

Urheberrechtliche Fragen von Open Source Software

von Jens Nödler

Seminararbeit im Rahmen des Blockseminars zum Multimedia- und Telekommunikationsrecht bei **Prof. Dr. Spindler** an der Georg-August-Universität Göttingen



Autor: Jens Nödler

Student der Angewandten Informatik im 4. Semester

Kontakt: <http://noedler.de> - Göttingen im Mai 2004

Inhaltsverzeichnis

1. Vorwort.....	3
2. Einführung in die Open Source Software.....	4
2.1. Software-Grundlagen: vom Quelltext zum Binärprogramm.....	4
2.2. Software-Grundlagen: Betriebssysteme.....	4
2.3. Die Idee von Open Source Software (GNU).....	5
2.4. Abgrenzung von anderen Software-Vertriebsmodellen.....	7
3. Einführung in das Urheberrecht.....	8
3.1. Urheber und Verwertungsrechte.....	8
3.2. Ausschließliche und einfache Nutzungsrechte.....	9
3.3. Bearbeitung eines Werks, Werkverbindung und Miturheber.....	9
4. Lizenzmodelle für Open Source Software.....	11
4.1. GNU General Public License - ein Lizenzmodell mit Copyleft.....	11
4.2. Lizenzmodelle mit schwachem und ohne Copyleft: BSD- und Mozilla- Lizenzen.....	13
5. Open Source Software unter GPL und das deutsche Urheberrecht.....	15
5.1. Einordnung der GPL in das Urheberrecht.....	15
5.1.1. Schutzgegenstand des UrhG und der GPL.....	15
5.1.2. Vervielfältigung, Nutzung und Verbreitung.....	16
5.1.3. Bearbeitung von Open Source Software.....	17
5.1.4. Neue Nutzungsarten und die GPL.....	19
5.1.5. Urheberpersönlichkeitsrecht und GPL.....	20
5.1.6. Urheber- und Miturheberschaft.....	21
5.1.7. Verzicht auf die Verwertungsrechte oder einfaches Nutzungsrecht an jedermann?	21
5.2. Aus der GPL abgeleitete Werke.....	22
5.3. Möglichkeiten der Doppellizenzierung.....	23
5.4. Entwicklung von OSS im Arbeitsverhältnis.....	24
6. Fazit.....	27
7. Literaturverzeichnis und Erklärung.....	28

1. Vorwort

Das Ziel dieser Seminararbeit ist die urheberrechtliche Einordnung von freier *Open Source Software* und die Beantwortung daraus entstehender rechtlicher Fragestellungen.

Zu Beginn werden die Grundlagen der Softwareentwicklung und die Entstehung der Open Source Idee erläutert, um anschließend mit einem Einstieg in das deutsche Urheberrecht fortzufahren.

Nach der Vorstellung der wichtigsten Open Source Software-Lizenzen wird im Hauptteil eine grundlegende urheberrechtliche Einordnung der wichtigsten Open Source Software-Lizenz *GPL* vorgenommen, um anschließend auf einige Fragestellungen aus dem Alltag von Open Source Software-Programmierern einzugehen.

2. Einführung in die Open Source Software

2.1. Software-Grundlagen: vom Quelltext zum Binärprogramm

Die Entwicklung von Software besteht im Wesentlichen aus zwei Arbeitsschritten: Zuerst wird der Quelltext in einer Programmiersprache¹ erstellt und im zweiten Schritt wird dieser Quelltext von einem Compiler² in ein von Computern ausführbares Binärprogramm umgewandelt.

Der Programmierer beschreibt die gewünschte Funktionalität des Programms im Quelltext, wobei meist so genannte Hochsprachen³ zum Einsatz kommen. Dieser Quelltext kann aber von Computern nicht direkt ausgeführt werden. Daher wird ein Compiler benutzt, um daraus computerlesbare Binärprogramme zu erstellen.

Die Vorteile dieser Zweiteilung liegen erstens in der effizienten Softwareentwicklung mit Hochsprachen und zweitens in der Wiederverwendbarkeit von Quelltexten auf verschiedenen Rechnerarchitekturen⁴, da der Quelltext nur neu kompiliert statt neu geschrieben werden muss.

2.2. Software-Grundlagen: Betriebssysteme

Ein Betriebssystem ermöglicht den Betrieb eines Computers und stellt die Schnittstelle zwischen Mensch und Maschine dar. Die Hauptbestandteile sind der Kern⁵ und Hilfsprogramme, die die einfache Benutzung des Systems ermöglichen. Die zentralen Aufgaben sind die Verwaltung von Ressourcen⁶, die Ein-

1 Definition: „Eine Programmiersprache ist eine formale Sprache zur Darstellung (Notation) von Computerprogrammen. Sie vermittelt dem Computersystem durch von einem Menschen geschriebenen Text genaue Angaben zu einer Kette von internen Verarbeitungsschritten, beteiligten Daten und deren Struktur in Abhängigkeit von internen oder externen Ereignissen.“ Quelle: <http://de.wikipedia.org/wiki/Programmiersprache>

2 Definition: „Ein Compiler (auch Kompilierer oder Übersetzer) ist ein Computerprogramm, das ein in einer Quellsprache geschriebenes Programm in ein semantisch äquivalentes Programm einer Zielsprache umwandelt.“ Quelle: <http://de.wikipedia.org/wiki/Compiler>

3 Hochsprachen sind Programmiersprachen (wie zum Beispiel C, C++ oder Java), die infolge ihres hohen Abstraktionsgrades und gleichzeitiger Verständlichkeit eine effektive Softwareentwicklung ermöglichen. Im Gegensatz dazu finden hardwarenahe Sprachen (wie Assembler) nur in Spezialfällen Anwendung.

4 Rechnerarchitekturen beschreiben die grundsätzliche Implementierung des Computers. Alle modernen Architekturen basieren auf der „von Neumann-Architektur“. Konkrete Implementierungen ist zum Beispiel die i386-Architektur von Intel.

5 engl. Kernel, zum Beispiel der Linux-Kernel als Kern des Betriebssystems GNU/Linux

6 Ressourcen wie Rechenleistung, Hauptspeicher und Festplattenspeicher

und Ausgabe von Daten und die Ausführung von Anwendungsprogrammen. Betriebssysteme stellen grundlegende Funktionen⁷ bereit, die von Anwendungsprogrammen benutzt werden und somit die Entwicklung von Anwendungsprogrammen deutlich erleichtern.

Nicht nur auf Computern, sondern auf nahezu allen elektronischen Geräten⁸ verrichten heutzutage Betriebssysteme in verschiedensten Ausführungen⁹ ihren Dienst. Die bekannten Vertreter *Microsoft Windows* und *Linux* stellen nur die Spitze des Eisberges der Betriebssystem-Welt dar.

2.3. Die Idee von Open Source Software (GNU)

Der englische Ausdruck *Open Source Software* (OSS) bedeutet einerseits die freie¹⁰ Verfügbarkeit und Nutzbarkeit des Quelltextes einer Software und andererseits die Bindung an folgende grundlegende Regeln¹¹:

- Die Software liegt als Quelltext in einer verständlichen Form vor.
- Die Software darf beliebig kopiert, verbreitet und genutzt werden.
- Die Software darf verändert und weitergegeben werden.

Um OSS zu verstehen, ist ein kurzer Ausflug in die Geschichte der Softwareentwicklung hilfreich.

In der Anfangszeit der Softwareentwicklung, die hauptsächlich an Universitäten im Rahmen von Forschungsarbeiten stattfand, war es selbstverständlich, die Quelltexte jedem Interessierten zugänglich zu machen. Auch im kommerziellen Umfeld war es durchaus üblich, Programme mit Quelltexten auszuliefern, um Anpassungen durch die Kunden zu ermöglichen¹².

