der freien Software geschaffen werden mussten. Zur Nutzung eines Computers liegt die Priorität zunächst auf einem Betriebssystem und Betriebssysteme waren ein Spezialgebiet Stallmans. Daraufhin initialisierte er im Jahr 1983 das GNU-Projekt, welches das Ziel hatte, ein Betriebssystem zu entwickeln, welches ein Äquivalent zu Unix darstellt, jedoch keine Zeile des von AT&T geschützten Codes enthielt und in freier Kooperation weiterentwickelt werden konnte ohne dabei im Alleineigentum des Entwicklers zu stehen und von der Allgemeinheit frei verwendet werden konnte.⁶

⁶https://mediarep.org/bitstream/handle/doc/4317/Grassmuck_2004_Freie_Software_.pdf?sequence=3&isAllowed=y, S. 211-222, (Letzter Zugriff: 18.05.2020).

3.2. Das GNU-Projekt und die FSF

Die Bezeichnung GNU steht für "GNU's not Unix". Richard Stallmans Vorhaben bestand darin, ein Äquivalent zum bestehenden Betriebssystem Unix zu entwickeln. Die Namensentscheidung resultierte daraus, dass es in seiner Zeit am MIT üblich war, für Programme, die anderen Programmen ähneln, rekursive Akronyme zu nutzen.

Seine Wahl zur Nähe zu Unix war keineswegs unbegründet. Zum einen hatte es sich bewährt, war portabel und verfügte bereits über eine weltweite Unix-Gemeinde. Des Weiteren war er überzeugt, dass ein grundlegend neues Betriebssystem, auf dem bestehende Programme nicht lauffähig wären, bei den meisten Unternehmen auf Ablehnung stoßen würde.⁷

Zum Zweck der Förderung freier Software und der Finanzierung des GNU-Projektes gründete Stallman 1985 die Free Software Foundation (FSF). Die Hauptaufgabe der FSF besteht in der juristischen, personellen, technischen und finanziellen Unterstützung des GNU-Projektes. Neben der Software beinhaltet dies auch die Lizenzen "General Public License" (GPL), "Lesser General Public License" (LGPL), "Affero General Public License" (AGPL) und "Free Documentation License" (FDL) sowieso allgemeine Aufklärung, Berichterstattung und Beratung rund um das Thema freie Software.⁸

Die erste bekannte Definition freier Software wurde im Februar 1986 von der FSF veröffentlicht. Diese Definition bezog sich zunächst noch auf nur 2 Freiheiten. Erstens, die Freiheit, ein Programm zu kopieren und weiterzuverbreiten, so dass Andere das Programm ebenfalls nutzen können. Zweitens, die Freiheit, ein Programm zu ändern, so dass der Nutzer es beherrscht und nicht vom Programm beherrscht wird. Für diesen Zweck muss der Zugang zum Quelltext des Programms gewährleistet sein.⁹

Zur Bereitstellung Freier Software existieren mehrere Methoden. Eine einfache Form besteht darin, das Programm in die Gemeinfreiheit - Public Domain - zu entlassen, was bedeutet dass das Programm nicht dem Urheberrecht unterliegt. Diese Form der

⁷ <https://www.gnu.org/gnu/manifesto> (Letzter Zugriff: 20.05.2020).

⁸ <https://www.fsf.org/about/> (Letzter Zugriff: 20.05.2020).

⁹ <https://www.gnu.org/bulletins/bull1.txt> (Letzter Zugriff: 18.05.2020).

Bereitstellung ermöglicht es zwar ein Programm und dessen Modifikationen mit anderen zu teilen, aber es besteht auch die Gefahr der Umwandlung des Programms in proprietäre Software.

