

GUTACHTEN ZU AUSGEWÄHLTEN RECHTLICHEN ASPEKTEN  
DER OPEN SOURCE SOFTWARE

JBB Rechtsanwälte  
Dr. Till Jaeger, München  
Dr. Carsten Schulz, München

Februar 2005

## **Inhaltsübersicht**

<b>Teil 1.</b>	<b>Einführung.....</b>	<b>1</b>
<b>Teil 2.</b>	<b>Immaterialgüterrechtlicher Schutz von Freier Software .....</b>	<b>4</b>
<b>Teil 3.</b>	<b>Lizenzrechtliche Aspekte der Freien Software .....</b>	<b>18</b>
<b>Teil 4.</b>	<b>Vertragliche Beziehungen bei der Nutzung von Freier Software.....</b>	<b>52</b>
<b>Teil 5.</b>	<b>Haftung und Gewährleistung .....</b>	<b>63</b>
<b>Teil 6.</b>	<b>Rechtliche Fragen bei der Gestaltung von Open Source Projekten.....</b>	<b>80</b>
<b>Teil 7.</b>	<b>Internationale Aspekte .....</b>	<b>89</b>

## Inhaltsverzeichnis

<b>Teil 1. Einführung</b> .....	<b>1</b>
I. Begriff der Freien Software/Open Source Software.....	1
II. Herkunft und Entwicklung des Lizenzmodells.....	3
<b>Teil 2. Immaterialgüterrechtlicher Schutz von Freier Software</b> .....	<b>4</b>
I. Abgrenzung patentrechtlicher und urheberrechtlicher Schutz .....	4
II. Urheberrecht .....	5
1. Grundlagen softwareurheberrechtlichen Schutzes.....	5
2. Lizenzierung als vertragliche Gestattung der Softwarenutzung.....	6
a) Die durch Open Source Lizenzen gewährten Rechte .....	7
b) Urheberschaft und Rechtsinhaberschaft bei Open Source Software: Der richtige Lizenzgeber .....	8
c) Rechtserwerb bei Fehlern in der „Lizenzkette“ .....	10
III. Patentrecht .....	11
1. Grundzüge patentrechtlichen Schutzes.....	11
2. Grundtypen von patentierbaren Computerprogramm .....	12
a) Programme zur Steuerung klassisch-technischer, mechanischer Einheiten	12
b) Neue, erfinderische Betriebsweise einer EDV-Anlage.....	12
c) Anwendungsprogramme.....	13
3. Mögliche Problemfelder.....	14
a) „Proprietäre“ Patentanmeldungen auf Freie Software.....	14
b) Das Risiko von Patentverletzungsprozessen.....	15
c) Patente im System der Open Source Lizenzierung.....	15
IV. Praktische Hinweise: Was ist bei Programmierung, Vertrieb und Verwertung von Open Source Software zu berücksichtigen, um Patent- und Urheberrechtsverletzungen zu vermeiden? .....	17

<b>Teil 3. Lizenzrechtliche Aspekte der Freien Software .....</b>	<b>18</b>
I. Einführung.....	18
II. Allgemeine Hinweise.....	19
1. Rechte aus der Open Source Lizenz.....	19
2. Wann muss die Open Source Lizenz abgeschlossen werden? .....	20
3. Welche Rechte werden gewährt? .....	21
a) Zugänglichmachung im Internet? .....	21
b) Ist Vermietung gestattet? .....	22
c) Ist Application Service Providing gestattet? .....	22
III. Rechte und Pflichten wichtiger Open Source Lizenzen .....	23
1. BSDartige Lizenzen.....	23
a) Rechte.....	24
b) Pflichten .....	24
(1) Pflichten beim Vertrieb unveränderter Versionen.....	24
(2) Pflichten bei der Bearbeitung .....	25
(3) Pflichten beim Vertrieb bearbeiteter Versionen.....	26
(4) Besonderheiten bei den BSD-Lizenzen.....	26
(5) Besondere Pflichten bei den Apache-Lizenzen.....	27
c) Kombination mit Softwarebestandteilen unter anderen Lizenzen.....	28
2. Die GNU General Public License (GPL).....	29
a) Rechte.....	29
b) Pflichten .....	30
(1) Pflichten beim Vertrieb unveränderter Versionen.....	30
(2) (Zusätzliche) Pflichten beim Vertrieb veränderter Versionen .....	31
c) Kombination mit Softwarebestandteilen unter anderen Lizenzen.....	32
(1) Kernelmodule.....	34
(2) Plugins.....	35
(3) Bibliotheken .....	35
(4) Lizenzkompatibilität: Kombination von GPL-Software mit Software unter anderen Lizenzen.....	36
3. Die GNU Lesser General Public License (LGPL).....	38
a) Rechte.....	38