7 in Form von Programmbibliotheken wie zum Beispiel für das Lesen und Schreiben von Dateien oder für die Kommunikation in einem Netzwerk

8 Mobiltelefone, Personal Digital Assistants (PDAs), mobile Musikwiedergabegeräte, Steuerungen von Industrieanlagen, Bankautomaten, etc. benutzen Betriebssysteme

9 Eine Übersicht liefert: http://de.wikipedia.org/wiki/Liste_der_Betriebssysteme

10 Wobei das Wort „frei“ nicht im Sinn von „kostenlos“, sondern im Sinn von „Freiheit“ zu verstehen ist. siehe Präambel der GPL: „When we speak of free software, we are referring to freedom, not price.“

11 Quelle der Regeln: http://de.wikipedia.org/wiki/Open_source

Eine weitere OSS-Definition findet sich unter: <http://www.opensource.org/docs/definition.php>

12 siehe Schiffner, S. 57 ff

Die University of California at Berkeley besaß seit 1975 Kopien der Quelltexte des Betriebssystems UNIX, das an den Bell Laboratories von AT&T entwickelt wurde, mit der Erlaubnis, den Quellcode einzusehen und zu modifizieren¹³. Schnell fanden sich interessierte Entwickler¹⁴, die das System unter der Koordination der Universität zu einem einsatzfähigen Betriebssystem ausbauten. Da eine kommerzielle Vermarktung dieser Forschungsergebnisse nicht erlaubt war, stellte die Universität UNIX unter die neu geschaffene *Berkeley Software Distribution Lizenz* (BSD)¹⁵, die den Nutzern weitreichende Rechte einräumte: die Verbreitung und Veränderung der Software waren mit Ausnahme der kommerziellen Verwertung erlaubt. Somit stellt die BSD-Lizenz die erste OSS-Lizenz der Geschichte dar.

Als AT&T 1984 im Zuge des Kartellprozesses aufgeteilt wurde, wurde ein kommerzieller Vertrieb des UNIX-Systems möglich¹⁶ und das ehemals freie System wurde zu proprietärer¹⁷ Software, die nur noch gegen Bezahlung erhältlich war. Dieser Schritt verärgerte viele der freiwilligen, engagierten Entwickler von UNIX nachhaltig.

Als Reaktion auf die „UNIX-Pleite“ und die Tatsache, dass im Zuge der Kommerzialisierung des Softwaremarktes immer seltener die Quelltexte von Programmen verfügbar waren¹⁸, gründete *Richard Stallman* 1984 die *Free Software Foundation*¹⁹, mit der Zielsetzung ein komplettes und vollkommen freies Betriebssystem zu erstellen. Das *GNU Project*²⁰ benannte Vorhaben erforderte eine Lizenz, die es ermöglicht, grundlegende Regeln von OSS (siehe oben) einzubeziehen und gleichzeitig die Überführung in proprietäre Software (so wie es bei UNIX passierte) zu verhindern. Das Ergebnis lag 1989 in

13 Quelle: http://de.wikipedia.org/wiki/Berkeley_Software_Distribution

14 Wobei die Motivation zur unbezahlten Mitarbeit an OSS-Projekten vielfältig ist: sie reicht von Idealismus über Neugier bis zu Anerkennung in der OSS-Gemeinde.

15 Die Lizenz im Volltext: <http://de.wikipedia.org/wiki/BSD-Lizenz>

16 Bis 1984 war es AT&T aus kartellrechtlichen Gründen nicht möglich, in den Softwaremarkt einzusteigen.

17 Hier in der Bedeutung, dass es sich *nicht* um freie Software handelt.

18 vergleiche: Grassmuck, „Einleitung“

19 Abkürzung: „FSF“, Website: <http://www.fsf.org>

20 GNU steht als Abkürzung für „GNU's not UNIX“ als deutliche Abgrenzung zum AT&T UNIX
Website: <http://www.fsf.org/gnu/thegnuproject.html>

Form der *GNU General Public License*²¹ vor, die die geforderten Bedingungen erfüllt und auf die ausführlicher im 4. Abschnitt dieser Arbeit eingegangen wird.

2.4. Abgrenzung von anderen Software-Vertriebsmodellen

Für Endanwender sind die wichtigsten Punkte bei dem Einsatz von OSS die kostenlose Verfügbarkeit und die Möglichkeit die Software den eigenen Vorstellungen anzupassen, was den Einsatz auch im professionellen Umfeld sehr interessant macht²².

Davon unterscheiden sich die traditionellen Software-Vertriebsmodelle wie kommerzielle Software, Freeware und Shareware deutlich²³. Allen drei Modellen ist gleich, dass der Quelltext im Allgemeinen nicht verfügbar ist. Free- und Shareware werden oft über das Internet kostenlos angeboten, wobei die Nutzung von Freeware zeitlich unbegrenzt ist, aber in manchen Fällen der kommerzielle Einsatz nicht erlaubt ist. Shareware ist der Zwischenschritt von Freeware zu kommerzieller Software, da das Nutzungsrecht zeitlich begrenzt gewährt wird. Nach dem Ablauf einer Zeitspanne²⁴ muss eine Lizenz erworben werden, wenn das Programm weiterhin genutzt werden soll. Die Begriffe Free- und Shareware besitzen keine eindeutige Definition wie zum Beispiel OSS unter der GPL, so dass im Zweifelsfall eine Einzelprüfung des Lizenzvertrages erforderlich sein kann²⁵.

Das Nutzungsrecht von kommerzieller Software ist immer an den Erwerb der passenden Lizenz gekoppelt und nur in seltenen Fällen wird Einblick in die Quelltexte²⁶ gewährt, da dieser das wichtigste Kapital der Hersteller von „Closed Source Software“ ist.

21 Die Lizenz im Volltext: <http://www.fsf.org/licenses/gpl.html>

22 siehe Schiffner, S. 20 ff

23 Übersichtliche schematische Einordnung der verschiedenen Software-Vertriebsmodelle finden sich bei Spindler, S. 19 und in „Open-Source-Software - Ein Leitfaden für kleine und mittlere Unternehmen“, S. 40

24 oft erlaubt der Autor der Shareware einen Testzeitraum von 30 Tagen

25 vergleiche: <http://de.wikipedia.org/wiki/Freeware> und <http://de.wikipedia.org/wiki/Shareware>

26 zum Beispiel erlaubt Microsoft Regierungsbehörden den Einblick in die Quelltexte einiger Programme im Rahmen des *Government Security Program*

Quelle: <http://www.microsoft.com/resources/sharedsource/Licensing/GSP.msp>

3. Einführung in das Urheberrecht

3.1. Urheber und Verwertungsrechte

Das Urheberrecht²⁷ dient dem Schutz der Schöpfer von Geisteswerken vor der Entstellung und vor der wirtschaftlichen Ausbeutung ihrer Werke. Das UrhR zerfällt daher in die beiden Bestandteile *Verwertungsrecht* (materiell) und *Urheberpersönlichkeitsrecht* (ideell), wobei Letzteres als Konkretisierung des allgemeinen Persönlichkeitsrechts (Art 2 I GG i.V.m. Art 1 I GG) zu verstehen ist²⁸.

Das UrhG stellt die Urheber von Werken der Literatur, Wissenschaft und Kunst unter Schutz, falls es sich dabei um persönliche geistige Schöpfungen nach § 2 II UrhG handelt, wobei sich Computerprogramme nach § 2 I, Nr. 1 UrhG bei den Sprachwerken wiederfinden und damit ebenfalls den Schutz des UrhR genießen. Weitere Kriterien sind die sinnliche Wahrnehmbarkeit des Werks und eine erforderliche Schöpfungshöhe, die allerdings für Computerprogramme mit der Einführung des §§ 69a ff.²⁹ UrhG soweit abgesenkt wurde, dass heute nahezu³⁰ alle Programme den Schutz des UrhG genießen³¹, was als „kleine Münze“ des Urheberrechts bezeichnet wird.