Eine weitere Methode der Bereitstellung ist das Copyleft. Copyleft ermöglicht es ein Programm frei nutzbar zu machen und verlangt zusätzlich, dass alle erweiterten und modifizierten Versionen ebenfalls frei nutzbar sind.¹⁰

3.3. Die GNU-Lizenzen

GPL

Das Copyleft findet seine Anwendung in der GNU "General Public License", welche Richard Stallman im Januar 1989 veröffentlichte. Die GPL gestattet es dem Nutzer lizenzgebührenfrei, die Software zu verbreiten, zu vervielfältigen, öffentlich zugänglich zu machen und beliebige Veränderungen vorzunehmen. Um die Veränderung zu ermöglichen muss der Quelltext der Software offengelegt werden. Des Weiteren fordert die GPL dass jede veränderte Version der Software die vertrieben wird ebenfalls der GPL unterliegt.¹¹

LGPL

Da es nach den Festlegungen der GPL nicht möglich wäre, auf Basis einer GPL-Softwareumgebung kommerzielle Software zu entwickeln, entstand 1991 die "Lesser General Public License". Die LGPL gestattet es Entwicklern, LGPL-Software in proprietärer Software zu verwenden oder einzubinden. Diese Einbindung geschieht meist in Form einer dynamischen Programmbibliothek, um die Trennung von LGPL-Software und proprietärer Software zu gewährleisten.¹²

¹⁰ <https://www.gnu.org/licenses/copyleft.html> (Letzter Zugriff: 24.05.2020).

¹¹ <https://www.gnu.org/licenses/gpl-3.0> (Letzter Zugriff: 24.05.2020).

¹² <https://www.gnu.org/licenses/lgpl-3.0> (Letzter Zugriff: 24.05.2020).

AGPL

Die "Affero General Public License" wurde für Programme geschaffen, die als Dienst auf einem Server ausgeführt werden. Diese räumt dem Nutzer das Recht ein, eine Downloadmöglichkeit für den Quelltext zu erhalten, auch wenn das Programm selbst nicht explizit zum Download angeboten wird.¹³

FDL

Bei der "Free Documentation License" handelt es sich um eine Copyleft-Lizenz für freie Dokumentation. Die FDL wurde geschaffen um Dokumente, wie z.B. Handbücher, die im Rahmen des GNU-Projekts geschaffen wurden, unter eine ähnliche Lizenz wie die Software zu stellen. Die Lizenz erlaubt die Vervielfältigung, Verbreitung und Veränderung des Werkes, auch zu kommerziellen Zwecken. Im Gegenzug verpflichtet sich der Nutzer zur Nennung des Autors und dazu, abgeleitete Werke unter dieselbe Lizenz zu stellen.¹⁴

3.4. Linus Torvalds - Linux

Bis zum Jahr 1990 waren die wichtigsten Komponenten des GNU-Systems entwickelt, mit Ausnahme eines Kernel. Als Kernel wird der Betriebssystemkern bezeichnet, der Prozess- und Datenorganisation festlegt. Er bildet die unterste Software-schicht mit direktem Zugriff auf die Hardware und stellt die Grundlage der Software-Bestandteile des Betriebssystems dar. Der finnische Informatikstudent Linus Torvalds entwickelte auf Grundlage des freien unixoiden Betriebssystems Minix und mit Hilfe der Werkzeuge des GNU-Projekts 1991 Linux, einen freien unixoiden Betriebssystemkern, welchen er 1992 unter der GNU GPL als freie Software veröffentlichte. Die Kombination des nahezu kompletten GNU-Systems mit dem Linux-Kernel führte im März 1994 zur Version 1.0 des vollständigen, freien Betriebssystems "GNU/Linux". Die Distribution des Systems erfolgte zunächst über das Netz, später auch auf Disketten und CD-ROM. Die zunehmende Beliebtheit spiegelt sich in den Nutzerzahlen wieder. So gab es im Jahr 1995 bereits 500.000 GNU/Linux-Nutzer. Bis 1997 steigerte sich diese Anzahl auf bereits 3,5 Millionen. Was sich als freie

¹³ <https://www.gnu.org/licenses/agpl-3.0.html> (Letzter Zugriff: 24.05.2020).

¹⁴ <https://www.gnu.org/licenses/fdl-1.3.html> (Letzter Zugriff: 24.05.2020).