b)	Pflichten.....	38
c)	Kombination mit Softwarebestandteilen unter anderen Lizenzen.....	39
(1)	Pflichten aus dem beschränkten Copyleft der Ziffer 6 LGPL .....	39
(2)	Pflichten aus dem beschränkten Copyleft der Ziffer 7 LGPL .....	40
4.	Die Common Public License (CPL).....	41
a)	Rechte.....	41
b)	Pflichten.....	42
(1)	Pflichten beim nicht-kommerziellen Vertrieb.....	42
(2)	Pflichten beim kommerziellen Vertrieb.....	43
c)	Kombination mit Softwarebestandteilen unter anderen Lizenzen.....	44
(1)	Eigenständige Softwaremodule.....	44
(2)	Kein abgeleitetes Werk .....	44
5.	Die Mozilla Public License (MPL).....	45
a)	Rechte.....	46
b)	Pflichten.....	46
c)	Kombination mit Softwarebestandteilen unter anderen Lizenzen.....	48
IV.	Nachträgliche Änderungen einer Open Source Lizenz.....	49
V.	Praktische Hinweise: Was ist bei der Auswahl einer Open Source Lizenz zu berücksichtigen?.....	51
<b>Teil 4.</b>	<b>Vertragliche Beziehungen bei der Nutzung von Freier Software.....</b>	<b>52</b>
I.	Überblick über die vertraglichen Beziehungen bei der Nutzung Freier Software.....	52
II.	Verträge über die Überlassung der Software zur Benutzung.....	53
1.	Dauerhafte Überlassung gegen einmalige Bezahlung.....	54
2.	Dauerhafte unentgeltliche Überlassung.....	55
III.	Verträge über die Einräumung von Entwicklungs- und Vertriebsrechten.....	55
1.	Non-Copyleft-Lizenzen.....	56
2.	Copyleft-Lizenzen.....	57
IV.	Erstellungs-, Anpassungs- und Wartungsverträge.....	58
1.	Neuprogrammierung einer Open Source Software .....	58

2.	Anpassung an kundenspezifische Bedürfnisse.....	59
3.	Service- und Supportverträge.....	60
V.	Grenzen der Aufspaltbarkeit in entgeltliche und unentgeltliche Vertragsteile ....	61
VI.	Praktische Hinweise: Was ist bei der vertraglichen Gestaltung von Programmierung, Vertrieb und Verwendung von Open Source Software zu berücksichtigen?.....	62
<b>Teil 5.</b>	<b>Haftung und Gewährleistung .....</b>	<b>63</b>
I.	Allgemeines / Unterschiede zu proprietär verwerteten Programmen.....	63
1.	Begrifflichkeiten: Haftung und Gewährleistung.....	63
2.	Gewährleistungs- und Haftungsmaßstäbe.....	64
3.	„Beschaffenheit“ und Sachmängelgewährleistung.....	65
4.	Unterschiede zu proprietär verwerteten Programmen .....	66
II.	Gewährleistung und Haftung in den jeweiligen Vertragsverhältnissen .....	67
1.	Gewährleistung und Haftung innerhalb der Überlassungsverträge.....	67
a)	Anwendung von Kaufrecht.....	67
b)	Anwendung von Schenkungsrecht.....	69
2.	Gewährleistung und Haftung innerhalb von Open Source Lizenzverträgen....	70
a)	Unwirksamkeit bestehender Haftungs- und Gewährleistungsausschlüsse..	70
b)	Gesetzlicher Haftungs- und Gewährleistungsmaßstab bei Non-Copyleft-Lizenzen.....	71
c)	Gesetzlicher Haftungs- und Gewährleistungsmaßstab bei Copyleft-Lizenzen	72
3.	Gewährleistung und Haftung bei Erstellung und Änderung von Freier Software.....	73
a)	Erstellung von Software im Kundenauftrag.....	73
b)	Anpassung von Open Source Software.....	74
III.	Außervertragliche Haftung.....	74
1.	Produkthaftung.....	74
a)	Wer haftet? .....	74
b)	Wofür wird gehaftet?.....	75