Eine wichtige Besonderheit des deutschen UrhR ist die feste Bindung des Urhebers an sein Werk und daraus resultierend die Unübertragbarkeit des Urheberrechts nach § 29 I UrhG im Gegensatz zum *Copyright* im englisch-amerikanischen Rechtssystem³². Dem Urheber ist es aber dafür möglich, Nutzungsrechte zu übertragen.

Die Verwertungsrechte des Urhebers sind nach § 15 I und II UrhG wiederum in körperliche und unkörperliche Verwertungsrechte unterteilt, wobei uns besonders die körperlichen Verwertungsrechte *Vervielfältigungsrecht* nach § 16 UrhG und *Verbreitungsrecht* nach § 17 I UrhG interessieren. Das Vervielfältigungs-

27 „Gesetz über Urheberrecht und verwandte Schutzrechte“ von 1965

28 siehe Fechner, S. 98 ff, Rdnr. 310, 313

29 § 69a ist im Rahmen der EG-Richtlinie 91/250/EWG (des Rates über den Rechtsschutz von Computerprogrammen vom 14. Mai 1991) in das UrhG im Abschnitt 8 (Besondere Bestimmungen für Computerprogramme) eingeflossen.

30 lediglich so genannte „Trivialprogramme“ sind davon ausgeschlossen

31 ausführlich: Schiffner, S. 44 ff

32 vergleiche: <http://de.wikipedia.org/wiki/Copyright>

fältigungsrecht erlaubt dem Urheber, Kopien seines Werks herzustellen, wobei der Paragraph technikneutral gestaltet ist, so dass auch „die Übersendung eines Werks im Internet an eine andere Internetadresse eine Vervielfältigung“³³ darstellt. Davon zu unterscheiden ist das Verbreitungsrecht, welches dem Urheber erlaubt das Original oder eine Kopie der Öffentlichkeit anzubieten oder in Verkehr zu bringen.

3.2. Ausschließliche und einfache Nutzungsrechte

Dem Urheber ist es möglich, Dritten *ausschließliche* oder *einfache* Nutzungsrechte an seinem Werk zu übertragen. Das ausschließliche Nutzungsrecht berechtigt einen Dritten unter Ausschluss aller anderen Personen³⁴ zur Nutzung des Werks³⁵. Bei der Übertragung eines einfachen Nutzungsrechts auf einen Dritten bestimmt der Urheber die Art und den Umfang dieses Rechts. Der Urheber kann nach § 31 I UrhG eine beliebige Anzahl einfacher Nutzungsrechte (auch Lizenzen genannt) vergeben, ohne auf bereits an Dritte vergebene Lizenzen achten zu müssen.

Der zwischen Urheber und einem Dritten ausgehandelte Umfang eines einfachen Nutzungsrechts wird in dem so genannten Lizenzvertrag festgelegt.

3.3. Bearbeitung eines Werks, Werkverbindung und Miturheber

Die Normen der §§ 3, 9 UrhG beschäftigen sich mit der Bearbeitung eines urheberrechtlich geschützten Werks und der Erstellung eines verbundenen Werks. Im Kontext der Open Source Software unter GPL kommen beide nach hM nicht zum Tragen, da § 3 der Genehmigung des Urhebers zur Veröffentlichung und Verwertung der Bearbeitung bedarf³⁶ und § 9 „steht und fällt [...] mit der Unabhängigkeit der einzelnen Werke, die

33 Fechner, S. 108, Rdnr. 361

34 einschließlich des Urhebers selbst

35 siehe UrhG, § 31 III, Besonderheit: Möchte der Inhaber des ausschließlichen Nutzungsrechts selbst einfache Nutzungsrechte vergeben, so bedarf dies trotzdem noch der Genehmigung des Urhebers nach § 34 I UrhG.

36 siehe UrhG, § 23 „Bearbeitungen oder andere Umgestaltungen des Werkes dürfen nur mit Einwilligung des Urhebers des bearbeiteten oder umgestalteten Werkes veröffentlicht oder verwertet werden.“

miteinander verbunden werden“³⁷, denn die Bestandteile der Werkverbindung sind im OSS-Bereich oftmals nicht selbstständig lauffähig, was aber wiederum von §§ 69 ff. UrhG gefordert wird.

In Bezug auf OSS ist die Miturheberschaft nach § 8 UrhG von Bedeutung, die Werke umfasst, die von mehreren geschaffen wurden und deren Anteile sich nicht gesondert verwerten lassen. Eine weitere Voraussetzung für die Miturheberschaft ist, „dass die Mitwirkung an einem Werk erfolgt, das nach einem gemeinsamen Plan bzw. einer gemeinsamen Idee entworfen worden ist“³⁸, was regelmäßig zumindest auf *Maintainer*³⁹ und Initiatoren eines Softwareprojekts zutrifft⁴⁰.

37 Spindler, S. 24, Verbundes Werk

38 Spindler, S. 22, Miturheber nach § 8 UrhG

39 Als *Maintainer* werden die leitenden Personen eines Softwareprojekts bezeichnet.

40 vergleiche: Schiffner, S. 122

4. Lizenzmodelle für Open Source Software

4.1. GNU General Public License - ein Lizenzmodell mit Copyleft

Eine Einführung in die Entstehung der GNU General Public License gab bereits der zweite Abschnitt dieser Arbeit. Jetzt soll es darum gehen, die GPL im Detail zu untersuchen.

Die GPL setzt das Copyright (das Urheberrecht) des Urhebers geschickt ein, um dem Nutzer weitreichende Rechte zu gewähren und ihn zugleich an bestimmte Pflichten zu binden. Dies bedeutet konkret:

- Es dürfen Kopien der Software⁴¹ erstellt und verbreitet werden, solange die Copyright-Hinweise im Rahmen der GPL erhalten bleiben⁴².
- Die Kopie der Software darf verändert werden, solange die Änderungen veröffentlicht und wieder unter die GPL gestellt werden⁴³. Weiterhin darf kein Entgelt für die Lizenzierung selbst verlangt werden.

Entgegen der häufig anzutreffenden Meinung, dass es an Open Source Software kein Urheberrecht gäbe⁴⁴, ist das genaue Gegenteil der Fall: die urheberrechtlichen Verwertungsrechte werden dazu gebraucht, die Nutzer wiederum an die GPL zu binden. Damit wird erreicht, dass OSS unter GPL niemals unter eine andere Lizenz gestellt werden kann und so zu proprietärer Software werden könnte⁴⁵. Dieser Effekt wird als *Copyleft*⁴⁶ bezeichnet.

Die GPL sieht für den Verstoß gegen das Copyleft-Gebot deutliche Regelungen vor: Der § 4 GPL stellt zuerst fest, dass

41 *Software* bezieht sich hierbei auf den Quelltext ebenso wie auf Binärprogramme. siehe GPL, § 3: „[...] in object code or executable form [...] with the complete corresponding machine-readable source code [...]“

42 siehe GPL, § 1: „You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; [...]“

43 siehe GPL, § 2: „You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, [...]“

44 dazu genauer: Spindler, S. 8

45 vergleiche: Schiffner, S. 78 ff

46 „Copyleft“ ist ein von der *Free Software Foundation* geprägter Begriff.

siehe: „Was ist das Copyleft?“ unter <http://www.gnu.org/copyleft/copyleft.de.html>

GPL-Software nicht unter andere Lizenzen gestellt werden darf⁴⁷ und benutzt zweitens als Sanktion den Entzug der durch die GPL gewährten Rechte im Fall der Zuwiderhandlung⁴⁸. Praktisch bedeutet dies, dass der Autor einer GPL-Software, der diese unter eine fremde Lizenz stellt, selbst die durch die GPL gewährten Rechte an seiner Software verliert.