Software aus dem GNU-Projekt entwickelte hatte und um weitere freie Software, wie dem Linux-Kernel erweitert wurde, stellte inzwischen eine interessante Plattform für Anbieter von Anwendungssoftware dar.¹⁵

3.5. Eric S Raymond - Die Kathedrale und der Basar

Zu diesem Zeitpunkt war die Entwicklung freier Software vor allem für den privaten und universitären Bereich interessant. Um nun auch der Wirtschaft die Vorteile dieses Entwicklungsmodells vor Augen zu führen, verfasste der Softwareentwickler Eric Steven Raymond 1997 das Essay “Die Kathedrale und der Basar” , welches als Schlüsseltext für eine begriffliche Wende angesehen wird.

Raymond stellt in seinem Essay ein zentral gelenktes Projekt wie den Bau einer Kathedrale, dem kreativen und selbstorganisierende Gewirr eines Basars gegenüber. Hierbei stellt Microsoft Windows die Kathedrale und die Linux-Gemeinde den Basar dar.¹⁶

3.6. Abspaltung der Open Source Bewegung

Ein Auslöser für den Beginn der Open Source Bewegung ist der Beschluss des Unternehmens Netscape, den Quelltext ihres Webbrowsers “Netscape Navigator” im Jahr 1998 zu veröffentlichen.¹⁷ Auch IBM stellte seinen Apache HTTP Server unter die Apache-Lizenz, einer anerkannten Freie-Software-Lizenz, als Reaktion auf die Veränderung der Marktbedingungen und die starke Konkurrenz durch die Entwicklungen im Internet.¹⁸ Eine Gruppe Interessenten für freie Software und GNU/Linux, darunter Eric S. Raymond und Bruce Perens, beschloss die Einführung eines neue Marketing-Begriff, welcher die Wettbewerbsfähigkeit freier Software steigern sollte. Sie entschieden sich für den Begriff “Open Source”. Open Source hat das Ziel, freie Software geschäftsfreundlich, weniger ideologisch belastet und frei von ethischen und sozialen Komponenten der FSF darzustellen. Im Jahr 1998 gründeten sie die

¹⁵https://mediarep.org/bitstream/handle/doc/4317/Grassmuck_2004_Freie_Software_.pdf?sequence=3&isAllowed=y, S. 226-229, (Letzter Zugriff: 18.05.2020).

¹⁶ https://www.selflinux.org/selflinux/pdf/die_kathedrale_und_der_basar.pdf (Letzter Zugriff: 25.05.2020).

¹⁷ <https://www.heise.de/newsticker/meldung/Netscape-Browser-gratis-Quelltext-frei-10711.html> (Letzter Zugriff: 25.05.2020).

¹⁸ <https://www.computerworld.com/article/2585020/evolution-of-ibm-s-open-source-strategy.html> (Letzter Zugriff: 25.05.2020).

Organisation "Open Source Initiative" (OSI). Auf Grundlage der "Debian Free Software Guidelines" die Bruce Perens, ehemaliger Projektleiter der Linux-Distribution Debian, bereits 1997 verfasste, entwarf er 1998 die offizielle Open Source Definition. Die Hauptaufgabe der OSI besteht darin, Softwarelizenzen anhand ihrer Definition des Begriffs "Open Source" zu zertifizieren.¹⁹