c)	Umfang der Haftung .....	76
2.	Verschuldensabhängige außervertragliche Haftung.....	76
a)	Haftung des Programmierers der Software .....	77
b)	Haftung des Distributors der Software.....	77
c)	Haftung von Core-Teams und Maintainern .....	78
IV.	Praktische Hinweise: Was ist bei Programmierung, Vertrieb und Verwendung von Open Source Software zu berücksichtigen, um unnötige Gewährleistungs- und Haftungsrisiken auszuschließen?.....	79
<b>Teil 6.    Rechtliche Fragen bei der Gestaltung von Open Source Projekten.....</b>		<b>80</b>
I.	Entwicklerzusammenschlüsse.....	80
1.	Zusammenschlüsse zur Schaffung und Verwertung Freier Software.....	81
a)	Miturheber .....	81
b)	Urheber verbundener Werke.....	82
c)	Angestellte Urheber .....	83
(1)	Bedeutung für neu geschaffene Software.....	83
(2)	Bedeutung für die Weiterentwicklung von Open Source Software.....	83
2.	Zusammenschlüsse zur Koordination von Open Source Projekten.....	84
II.	Von Unternehmen initiierte Open Source Projekte .....	86
1.	Rechte.....	86
2.	Auswahl der Lizenz.....	87
3.	Auswahl der Koordinationseinheiten.....	87
III.	Nachverfolgbarkeit von Entwicklungsbeiträgen.....	87
IV.	Praktische Hinweise: Merkposten bei der rechtlichen Gestaltung von freien Projekten.....	88
<b>Teil 7.    Internationale Aspekte .....</b>		<b>89</b>
I.	Bedeutung der Frage nach dem anwendbaren Recht .....	89
II.	Wann ist deutsches Recht anwendbar? .....	90

1.	Unterscheidung zwischen Urheber- und Vertragsstatut .....	90
a)	Die Bestimmung des anwendbaren Rechts über das Vertragsstatut .....	90
(1)	Lizenzverträge.....	91
(2)	Andere Vertragstypen .....	93
b)	Die Bestimmung des anwendbaren Rechts über das Urheberstatut .....	94
2.	Fazit.....	95
III.	Praktische Hinweise: Sollten Rechtswahlklauseln verwendet werden?.....	96

## Literaturverzeichnis

- Bamberger, Heinz Georg/ Roth, Herbert (Hrsg.)*, Kommentar zum Bürgerlichen Gesetzbuch,  
Band 1, §§ 1-610, München 2003.  
Band 2, §§ 611-1296, München 2003.
- Bettinger, Torsten/ Schefelt, Michael*, Application Service Providing: Vertragsgestaltung und Konfliktmanagement, in: CR 2001, Seite 729 ff.
- Busse, Rudolf (Begr.)*, Patentgesetz, Kommentar, 6. Auflage, Berlin 2003.
- Deike, Thies*, Open Source Software: IPR-Fragen und Einordnung ins deutsche Rechtssystem, in: CR 2003, Seite 9 ff.
- Dittrich, Robert (Hrsg.)*, Beiträge zum Urheberrecht V, Wien 1997.
- Dreier, Thomas/ Schulze, Gernot*, Urheberrechtsgesetz, Kommentar, München 2004.
- Gernhuber, Joachim*, Handbuch des Schuldrechts, Band 8, Das Schuldverhältnis, Tübingen 1989.
- Grassmuck, Volker*, Freie Software zwischen Privat- und Gemeineigentum, Bonn 2002.
- Jaeger, Till/ Metzger, Axel*, Open Source Software. Rechtliche Rahmenbedingungen der Freien Software, München 2002.
- Koch, Frank A.*, Urheber- und kartellrechtliche Aspekte der Nutzung von Open Source Software, Teile I und II, in: CR 2000, Seite 273 ff. (Teil I) und CR 2000, Seite 333 ff. (Teil II).
- Lehmann, Michael (Hrsg.)*, Rechtsschutz und Verwertung von Computerprogrammen, 2. Auflage, Köln 1993.
- Marly, Jochen*, Softwareüberlassungsverträge, 4. Auflage, München 2004.
- Medicus, Dieter*, Bürgerliches Recht, Eine nach Anspruchsgrundlagen geordnete Darstellung zur Examensvorbereitung, 20. Auflage, Köln 2004.
- Mes, Peter*, Patentgesetz – Gebrauchsmustergesetz, Kommentar, München 1997.