Die Tatsache, dass bei der Entwicklung von Software nahezu immer auf Programmbibliotheken zurückgegriffen wird, machte die Entwicklung der *GNU Lesser GPL*⁴⁹ notwendig. Die Programmbibliotheken werden bei dem Kompilervorgang mit dem Binärprogramm verbunden⁵⁰, so dass das Binärprogramm unter GPL stehen müsste, wenn die Bibliotheken unter GPL stehen würden.

Praktisch würde dies bedeuten, dass zum Beispiel mit dem freien Compiler *GCC*⁵¹ nur GPL-Programme erzeugt werden könnten, was dessen Einsatzbereich unnötig einschränken würde. Daher erlaubt die LGPL die Ausnahme, das Copyleft-Gebot für die Erzeugung von Binärprogrammen, die Teile der Bibliotheken enthalten⁵², auszusetzen. Die Free Software Foundation versprach sich mit der Einführung der LGPL, die Verbreitung freier Software zu fördern⁵³.

Die LGPL ist der GPL inhaltlich sehr ähnlich, bietet aber entsprechend im § 6 die eben beschriebene Ausnahme. Daher „ist diese eine echte Copyleft-Lizenz“⁵⁴, denn der Nutzer ist nach § 2 LGPL ebenfalls an die von der GPL bekannten Pflichten gebunden⁵⁵.

-
- 47 siehe GPL, § 4: „You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License.“
- 48 siehe GPL, § 4: „Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License.“
- 49 Abkürzung: LGPL, Die Lizenz im Volltext: <http://www.fsf.org/licenses/lgpl.html>
- 50 Dieses „Verbinden“ wird oft als „linken“ bezeichnet und meint das Verknüpfen des eigentlichen Programms mit den Bibliotheken des Compilers und/oder des Betriebssystems. Details unter: <http://de.wikipedia.org/wiki/Linker>
- 51 „GNU Compiler Collection“. Ein freier Compiler für „C, C++, Objective-C, Fortran, Java, and Ada“, dessen Bibliotheken unter LGPL stehen. Website: <http://gcc.gnu.org/>
- 52 siehe LGPL, § 6: „[...] a work containing portions of the Library [...]“
- 53 siehe Präambel der LGPL und Schiffner, S. 82 ff
- 54 Schiffner, S. 85, 1. Absatz
- 55 Wobei die LGPL als Besonderheit auch die Option ermöglicht in das Lizenzmodell der GPL zu wechseln. siehe LGPL, § 3 und Spindler, S. 15

4.2. Lizenzmodelle mit schwachem und ohne Copyleft: BSD- und Mozilla-Lizenzen

Im zweiten Abschnitt dieser Arbeit wurde die *Berkeley Software Distribution* (BSD) bereits als erste OSS-Lizenz der Geschichte vorgestellt. Auch sie erlaubt die Verbreitung von Software „in jeder Form, mit und ohne Veränderungen, sofern ein Copyright-Vermerk und die Lizenzbestimmung angefügt werden“⁵⁶. Im Gegensatz zur GPL besitzt die BSD-Lizenz nicht den „Viral Effect“⁵⁷ der GPL, so dass die Überführung in proprietäre Software - wie im Fall von UNIX passiert - möglich ist. Die BSD-Lizenz bzw. Derivate⁵⁸ der Lizenz sind heute hauptsächlich im Bereich der freien UNIX-Nachfolger NetBSD, OpenBSD und FreeBSD⁵⁹ zu finden.

Die *Mozilla Public License*⁶⁰ (MPL) entstand im Zuge der Freigabe der Quelltexte des Internetbrowsers Netscape im Jahre 1998⁶¹, weil dieser unter der restriktiven *Netscape Public License*⁶² (NPL) veröffentlicht werden sollte. Die NPL sah vor, dass Netscape die Quelltexte auch außerhalb der NPL für andere Produkte verwenden darf, was auf heftige Kritik der Open Source Gemeinde stieß. Daraufhin entstand die MPL, die bis auf die Sonderregelungen für Netscape wortgleich mit der NPL ist. Ansonsten sind die Rechte der MPL mit denen der GPL vergleichbar: die Nutzung, Verbreitung, und Veränderung sind erlaubt. Allerdings erlaubt die MPL einen erweiterten Vertrieb auch gemeinsam mit proprietärer Software, was bei der GPL in dieser Form nicht möglich ist. Ist Software unter MPL freigegeben, so müssen auch die Änderungen wieder unter MPL gestellt werden. Neu zu einem MPL-Projekt hinzugefügte Dateien können jedoch unter einer beliebigen Lizenz veröffentlicht

56 Schiffner, S. 85, 3. Absatz

57 Als „Viral Effect“ wird das Copyleft der GPL bezeichnet, weil bereits die Verwendung von wenigen Zeilen GPL-Codes ausreichen, um die restliche Software zu infizieren und die gesamte Software unter GPL gestellt werden muss.

58 zum Beispiel basiert die Apache Lizenz auf der BSD-Lizenz: <http://apache.org/licenses/>

59 Eine Übersicht über verschiedene UNIX-Varianten bietet: <http://de.wikipedia.org/wiki/Unix>

60 Die Lizenz im Volltext: <http://mozilla.org/MPL/MPL-1.1.html>

61 <http://wp.netscape.com/newsref/pr/newsrelease591.html>
<http://www.heise.de/newsticker/meldung/2183/>

62 Die Lizenz im Volltext: <http://mozilla.org/MPL/NPL-1.1.html>

werden⁶³. Insgesamt ist der Copyleft-Einfluss also nicht so stark ausgeprägt wie bei der GPL⁶⁴.

Es existieren viele weitere OSS-Lizenzen, die meist in der Konzeption den hier vorgestellten Modellen ähnlich sind und davon nur in Details abweichen. Da die GNU GPL einen beherrschenden Marktanteil von mindestens 75% besitzt⁶⁵, wird im weiteren Verlauf hauptsächlich auf die GPL Bezug genommen.

-
- 63 vergleiche: „Files which fall under the MPL because they are or contain Modifications must be made available as detailed in the license [...]. Other files may be kept proprietary.“
Quelle: <http://mozilla.org/MPL/mpl-faq.html>
- 64 vergleiche: „The MPL has a limited amount of 'copyleft' - more copyleft than the BSD family of licenses, which have no copyleft at all, but less than the LGPL or the GPL.“
Quelle: <http://mozilla.org/MPL/mpl-faq.html>
- 65 Hang/Hohensohn, S. 21

5. Open Source Software unter GPL und das deutsche Urheberrecht

In diesem Teil wird auf die urheberrechtlichen Aspekte von OSS unter GPL eingegangen. Auf weitere Problembereiche, die sich im deutschen Recht bei dem Einsatz von Open Source Software ergeben, wie zum Beispiel im Patent-, Vertrags-, Haftungs- und Gewährleistungsrecht, soll im Rahmen dieser Arbeit nicht näher eingegangen werden.

5.1. Einordnung der GPL in das Urheberrecht

Die Notwendigkeit der Einordnung der GPL in das deutsche Urheberrecht ergibt sich aus der Tatsache, dass sich die GPL stark an das US-amerikanische Lizenzvertragsrecht anlehnt und die Gültigkeit der einzelnen Bestimmungen der GPL gegen die entsprechenden Normen des UrhG geprüft werden müssen.

Auch wenn die GPL nach allen Regeln der Kunst juristisch wasserdicht konstruiert wurde, darf jedoch nicht vergessen werden, dass alle hier getroffenen urheberrechtlichen Bewertungen noch nicht endgültig gerichtlich bestätigt wurden⁶⁶.

Daher ist es besonders spannend, dass erst vor kurzem das Münchner Bezirksgericht eine einstweilige Verfügung gegen den Router-Hersteller *Sitecom* ausgesprochen hat⁶⁷, der die GPL-Software des Open Source-Projekts *iptables/netfilter*⁶⁸ einsetzt, ohne sich an die Bestimmungen der GPL zu halten.