3.7. Mozilla / Android

Mit der Veröffentlichung des Quelltextes der Netscape-Browsersuite entstand 1998 das Mozilla Projekt. Die Mitglieder dieser offenen Gemeinschaft arbeiteten nicht einfach am nächsten Netscape-Browser, sondern begannen viele Browser, Entwicklerwerkzeuge und andere Projekte zu entwickeln. Mit Leidenschaft schufen sie freie Software, die es Menschen ermöglichen sollte in ihrem Interneterlebnis eine Wahl zu haben. 2002 erschien Mozilla 1.0, welche viele Verbesserungen im Browser und im E-Mail-Programm brachte, welche zunächst aber kaum genutzt wurde, da über 90% der Internetnutzer mit dem Microsoft Internet Explorer surfen. Im Jahr 2004 veröffentlichte Mozilla den Browser Firefox 1.0 mit großem Erfolg. Mehr als 100 Millionen Downloads in weniger als einem Jahr beschleunigten die Innovation und verbesserten das Internet für alle. Regelmäßige Updates stellen neue Rekorde auf. 2013 schuf Mozilla mit Firefox OS ein quelloffenes, linux-basiertes Betriebssystem für Smartphones und Tabletcomputer mit dem Ziel Benutzeroberfläche und Apps mit Web-Techniken wie HTML, CSS und JavaScript zu realisieren und damit Nutzern und auch Programmierern größtmögliche Offenheit und Kompatibilität zu bieten. Die Entwicklung wurde 2016 eingestellt, aber da Firefox OS Open Source ist, wird es weiterhin für eigenständige Nachfolgeprojekte verwendet, die auf dem Quelltext von Firefox OS aufsetzen.²⁰

Das Unternehmen Google kaufte 2005 das im Jahr 2003 von Andy Rubin gegründete Unternehmen Android, von dem bekannt war das es Software für Mobiltelefone entwickelte und standortbezogene Dienste behandelte. Ursprünglich sollte Android ausschließlich zur Steuerung von Digitalkameras dienen. Android wurde nicht von Grund auf neu entwickelt, sondern aus mehreren Open Source Komponenten neu zusammengestellt. Die wichtigsten Komponenten sein hierbei der Linux-Kernel, Java und

¹⁹ <https://opensource.org/history> (Letzter Zugriff: 25.05.2020).

²⁰ <https://www.mozilla.org/de/about/history/> (Letzter Zugriff: 23.09.2020).

die C-Bibliothek Bionic sowie Bibliotheken und Daemons des GNU-Projekts. Google kündigte 2007 an, gemeinsam mit 33 anderen Mitgliedern der Open Handset Alliance, ein Mobiltelefon-Betriebssystem namens Android weiter zu entwickeln, welches seit dem 21. Oktober 2008 offiziell verfügbar ist.²¹

4. Gemeinsamkeiten und Unterschiede

Erst mit der Bildung einer Open Source Bewegung, welche sich aus der Freien Software Bewegung entwickelte, lassen sich Gemeinsamkeiten und Unterschiede beider Konzepte aufzeigen. Die Gründung der "Open Source Initiative" im Jahr 1998 bildet den Ausgangspunkt dieser Betrachtung.

Freie Software und Open Source Software stehen für eine ähnliche Programmvierfalt, dennoch stehen beide Begriffe für grundlegend unterschiedliche Sachverhalte in Programmen, die auf andersartigen Werten basieren.

4.1. Gemeinsamkeiten

Da die Idee von Open Source ursprünglich darin bestand, das Konzept der freien Software wirtschaftlicher zu gestalten, ist es naheliegend das sich viele Gemeinsamkeiten zwischen Freier Software und Open Source Software finden lassen. Zunächst gilt es zu sagen, dass die Definitionen beider Begriffe in den meistens Fällen übereinstimmen. Die Begriffe "Freie Software" und "Open Source Software" vermitteln den Eindruck es handelt sich um kostenlose Software, was nicht zwingend der Fall sein muss. Die Adjektive "frei" (engl. free) und "offen"(engl. open) werden aufgrund ihrer Mehrdeutigkeit meist als "gratis" interpretiert. Richard Stallman verdeutlichte die Bedeutung eines besseren Verständnisses der Definition mit "Frei wie in Redefreiheit, nicht wie in Freibier". Sowohl freie Software als auch Open Source Software verfügen über die Eigenschaft, dass ihr Quelltext der Öffentlichkeit zugänglich ist und es keine rechtlichen Einschränkungen zur Nutzung der Software gibt. Geringe bzw. keine Initialkosten der Software motivieren zu freiwilliger Mitarbeit.²² Freie Software und Open Source Software werden meist direkt von ihrem Endnutzer entwickelt und

²¹ <https://source.android.com/> (Letzter Zugriff: 23.09.2020).