- Metzger, Axel*, Rechtsgeschäfte über das Droit moral im deutschen und französischen Urheberrecht, München 2002.
- Metzger, Axel*, Softwarepatente im künftigen europäischen Patentrecht, in: CR 2003, Seite 313 ff.
- Metzger, Axel/ Jaeger, Till*, Open Source Software und deutsches Urheberrecht, in: GRUR Int. 1999, Seite 839 ff.
- Möhring, Philipp/ Nicolini, Käte*, Urheberrechtsgesetz. Kommentar, 2. Auflage, München 2000.
- Palandt*, Bürgerliches Gesetzbuch, 63. Auflage, München 2004.
- Pres, Andreas*, Gestaltungsformen urheberrechtlicher Softwarelizenzverträge, Köln 1994.
- Schertz, Christian/ Omsels, Hermann-Josef (Hrsg.)*, Festschrift für Paul W. Hertin zum 60. Geburtstag am 15. November 2000, München 2000.
- Schiffner, Thomas*, Open Source Software. Freie Software im deutschen Urheber- und Vertragsrecht, München 2002.
- Schricker, Gerhard (Hrsg.)*, Urheberrecht. Kommentar, 2. Auflage, München 1999.
- Schulz, Carsten*, Dezentrale Softwareentwicklungs- und Softwarevermarktungskonzepte; Vertragsstrukturen in Open Source Modellen, erscheint Ende März 2005.
- Spindler, Gerald (Hrsg.)*, Rechtsfragen bei Open Source, Köln 2004.
- Spindler, Gerald*, Rechtsfragen der Open Source Software, Gutachten im Auftrag des Verbandes der Softwareindustrie Deutschland e.V., 2003, Download unter: <[http://www.vsi.de/inhalte/aktuell/studie\\_final\\_safe.pdf](http://www.vsi.de/inhalte/aktuell/studie_final_safe.pdf)>.
- Spindler, Gerald/ Wiebe, Andreas*, Open Source-Vertrieb. Rechteeinräumung und Nutzungsberechtigung, in: CR 2003, Seite 873 ff.
- Thewalt, Stefan*, Softwareerstellung als Kaufvertrag mit werkvertraglichem Einschlag. § 651 BGB nach der Schuldrechtsreform, in: CR 2002, Seite 1 ff.

*Ulmer, Eugen*, Die Immaterialgüterrechte im internationalen Privatrecht, München 1975.

*Ulmer, Eugen*, Urheber- und Verlagsrecht, 3. Auflage, Berlin u.a. 1980.

*Wandtke, Artur-Axel/ Bullinger, Winfried*, Praxiskommentar zum Urheberrecht, München 2002.

## **Teil 1. Einführung**

### ***I. Allgemeines***

Das vorliegende Gutachten behandelt Rechtsfragen der Freien bzw. Open Source Software. Ziel der Studie ist es, einen vertieften Einblick in die zentralen rechtlichen Rahmenbedingungen der Open Source Entwicklung und des Open Source Vertriebs zu vermitteln und auf mögliche Problembereiche hinzuweisen. Zielgruppe sind insbesondere die an der Entwicklung und am Vertrieb beteiligten Personen und Unternehmen, denen Leitlinien für eine praktische Umsetzung an die Hand gegeben werden sollen.