5.1.1. Schutzgegenstand des UrhG und der GPL

Der Schutzgegenstand des UrhG im Bezug auf Computerprogramme ist sehr umfangreich. § 69a I und II schützen Programme in jeder Ausdrucksform und zugleich das Entwurfsmaterial, was Dokumentation, Schnittstellen und die Idee hinter dem Programm umfasst, solange die Grundlagen des UrhG eingehalten werden. Daher ist es auch hier nötig, dass das Werk ein Produkt einer geistigen Schöpfung ist. Die in der GPL genannten Erscheinungsformen als Quelltext und Binärpro-

66 Metzger in „Freie Software – Rechtlich sicher?“ und Grassmuck, „GPL und deutsches Recht“

67 Tickermeldung „Open-Source-Projekt erwirkt Verfügung wegen GPL-Verletzung“ vom 15.04.2004, Quelle: <http://www.heise.de/newsticker/meldung/46551>

68 Website: <http://www.netfilter.org/>

gramm („object code“) fallen somit in den Wirkungsbereich der Regelungen des UrhG⁶⁹.

Obwohl die GPL die Nutzungsrechtseinräumung für das Entwurfsmaterial (Dokumentation) nicht explizit behandelt, geht *Spindler* davon aus, dass „der Urheber verpflichtet sein [wird], der Änderung der Dokumentation zuzustimmen, wenn schon Änderungen an der Software selbst unter der GPL erlaubt werden“⁷⁰. *Schiffner* geht etwas differenzierter vor und begutachtet die Vervielfältigung und Bearbeitung der Dokumentation nach § 23 UrhG und stellt fest, dass dafür die Einwilligung des Urhebers eingeholt werden müsste, was dieser nach Treu und Glauben nicht verweigern dürfe, woraus sich ableiten lässt, dass auch die Vervielfältigung und Bearbeitung der Dokumentation von GPL Programmen in Einzelfällen möglich ist. Insgesamt stellt sich das Problem nicht als sonderlich schwerwiegend heraus, da die GPL nicht zur Dokumentation der Programme verpflichtet und in der Praxis das Hauptaugenmerk nicht auf die Pflege der Dokumentation gelegt wird⁷¹.

5.1.2. Vervielfältigung, Nutzung und Verbreitung

Die Frage, ob die Ausführung eines Programms und das damit verbundene Laden des Binärprogramms in den Hauptspeicher bereits eine urheberrechtliche Handlung (Vervielfältigung) darstellt, wird heute nach hM bejaht⁷². Für diese Vervielfältigung ist nach §§ 16, 69c Nr. 1 UrhG die Zustimmung des Urhebers notwendig. Grundlage hierfür kann nur ein geschlossener Lizenzvertrag (hier die GPL) sein.

Ist zwischen Urheber und Nutzer die GPL als Lizenzvertrag gültig abgeschlossen, so gewährt die GPL selbstverständlich auch die Ausführung des Programms als elementaren Bestandteil der gewährten Rechte und schränkt die Ausführung und damit die Nutzung des Programms nicht ein⁷³.

Der § 69d UrhG, der hauptsächlich die Rechte von Lizenznehmern proprietärer Software stärkt (wie zum Beispiel das

69 vergleiche: Schiffner, S. 113

70 Spindler, S. 27

71 Schiffner, S. 114

72 ausführlich: Schiffner, S 124 ff. und Spindler, S. 44

73 siehe GPL, § 0: „The act of running the Program is not restricted [...]“

Recht zum Anfertigen einer Sicherungskopie), steht in keinem Konflikt mit der GPL, da diese Rechte in weit größerem Umfang gewährt als § 69d UrhG.

Insgesamt „kann von einem Nutzungsrecht ausgegangen werden“⁷⁴, da die Nutzung und das dadurch implizierte Recht auf Vervielfältigung bei einem bestehenden GPL-Vertrag einschlägig sind.

Die Verbreitung von Software unterliegt den §§ 17 I, 69c Nr. 3 UrhG und ist ebenfalls von der Genehmigung des Urhebers abhängig. Unter der Verbreitung ist sowohl das Anbieten als auch das Inverkehrbringen zu verstehen, was heutzutage meist in Form physikalischer Datenträger und der Übertragung über das Internet passiert, wobei die rechtliche Einordnung des Internet-Downloads als Verbreitung aufgrund der fehlenden Körperlichkeit nach hM abgelehnt wird⁷⁵.

Auch *Spindler* gibt zu bedenken, dass die GPL gerade das für die Verbreitung über das Internet erforderliche Recht der Zugänglichmachung „nur bei extensiver Auslegung des Begriffes *distribute* in § 2 GPL“ ermöglicht⁷⁶.

Es bleibt also festzuhalten, dass das durch die GPL gewährte Recht auf Verbreitung nur für körperliche Vervielfältigungen gesichert ist und die Verbreitung über ein Datennetz (trotz alltäglicher Praxis, wie *Schiffner* zu bedenken gibt⁷⁷) mit rechtlicher Unsicherheit verbunden ist.

5.1.3. Bearbeitung von Open Source Software

Die Möglichkeit GPL-Software zu bearbeiten, um neue Funktionen zu einem Programm hinzuzufügen oder Fehler zu beheben und diese Bearbeitung wieder frei verbreiten zu dürfen, zählt zu den wichtigsten Eigenschaften von OSS unter GPL, daher bedarf auch dieser Aspekt einer urheberrechtlichen Betrachtung⁷⁸.

Die GPL gewährt dem Nutzer das Recht auf Bearbeitung nach § 2 GPL, wobei abgeleitete Werke (siehe 5.2.) wieder un-

74 Spindler, S. 44

75 Schiffner, S. 133

76 Spindler, S. 44

77 Schiffner, S. 134

78 Schiffner, S. 142

ter die GPL gestellt werden müssen, Änderungen mit einer deutlichen Kennzeichnung und dem Bearbeitungsdatum markiert werden müssen und ein Hinweis auf die GPL selbst erhalten bleiben muss.

Die Norm § 24 UrhG regelt die Erstellung eines neuen Werks, das sich an ein vorhandenes anlehnt, ohne aber eine Abhängigkeit von diesem Werk zu besitzen. In diesem Fall sollen sich die Rechte des Urhebers nicht auf das neue Werk erstrecken. Die Neuschaffung von Software in Anlehnung an bereits bestehende Software dürfte im OSS-Bereich selten auftreten, da die Bearbeitung vorhandener Werke erwünscht ist, um die Synergie-Effekte der gemeinschaftlichen Entwicklung nutzen zu können.

Interessanter ist daher § 23 UrhG, der sich mit der Bearbeitung und Umgestaltung von urheberrechtlich geschützten Werken befasst. So ist eine Bearbeitung durch Dritte grundsätzlich zulässig, allerdings wird für die Veröffentlichung der Bearbeitung das Einverständnis des Urhebers benötigt. Für Computerprogramme gilt die Sonderregelung in § 69c Nr. 2 UrhG⁷⁹, dass bereits die Bearbeitung und nicht erst die Veröffentlichung der Genehmigung des Urhebers bedarf, wobei *Schiffner* darauf hinweist, dass als Bearbeitung nur solche Handlungen gelten, die die konkrete Gestalt des Werks ändern⁸⁰. Eine Besonderheit ergibt sich aus dem 2. Satz des § 69c Nr. 2 UrhG, der die Rechte des Bearbeiters betrifft. So entsteht dem Bearbeiter unabhängig von den Rechten des Urhebers das so genannte Bearbeiterurheberrecht, welches jedoch ebenfalls von der Zustimmung des ursprünglichen Urhebers abhängig ist⁸¹.