²² <http://www.opensourcejahrbuch.de/download/jb2007/osjb2007-02-03-perens.pdf>, S.145-148, (Letzter Zugriff: 26.05.2020).

bieten somit die volle Kontrolle über die Anpassung des Endproduktes. Diese Tatsache führt oft zu einer effizienten Wartung und Weiterentwicklung.²³

Des Weiteren sehen sie proprietäre Software als ihren "Feind" an, da diese ihr Wissen der Allgemeinheit vorenthält. Da weder Freie Software noch Open Source Software dem proprietären Vertriebsmodell entstammen, decken diese auch viele Bereiche ab, die für einen Entwickler proprietärer Software nicht wirtschaftlich genug wäre.²⁴ Nicht zuletzt dadurch dass die Entwicklungskosten nicht von einer Person oder einem Unternehmen allein getragen werden, sondern durch eine hohe Effizienz der Zuweisung von Ressourcen für Verteilung der Kosten und Risiken kompensiert werden und zahlreiche Personen, Unternehmen und akademische Einrichtungen an der Entwicklung beteiligt sind.²⁵²⁶

Obleich sich Freie Software und Open Source in ihren Ansichten unterscheiden führen sie in vielen Fällen zum gleichen praktischen Verhalten, der Entwicklung leistungsstarker und zuverlässiger Software, oft auch in Kooperation.

4.2. Unterschiede

Auch wenn Freie Software und Open Source Software sich ähnlich sind, gibt es dennoch grundlegende Unterschiede. Zunächst sei gesagt dass Freie Software und Open Source zwei eigenständige Bewegungen, mit einer eigenen Sichtweise auf die Software-Welt sind, die jeweils andere Ansätze und Kriterien hat sowie eigene Philosophien verfolgen.

Das Hauptanliegen der Freie Software Bewegung besteht darin, die Freiheitsrechte im Bereich Software zu wahren und es jedem zu ermöglichen diese Software ohne Einschränkungen zu nutzen. Das Konzept der Freien Software spricht hauptsächlich Programmierer an. Das Problem proprietärer Software ist für sie eine ethische Frage und stellt ein soziales Problem dar. Sie stellen den Wert der Freiheit und dessen

²³ <http://www.opensourcejahrbuch.de/download/jb2007/osjb2007-02-03-perens.pdf>, S.144, (Letzter Zugriff: 26.05.2020).

²⁴ <http://www.opensourcejahrbuch.de/download/jb2007/osjb2007-02-03-perens.pdf>, S.139-141, (Letzter Zugriff: 26.05.2020).

²⁵ <http://www.opensourcejahrbuch.de/download/jb2007/osjb2007-02-03-perens.pdf>, S.144, (Letzter Zugriff: 26.05.2020).

²⁶ <http://www.opensourcejahrbuch.de/download/jb2007/osjb2007-02-03-perens.pdf>, S.154-156, (Letzter Zugriff: 26.05.2020).

Vermittlung in den Vordergrund. Freie Software Lizenzen qualifizieren sich als Open Source Software, jedoch ist dies andersherum nicht immer gewährleistet.

Im Gegensatz zu Freier Software stellt Open Source Software ein ökonomisches Entwicklungsmodell dar.²⁷ Open Source Software wurde geschaffen um Geschäftsleute anzusprechen.²⁸ Sie zeigt die unmittelbaren praktischen Vorzüge für die kommerzielle Nutzung in Firmen auf, ohne dabei auf Themen wie Freiheit und Ethik einzugehen, da diese meist Unbehagen auslösen und deshalb gemieden werden. Für Open Source Software handelt es sich bei proprietärer Software um eine praktische Frage und wird als suboptimale Lösung angesehen. Open Source Software schließt zwar auch Freie Software mit ein, jedoch auch einige proprietäre Programme wie z.B. Qt bis zum Jahr 2008.