Zahlreiche Fragestellungen im Zusammenhang mit Open Source Software sind nach wie vor unter Juristen stark umstritten. Diese Vielzahl der unterschiedlichen Meinungen konnte im Rahmen der vorliegenden Studie schon aus Gründen des Umfangs und der Übersichtlichkeit sowie im Hinblick auf die Zielgruppe nicht vollständig und erschöpfend dargestellt werden. Es wurde vielmehr versucht, durchgängig die jeweils führenden Ansichten zu den einzelnen Fragestellungen darzustellen, um so die für die Praxis erforderliche schnelle Orientierung sicherzustellen.

Die für ein Gutachten erforderliche Verallgemeinerung bringt es mit sich, dass mit der vorliegenden Studie Rechtsberatung im Einzelfall nicht geleistet werden kann. Wo es um die Lösung von Detailproblemen und fallspezifischen Fragestellungen geht, ersetzt die nachfolgende Darstellung eine anwaltliche Beratung nicht.

### ***II. Begriff der Freien Software/Open Source Software***

Was unter die Begriffe „Freie Software“ und „Open Source Software“ fällt, ist anhand von allgemein anerkannten Definitionen gut bestimmbar. Die Definition von Freier Software (free software) wurde von der Free Software Foundation (FSF) erarbeitet, die auch einige wesentliche Lizenzen erstellt hat (insbesondere die GNU General Public License, GPL). Die „Free Software Definition“<sup>1</sup> beschreibt die Anforderungen an das, was als Freie

---

<sup>1</sup> <<http://www.gnu.org/philosophy/free-sw.html>>.

Software bezeichnet werden kann. Diese Anforderungen sind nicht technischer Art und beschreiben auch kein Entwicklungsmodell, sondern rechtliche Vorgaben, wie sie sich in der Lizenz ausdrücken. Was Freie Software ist, ergibt sich demnach aus der Softwarelizenz.

Wichtigstes Kriterium für Freie Software ist die Einräumung von umfassenden (einfachen) Nutzungsrechten an jedermann. Ohne Einschränkung des Nutzerkreises muss jede natürliche oder juristische Person das Recht erhalten, die Software in unveränderter oder veränderter Form zu vervielfältigen, zu verbreiten und öffentlich zugänglich zu machen. Voraussetzung dafür, dass diese Rechte auch praktisch genutzt werden können, ist der Zugang zum Quellcode und der Ausschluss von Lizenzgebühren, da ansonsten eine freie Weitergabe bzw. Weiterentwicklungen nicht möglich wären. Das Verbot von Lizenzgebühren schließt allerdings nicht die kommerzielle Nutzung aus. Wie die Free Software Definition ausdrücklich betont, darf das Verwendungsgebiet der Software nicht begrenzt werden (z.B. auf nicht-militärische Nutzung), auch nicht auf eine nicht-kommerzielle Nutzung. Demnach darf für alle Leistungen ein Entgelt verlangt werden, lediglich für die Einräumung von Nutzungsrechten ist dies nicht gestattet. Praktisch bedeutet dies, dass dem Lizenznehmer nicht verboten werden kann, die Software kostenlos an Dritte weiterzugeben.

Freie Software Lizenzen dürfen dem Lizenznehmer durchaus Pflichten auferlegen, etwa Hinweispflichten oder die Pflicht, Weiterentwicklungen nur unter bestimmten freien Lizenzbedingungen weiterzugeben (sog. „Copyleft“). Solche Pflichten dürfen aber nicht dazu führen, die oben genannten Rechte praktisch zu entwerten.