Insgesamt kommen *Schiffner* und *Spindler* darin überein, dass die Bearbeitung von Open Source Software regelmäßig nach § 69c Nr. 2 UrhG geschieht und von der Zustimmung des

79 UrhG, § 69c Nr. 2: „Der Rechtsinhaber hat das ausschließliche Recht, folgende Handlungen vorzunehmen oder zu gestatten: [...] 2. die Übersetzung, die Bearbeitung, das Arrangement und andere Umarbeitungen eines Computerprogramms sowie die Vervielfältigung der erzielten Ergebnisse. [...]“

80 Schiffner, S. 144

81 Schiffner, S. 146 und Spindler, S. 24

Urhebers durch einen gültigen Lizenzvertrag (hier der GPL) ausgegangen werden kann⁸².

5.1.4. Neue Nutzungsarten und die GPL

Der § 9 GPL behält der FSF als Herausgeber der GPL das Recht vor, neue Versionen der GPL zu publizieren, die im Sinn der jetzigen Fassung der GPL entsprechen müssen, aber um neue Regelungen ergänzt sein können, um entstandene Probleme zu behandeln oder Detailfragen klarzustellen⁸³.

Problematisch ist, dass es laut GPL möglich ist Programme unter der aktuellen GPL und „any later version“⁸⁴ zu veröffentlichen, wobei der inhaltliche Umfang späterer Versionen nicht spezifiziert ist. Dies kollidiert mit § 31 IV UrhG, der die Einräumung von Nutzungsrechten an unbekanntem Nutzungsarten als unwirksam erklärt⁸⁵. Doch genau dieser Fall könnte eintreten, falls eine Neufassung der GPL neue Nutzungsrechte einführen würde.

Allgemein ist es daher dringend zu empfehlen, bei der Veröffentlichung einer Software unter GPL immer die genaue Version (aktuell: GPL Version 2 vom Juni 1991⁸⁶) anzugeben, um sich zumindest dieser Problematik zu entziehen⁸⁷.

Woraus *Spindler* folgert, dass durch die explizite Nennung der drei GPL-Nutzungsarten (Vervielfältigung, Verbreitung, Bearbeitung) in § 0 GPL, die Zweckübertragungslehre nach § 31 V UrhG nicht greift und daher auch eine Verbreitung von OSS über das Internet (als selbstständige Nutzungsart) nicht zulässig wäre⁸⁸, was aber gängige Praxis und daher sehr umstritten ist⁸⁹.

82 Schiffner, S. 144 und Spindler, S. 55 ff.

83 vergleiche GPL, § 9: „The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.“

84 siehe GPL, § 9 II

85 ausführlich: Spindler, S. 39 ff.

86 siehe: Kopf der GPL

87 siehe: Grassmuck, „GPL und deutsches Recht“

88 Spindler, S. 43

89 Schulz in Linux-Magazin, Ausgabe 9/2003, Seite 69

5.1.5. Urheberpersönlichkeitsrecht und GPL

Das *Copyright* des englisch-amerikanischen Rechtssystems kennt die urheberpersönlichkeitsrechtlichen Regelungen des UrhG nicht, daher bedarf dieser Aspekt ebenfalls einer Betrachtung.

Von besonderer Bedeutung sind dabei §§ 13, 14 UrhG, die die Urheberbezeichnung (zum Beispiel die Nennung des Namens/Künstlernamens) und den Schutz vor der Entstellung des eigenen Werks regeln.

Der § 13 UrhG findet sich in §§ 1, 2a GPL wieder, da bei jeder Vervielfältigung und Bearbeitung Hinweise auf die Urheber beibehalten bzw. hinzugefügt werden müssen⁹⁰, so dass hier kein Konflikt mit dem UrhG entsteht. Als möglichen Konflikt sieht *Schiffner* lediglich, dass die anonyme Urheberschaft nach § 39 I durch die Pflicht zur Kennzeichnung von Änderungen nach § 2a GPL nicht ausgeübt werden könnte⁹¹. Dem entgegen *Metzger/Jaeger* jedoch, dass die Anonymität durchaus gewahrt werden könnte, da nach § 2a GPL „nur ein auffälliger Vermerk⁹², aber keinen "©-Vermerk" i.S.d. U.S. Copyright Act, enthalten sein muss“⁹³.

Problematischer stellt sich die Bewertung von § 14 UrhG dar. So werfen *Metzger/Jaeger* die Frage auf, ob sich der Lizenznehmer einer GPL-Software der Gefahr aussetzt, „vom Urheber der vorbestehenden Software eine weitere Verwertung aus dem Gesichtspunkt des Integritätsschutzes [nach §§ 14, 97 UrhG] untersagt zu bekommen“⁹⁴, da der Integritätsschutz des Urhebers nicht abbedungen werden kann⁹⁵.

Jedoch werden in den meisten Fällen die Bedingungen nach §§ 14, 97 UrhG nicht vorliegen, da bei Werken der „kleinen Münze“ des Urheberrechts eine Gefährdung nur selten vorliegen dürfte. Auszuschließen ist allerdings nicht, dass „es trotz der weitreichenden Freistellung der Open Source Softwa-

90 siehe GPL, § 1: „[...] publish [...] each copy [with] an appropriate copyright notice [...]“ und GPL, § 2a: „You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.“

91 Schiffner, S. 190

92 Ein auffälliger Vermerk könnte zum Beispiel unter Verwendung eines Pseudonyms erfolgen.

93 Metzger/Jaeger, „Berücksichtigung des Rechts auf Namensnennung“

94 Metzger/Jaeger, „Integritätsschutz des Urhebers - abbedungen durch die GPL?“

95 vergleiche: Spindler, S. 47

re durch die GPL für den Bearbeiter bei dem Risiko bleibt, sich in Ausnahmefällen einem Verbot des Urhebers auf der Grundlage des § 14 UrhG gegenüber zu sehen. Betroffen hiervon ist insbesondere künstlerisch gestaltete Software.“⁹⁶

5.1.6. Urheber- und Miturheberschaft

Die GPL kennt nur den Begriff des Urhebers und differenziert nicht, wie das UrhG, zwischen Urheber und Miturheber. Ob man als Urheber oder Miturheber eines OSS-Projekts gilt, entscheidet (wie bereits in 3.3 besprochen) hauptsächlich die eigene Beteiligung an dem Entwicklungsprozess des Projekts.

Besteht eine Miturheberschaft, so greifen besondere Regeln. Zwischen den Miturhebern entsteht eine Gesamthandsgemeinschaft nach § 8 II 1 UrhG, die durch die Normen §§ 705 ff. BGB ergänzt wird⁹⁷. Daraus ergibt sich, dass auch die Verwertungsrechte gesamthänderisch gebunden sind, ebenso wie die gemeinsame Haftung aller Miturheber, falls Rechte Dritter verletzt wurden, da die Miturheber einen gemeinsamen Plan verfolgen⁹⁸.

In Grenzfällen der Softwareentwicklung, zum Beispiel bei der Erweiterung eines OSS-Projekts um ein selbstständiges Modul, tritt an die Stelle der Miturheberschaft die einfache Urheberschaft, da es sich um eine Erweiterung des ursprünglichen Programms handelt, die auch gesondert verwertbar sein kann, was gegen die Miturheberschaft spricht.

5.1.7. Verzicht auf die Verwertungsrechte oder einfaches Nutzungsrecht an jedermann?

Da die GPL laut § 4 nicht auf dem Verzicht der Verwertungsrechte durch den Urheber beruht⁹⁹, was im deutschen UrhR nach § 29 I UrhG auch nicht möglich wäre, sondern davon

96 Metzger/Jaeger, „Integritätsschutz des Urhebers - abbedungen durch die GPL?“

97 Als Besonderheit ist festzuhalten, dass diese Gemeinschaft entgegen der Regelungen der §§ 705 ff. BGB nicht kündbar ist, sondern bis zur Ablauf der Schutzfrist des UrhG (70 Jahre nach dem Tod des letzten Miturhebers) bestehen bleibt.