Des Weiteren sind einige Open Source Lizenzen beschränkt, da sie es beispielsweise nicht gestatten, modifizierte Versionen einer Software zu erstellen und diese privat zu nutzen. Sollte der Quellcode einer Software eine Lizenz ohne Copyleft ausweisen, so können die ausführbaren Dateien zusätzliche Bedingungen enthalten. Damit gelten sie als Open Source, wenn sie dem freigegebenen Quellcode vollständig entsprechen, jedoch nicht als Freie Software. Viele Open Source Software-Produkte erhalten Prüfsignaturen von Rechnern in ihren Programmen, zur Beschränkung der Installation abweichender ausführbarer Dateien. Die Beschränkung wird als "Tivosierung" bezeichnet. Auch wenn die Software aus freiem Quellcode erstellt wurde und somit über eine freie Lizenz verfügt, ist es dem Benutzer nicht möglich eine modifizierte Version des Programms auszuführen, was die ausführbare Datei also unfrei macht. Ein Beispiel hierfür stellt "Microsoft Visual Studio" dar.²⁹

Anfangs wurde Open Source Software in der Wirtschaft als nicht zukunftsfähig angesehen. Die Annahme war, dass es nicht in der Lage sein würde, die Bedürfnisse des Marktes nach neuer Technologie zu befriedigen.³⁰

²⁷ <http://www.opensourcejahrbuch.de/download/jb2007/osjb2007-02-03-perens.pdf>, S.131, (Letzter Zugriff: 26.05.2020).

²⁸ <http://www.opensourcejahrbuch.de/download/jb2007/osjb2007-02-03-perens.pdf>, S.134, (Letzter Zugriff: 26.05.2020).

²⁹ <http://www.gnu.org/philosophy/open-source-misses-the-point.html> (Letzter Zugriff: 26.05.2020).

³⁰ <http://www.opensourcejahrbuch.de/download/jb2007/osjb2007-02-03-perens.pdf>, S.132, (Letzter Zugriff: 26.05.2020).

5. Fazit

Richard Stallmans bemerkenswertes Vordenken im Bereich der Softwareentwicklung stieß auf regen Zuspruch, da es die Softwarenutzer unabhängig vom Zwang durch proprietäre Entwickler machen sollte. Da Stallman selbst Programmierer war, sprach seine Sichtweise, wie Software behandelt werden sollte, auch zur damaligen Zeit zum großen Teil nur die Programmierer/Entwickler von Software an. Doch der heutige Standardnutzer, der hin und wieder Programme zur Textverarbeitung, für E-Mail und einige Internetdienste wie z.B. Amazon nutzt, was einen Großteil darstellt, bleibt davon unberührt. Meist fällt seine Wahl, ob er nun Open Office oder Microsoft Office nutzen soll, auf Open Office, nicht weil ihm ein offener Quelltext oder der Wert der Freiheit wichtig ist, sondern aufgrund des Preises.

Auch wird deutlich, dass nicht jeder Entwickler/Nutzer freier Software Stallmans Sichtweise teilte, was man an der Bildung der Open Source Bewegung erkennen kann. Raymonds und Perens Versuch das Konzept freier Software auch in der Wirtschaft zu etablieren, und eben neben den Entwicklern auch Geschäftsleute anzusprechen, war geglückt, da sich der Begriff "Open Source" eben von Ethik, Freiheit und sozialen Problemen los sagte und nur die unmittelbaren Vorteile des Modells in den Vordergrund rückte.

Anhand des vorgestellten Vergleichs fällt auf, dass Open Source Software und Freie Software mehr Gemeinsamkeiten als Unterschiede aufweisen. Das lässt sich damit erklären, dass Freie Software einen Teil der Open Source Software darstellt. Die wichtigsten Unterschiede resultieren aus den unterschiedlichen Philosophieansätzen. Steht bei freier Software die Ethik im Vordergrund, konzentriert es sich bei Open Source Software auf die Wirtschaftlichkeit. Gemeinsam haben beide Softwaremodelle, dass die Freiheit im Vordergrund steht.