Die Open Source Definition (OSD)<sup>2</sup> enthält eine im Wesentlichen die gleiche Definition wie die „Free Software Definition“ und unterscheidet sich nur in der Wortwahl und Nuancen. Daher können die Begriff „Freie Software“ und „Open Source Software“ synonym verwendet werden. Die OSD wurde von Bruce Perens und der Open Source Initiative (OSI) erstellt. Die OSI „zertifiziert“ Lizenzen als Open Source Lizenzen.<sup>3</sup> Damit wird für

---

<sup>2</sup> <[http://www.opensource.org/docs/definition\\_plain.html](http://www.opensource.org/docs/definition_plain.html)>.

<sup>3</sup> <[http://opensource.org/docs/certification\\_mark.php](http://opensource.org/docs/certification_mark.php)>.

Nutzer eine einfache Möglichkeit geschaffen, zu ermitteln, ob eine bestimmte Software tatsächlich als Open Source Software anzusehen ist.

Allerdings wird in der Praxis der Begriff „Open Source Software“ zunehmend auch in einem weiteren Sinn verstanden, nämlich als Software, deren Quellcode einsehbar ist. Der Begriff „Freie Software“ wird gelegentlich mit „Freeware“ bzw. „kostenloser Software“ verwechselt. Zur Vermeidung von Begriffsverwirrungen, sollten „Freie Software“ und „Open Source Software“ nur im Sinne der genannten Definitionen verwendet werden. Denn diese Definitionen beschreiben ein eigenes Lizenzmodell, das letztlich auch eine eigenständige rechtliche Bewertung verdient.

### **III. Herkunft und Entwicklung des Lizenzmodells**

Die Entwicklung des Lizenzmodells für Freie Software hat seine Ursprünge im Beginn der 80er Jahre. Nachdem in den 70er Jahren Software als eigenständiges Wirtschaftsgut „entdeckt“ wurde, gaben die entstehenden Softwareunternehmen die bis dahin von den Hardwareanbietern gepflegte Praxis auf, ihren Kunden den Quellcode zugänglich zu machen.<sup>4</sup>

Für den am MIT (Massachusetts Institute of Technology) tätigen Programmierer Richard Stallman waren die sich zunehmend entwickelnden Restriktionen bei der Softwareentwicklung und die persönliche Erfahrung, dass Programme der am MIT beschäftigten Entwickler frei ausgetauscht wurden, der Anlass, ein freies Betriebssystem zu initiieren. Technisch wurde die Entwicklung seit 1984 vom „GNU“-Projekt getragen.<sup>5</sup> „GNU“ steht für „GNU's not Unix“ und macht die Intention deutlich, ein Unix-artiges Betriebssystem zu entwickeln, das jedoch nicht der proprietären Lizenz des damals von AT&T vertriebenen Ur-Unix unterliegt.

Die inhaltliche Entwicklung der Idee dessen, was Freie Software ausmachen soll, wurde von der 1985 gegründeten Free Software Foundation (FSF), einer gemeinnützigen Stiftung, übernommen. Parallel dazu wurde auf juristischer Ebene eine Lizenz entwickelt, die die Freiheitsrechte der Nutzer sichern sollte. Schließlich wurde 1989 die erste Versi-

---

<sup>4</sup> Zur Geschichte der Freien Software siehe *Grassmuck*, Freie Software, S. 211 ff.

<sup>5</sup> <<http://www.gnu.org/gnu/thegnuproject.html>>; <<http://www.gnu.org/gnu/gnu-history.html>>.

on der GNU General Public License (GPL) veröffentlicht, die auf früheren Entwürfen aufbaute und im Wesentlichen ebenfalls auf die Arbeit Richard Stallmans zurückgeht.

Mit der Verbindung der GNU-Software mit dem von Linus Torvalds entwickelten Betriebssystemkernel „Linux“ liegt seit Anfang der 90er Jahre ein funktionsfähiges Betriebssystem vor, das zunehmende Verbreitung findet und die Basis für zahlreiche Produkte auch von kommerziellen Anbietern bildet. Die steigende Bedeutung Freier Software in der Wirtschaft findet in der Open Source Initiative ihren Ausdruck. Im Zusammenhang mit der Freigabe des Netscape Browsers wurde ein neuer Begriff gesucht, der im Wirtschaftsleben einen positiveren Anklang als „free software“ finden sollte. Der 1998 erfundene Begriff „Open Source Software“ setzte sich umgehend durch. Innerhalb der Community werden seitdem mit den Begriffen „Freie Software“ und „Open Source Software“ unterschiedliche politische Linien verbunden. Während mit ersterem die Nutzerfreiheiten betont werden sollen, steht zweiterer für eine pragmatischere, weniger philosophische Richtung.