98 Spindler, S. 23

99 siehe GPL, § 4: „However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.“

spricht, dass ein Lizenzvertrag zwischen dem Urheber und dem Nutzer geschlossen wird, muss eine Einräumung eines einfachen Nutzungsrechts nach § 31 UrhG betrachtet werden.

Das einzuräumende Nutzungsrecht umfasst laut GPL § 0 das Recht zur Vervielfältigung, Verbreitung und Bearbeitung, wobei möglicherweise inkludierte weitere Rechte nach § 31 V UrhG und künftige unbekannte Verwertungsrechte explizit ausgeschlossen werden¹⁰⁰, was *Spindler* zu dem Schluss kommen lässt, dass die Vermietung von GPL-Software zu „einem unauflösbaren Konflikt“¹⁰¹ führt, da die Vermietung von der GPL nicht explizit als Nutzungsart genannt wird.

Erwähnung muss in diesem Zusammenhang auch die so genannte „Linux-Klausel“ des § 32 III UrhG finden, da sie es dem Urheber erlaubt, ein einfaches Nutzungsrecht an jedermann unentgeltlich einzuräumen¹⁰², was bei der Lizenzierung einer GPL-Software immer der Fall ist. Lediglich für die Erstellung der Kopie der Software ist die Möglichkeit einer Vergütung vorgesehen¹⁰³.

Bei der Schlussfolgerung der rechtlichen Konstruktion unterscheiden sich *Spindler* und *Schiffner* („inhaltlich beschränktes einfaches Nutzungsrecht“¹⁰⁴) von *Metzger/Jaeger* („einfaches Nutzungsrechts unter der auflösenden Bedingung des § 158 BGB“¹⁰⁵), weil sich die Möglichkeit der inhaltlichen Beschränkung durch die Eigenständigkeit der Nutzungsart „Open Source Software“ anbietet.

Insgesamt hält die Konstruktion der GPL der Nutzungsrechtseinräumung durch den jeweiligen Urheber der urheberrechtlichen Prüfung stand.

5.2. Aus der GPL abgeleitete Werke

Die GPL definiert in § 2 II genau, was ein abgeleitetes Werk („derived work“) ist, da das Copyleft-Gebot der GPL nur für abgeleitete Werke gilt. Demnach handelt es sich bei einem

100 siehe GPL, § 0: „Activities other than copying, distribution and modification are not covered by this License“

101 Spindler, S. 45

102 vergleiche Schiffner, S. 172

103 siehe GPL, § 1: „You may charge a fee for the physical act of transferring a copy, [...].“

104 Spindler, S. 29 und Schiffner, S.170

105 Metzger/Jaeger, „2. Verknüpfung von Nutzungsrecht und Verpflichtungen aus der GPL - rechtliche Konstruktion“

Werk um ein abgeleitetes Werk, wenn es identifizierbare Teile eines unter GPL stehenden Werks enthält¹⁰⁶. Als eigenständige Werke gelten hingegen solche, die unabhängig und eigenständig sind und keine Teile von GPL-geschützten Werken enthalten. Solche Werke sollen demnach nicht in den Wirkungsbereich der GPL fallen.

Der Status von abgeleiteten Werken entspricht daher den Normen §§ 23, 69c Nr. 2 UrhG, die die Bearbeitung eines Werks regeln, für die das Einverständnis des Urhebers notwendig ist¹⁰⁷.

Die zweite Regelung in § 2 II GPL betrifft Werke, die zwar nicht abgeleitet sind, aber fest an GPL-Werke gekoppelt sind und gemeinsam als ein Werk verbreitet werden. Für eine solche Verbindung mehrerer Werke soll nach § 2 II GPL für das Ganze umfassend die GPL gelten¹⁰⁸, was dem deutschen Urheberrecht in dieser Form fremd ist. Im UrhG regelt § 9 die Verbindung zweier Werke, wobei allerdings jedes für sich als eigenständiges Werk bestehen bleibt, was die Regelung der GPL aber nicht aussetzt.

Daher stellt *Spindler* einige Indizien auf, wann ein verbundenes Werk laut GPL als Ganzes zu werten sei. Dies sei dann der Fall, wenn das neue Werk nicht ohne GPL-Code ausgeführt werden könne und der Code des neuen Teils gemeinsam mit GPL-Code in einem Binärprogramm verbunden sei. Die Abhängigkeit von einem Betriebssystem wird als Kriterium nicht herangezogen, „da sonst sämtliche an sich eigenständigen Programme stets ein Derivat vom Betriebssystem wären“¹⁰⁹.

5.3. Möglichkeiten der Doppellizenzierung

In manchen Fällen kann es sinnvoll erscheinen, eine Software sowohl unter GPL als auch unter einer proprietären Lizenz

106 siehe GPL, § 2: „If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works.“

107 Spindler, S. 55

108 siehe GPL, § 2: „But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License [...]“

109 Spindler, S. 56

zu veröffentlichen. Dies ist natürlich nur dann möglich, wenn die Software nicht bereits zuvor unter GPL gestellt wurde.

Mit dieser Vorgehensweise kann ein Anbieter durch den Einblick in die Quelltexte seiner Software das Vertrauen in sein Produkt stärken, ohne das traditionelle Modell der kostenpflichtigen Softwarelizenzierung aufgeben zu müssen. Allerdings muss auch beachtet werden, dass mögliche Weiterentwicklungen im Open Source-Zweig der Software nicht in den proprietären Zweig übernommen werden dürfen¹¹⁰, sondern ggf. neu entwickelt werden müssen.

Häufig wird bei der Doppellizenzierung jedoch nicht die GPL als Lizenz gewählt, sondern eine Lizenz, die nur den nicht kommerziellen Einsatz der Software erlaubt, um so den Vertrieb der Software unter der proprietären Lizenz nicht zu behindern¹¹¹.

5.4. Entwicklung von OSS im Arbeitsverhältnis

Da Open Source Software heute nicht mehr ausschließlich von privaten Programmierern in ihrer Freizeit erstellt wird, sondern zu einem erheblichen Teil von Programmierern aus großen Firmen¹¹², stellt sich die Frage, wer die urheberrechtlichen Nutzungsrechte der Software erhält. Grundsätzlich sind zwei Situationen zu unterscheiden: Auf der einen Seite der Angestellte einer Firma und auf der anderen Seite ein selbstständiger Programmierer, der für einen Auftraggeber Software entwickelt.

Der selbstständige Entwickler lizenziert seine Software an den Käufer (überträgt ein einfaches Nutzungsrecht im vertraglich festgelegten Umfang) und bleibt selbst im Besitz der ausschließlichen Nutzungsrechte, so dass er seine Software auch unter einer OSS-Lizenz freigeben könnte.

Bei angestellten Entwicklern regelt § 69b UrhG die Ausübung der vermögensrechtlichen Befugnisse. So gehen sämtliche Befugnisse an den Arbeitgeber, falls die Software in Wahrnehmung der Aufgaben des Arbeitnehmers oder nach den

110 Spindler, S. 17

111 Schiffner, S. 99

112 IBM zum Beispiel beschäftigt eine große Anzahl OSS-Programmierer.
Quelle: <http://www.heise.de/newsticker/meldung/39629>

Anweisungen seines Arbeitgebers geschaffen wurden. Bei dem Angestellten verbleiben lediglich die „persönlichkeitsrechtlichen Bestandteile des Urheberrechts, etwa das Recht der Nennung seines Namens“¹¹³ nach § 13 UrhG als Autor der Software.

Problematisch ist jedoch die Einordnung von Open Source Software, die zwar in der Arbeitszeit des Entwicklers entstanden ist, aber nicht auf Anweisung des Arbeitgebers. Hier hat das OLG München im Jahr 2000 entschieden¹¹⁴, dass die Verwertungsrechte beim Autor verbleiben und § 69b nicht greift. Dies kann aber in Grenzfällen zu Problemen führen, falls die Software vom Entwickler unter GPL veröffentlicht wird und sich später herausstellt, dass der Arbeitgeber doch die ausschließlichen Nutzungsrecht besitzt. Denn dann könnte der Arbeitgeber „jedermann die Nutzung dieser Programmteile [...] untersagen“¹¹⁵. In vielen Fällen enthält der Arbeitsvertrag aber eine Klausel, die alle Vermögensrechte der in der Arbeitszeit entwickelten Software an den Arbeitgeber überträgt. Damit dürften die geschilderten Probleme in der Praxis selten auftreten.

Jaeger wirft die Frage der angemessenen Vergütung nach § 32a auf, da der Bundesgerichtshof die Anwendbarkeit auch für angestellte Programmierer bejaht hat¹¹⁶. Sollte der Arbeitslohn in einem auffälligen Missverhältnis zu den Erträgen und Vorteilen stehen, die sich ein Dritter mit der Software verschafft hat, entsteht dem Programmierer ein Anspruch auf Änderung des Lizenzvertrages. Ob dies für OSS zutreffend ist, muss noch geklärt werden. Sollte es bestätigt werden, wäre die Situation prekär: Denn auch die Möglichkeit, auf weitere Vergütung zu verzichten (durch ein unentgeltliches Nutzungsrecht an jedermann nach § 32 Abs. 3 Satz 3) würde nicht greifen, da dieses nur durch den Lizenzgeber (Arbeitgeber) einsetzbar ist, was nicht vor Forderungen nach angemessener Vergütung des Urhebers (Arbeitnehmer) schützen würde. Als Lösung empfiehlt *Jaeger* „durch eine vertragliche Regelung zwischen

113 Jaeger in Linux-Magazin, Ausgabe 2/2004, Seite 78

114 OLG München a. a. O., CR 2000, 429.

115 Böken in iX, Ausgabe 5/2004, Seite 92

116 Das Urteil im Volltext: <http://www.jurpc.de/rechtspr/20010254.htm>

Arbeitnehmer und Arbeitgeber dafür zu sorgen, dass die Software durch den Arbeitnehmer [...] unter einer Open-Source-Lizenz freigegeben wird. Dies kann durch eine Rücklizenzierung geschehen [...]“¹¹⁷, wobei diese Konstruktion in der Praxis ebenfalls selten zu finden sein dürfte.

Es ist also dringend zu empfehlen, bei der Ausgestaltung von Arbeitsverträgen für Softwareentwickler umsichtig vorzugehen und Besonderheiten (zum Beispiel die Mitarbeit bei einem OSS-Projekt in der Freizeit) einzubeziehen, damit es nicht zu Problemen zwischen Arbeitgeber und Arbeitnehmer kommen kann.

6. Fazit

Insgesamt lässt sich feststellen, dass sich die urheberrechtlichen Aspekte der GPL zum Großteil gut in das deutsche Rechtssystem einfügen. Das dies nicht immer problemlos möglich ist und sich die Auslegungen teils erheblich voneinander unterscheiden, darf aufgrund der US-amerikanischen Herkunft der GPL nicht verwundern.

Ob die Zukunftsidee einer landesspezifischen sprachlich und rechtlich angepassten Variante der GPL, eine Lösung für die verbleibenden Probleme darstellen könnte, bleibt fraglich, da die Entwicklung von Open Source Software weltweit vorangetrieben wird und eine landesspezifische Anpassung der GPL die Gefahr einer Zersplitterung in sich birgt.

Da die Open Source Bewegung und die damit verbundenen rechtlichen Fragestellungen noch sehr jung sind, wird die künftige Entwicklung spannend bleiben, wobei die Beständigkeit der GPL vor Gerichten von besonderem Interesse ist.

Die Idee der freien Software nach dem Prinzip der GPL hat allerdings - abseits aller rechtlichen Fragen - bereits heute einen beachtlichen Erfolg errungen und wird die Entwicklung der IT-Industrie in den kommenden Jahren entscheidend beeinflussen.

Oder wie es *Grassmuck* ausdrückt: „In der Gesamtschau ist es keine Übertreibung, die GPL als den größten Hack der Wissensordnung zu bezeichnen, seit britische Verlegergilden das Urheberrecht erfanden.“

7. Literaturverzeichnis und Erklärung

- Schiffner, Thomas
„Open Source Software: Freie Software im deutschen Urheber- und Vertragsrecht“, ISBN 3894814659
- Spindler, Gerald
„Rechtsfragen der Open Source Software“
(Studie im Auftrag des Verbands der Softwareindustrie Deutschlands e.V.)
http://www.vsi.de/inhalte/aktuell/studie_final_safe.pdf
- Fechner, Frank
„Medienrecht“, 4. Auflage, ISBN 3828221547
- Metzger, Axel und Jaeger, Till
„Open Source Software und deutsches Urheberrecht“
http://www.ifross.de/ifross_html/art1.html
- „Gesetz über Urheberrecht und verwandte Schutzrechte“ von 1965
<http://bundesrecht.juris.de/bundesrecht/urhg/gesamt.pdf>
- GNU General Public License
<http://www.fsf.org/licenses/gpl.html>
- Deutsche Übersetzung der GNU General Public License
<http://www.gnu.de/gpl-ger.html>
- Jaeger, Till
„Tux im Dienst“, Linux-Magazin, Linux New Media AG München, Ausgabe 2/2004, Seite 78-80 (zit. Seiten 78 und 80)
http://www.ifross.de/ifross_html/art43.pdf
- Schulz, Carsten
„Die scharfe Klinge des Gesetzes?“, Linux-Magazin, Linux New Media AG München, Ausgabe 9/2003, Seite 68-70 (zit. Seite 69)
http://www.ifross.de/ifross_html/art35.pdf
- Böken, Arnd
„Trennschärfe“, iX, Heise Zeitschriften Verlag GmbH & Co. KG Hannover, Ausgabe 5/2004, Seite 91-92 (zit. Seite 92)
- Grassmuck, Volker
„Lizenzmodelle“
<http://www.mikro.org/Events/OS/text/lizenzen.html>
- Wikipedia - die freien Enzyklopädie
<http://de.wikipedia.org/>

- Bundesministeriums für Wirtschaft und Technologie
„Open-Source-Software - Ein Leitfaden für kleine und mittlere Unternehmen“
http://www.bmwi.de/Redaktion/Inhalte/Downloads/Homepage_2Fdownload_2Finfogesellschaft_2FOpen-Source-Software.pdf,property=pdf.pdf
- Hang, Jiayin und Dr. Hohensohn
„Eine Einführung zum Open Source Konzept aus Sicht der wirtschaftlichen und rechtlichen Aspekte“
Eine Studie im Rahmen des Projektes NOW: Nutzung des Open Source Konzepts in Wirtschaft und Industrie
http://now.c-lab.de/veroeffentlichungen/einfuehrung/Anforderungskatalog_Teil_final.pdf
- Metzger, Axel
„Freie Software – Rechtlich sicher?“
<http://www.heise.de/tp/deutsch/inhalt/te/7947/1.html>

Hinsichtlich der verwendeten Abkürzungen wird auf Kirchner, Butz, „Abkürzungsverzeichnis der Rechtssprache“, 5. Auflage, Berlin 2003 verwiesen.

Alle in dieser Arbeit ausgewiesenen URLs wurden am 13. Mai 2004 abschließend auf ihre Gültigkeit überprüft. Allerdings kann für die künftige Erreichbarkeit der verlinkten Inhalte keinerlei Garantie übernommen werden. Falls eine URL nicht mehr erreichbar sein sollte, können Kopien der Inhalte bei dem Autor dieser Arbeit angefordert werden.

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig und nur unter der Nutzung der angegebenen Literatur angefertigt habe. Wörtlich übernommene Sätze oder Satzteile sind als Zitate belegt, andere Anlehnungen sind unter Angabe der Quellen in den Fußnoten kenntlich gemacht.

Jens Nödler, 14. Mai 2004