## **Teil 2. Immaterialgüterrechtlicher Schutz von Freier Software**

### ***1. Abgrenzung patentrechtlicher und urheberrechtlicher Schutz***

Patent- und Urheberrecht sind eigenständige Schutzrechte, die auch nebeneinander für eine einzige Software zur Anwendung kommen können. Beide Rechte haben eine unterschiedliche Schutzrichtung. Das Urheberrecht schützt ein Programm in seiner konkreten Programmierung und zwar in jeder Gestalt (§ 69a Absatz 1 UrhG). Nicht vom Urheberrecht erfasst sind dagegen Ideen und Grundsätze, die einem Element des Computerprogramms oder den Schnittstellen zugrunde liegen (siehe § 69a Absatz 2, Satz 2 UrhG). Allgemein bekannte mathematische Rechenregeln in Form eines Algorithmus sind daher ebenso wenig schutzfähig, wie die Idee, eine Aufgabe mit den Mitteln der Datenverarbeitung auf bestimmte Art und Weise zu lösen. Im Unterschied zum Urheberrecht schützt das Patentrecht nicht die Software an sich, sondern Verfahren, die durch die Software implementiert werden. Dabei erfasst der Schutzzumfang eines Softwarepatents auch und besonders die hinter einem Programm stehenden Ideen und

Algorithmen. Dies führt dazu, dass der Patentschutz häufig wesentlich weitergehend ist, als der Urheberrechtsschutz. Ist ein Algorithmus vom Recht aus dem Patent erfasst, wird dieser monopolisiert, also den ausschließlichen Rechten einer Person zugeordnet. Der Patentrechtsinhaber kann daher mit seinen Abwehrrechten (vgl. § 9 PatG) die Benutzung des Algorithmus durch einen Dritten verbieten, unabhängig davon, mit welchem Code dieser implementiert wird.

Ein weiterer Unterschied zwischen Patent- und Urheberrecht ist das im Patentrecht geltende Prioritäts- besser Neuheitsprinzip (siehe §§ 3, 6 PatG). Dies führt dazu, dass für jede Erfindung nur ein Patent erteilt werden kann. Erfinden mehrere Personen gleichzeitig und unabhängig voneinander das Gleiche, hat nur derjenige einen Anspruch auf das Patent, der dies zuerst angemeldet hat (vgl. § 6 Satz 3 PatG). Im Urheberrecht sind dagegen mehrere „zufällige Doppelschöpfungen“, wenn sie auch in der Praxis kaum vorkommen dürften, ohne weiteres schutzfähig.

Besonders bedeutsam für die Softwareindustrie ist, dass Rechte aus einem Patent nur dann geltend gemacht werden können, wenn das Patent durch formalen, hoheitlichen Akt erteilt wurde. Das Urheberrecht mit allen Befugnissen entsteht dagegen mit der „Schöpfung“, ohne dass es einer staatlichen oder sonstigen Zuerkennung bedarf. Damit ist der Urheberrechtsschutz kostenlos und ohne Verwaltungsaufwand, der Patentschutz hingegen nur über ein mit teilweise hohen Kosten verbundenes Verfahren zu erlangen.

## **II. Urheberrecht**

### **1. Grundlagen softwareurheberrechtlichen Schutzes**

Besondere urheberrechtliche Regelungen zu Computerprogrammen finden sich in den §§ 69a - 69g Urheberrechtsgesetz (UrhG). Der Gesetzgeber hat diesen Vorschriften bewusst einen eigenen Abschnitt im Gesetz gewidmet. Dies geht zum einen auf die Tatsache zurück, dass der Softwareschutz im Rahmen der Umsetzung einer EU-Richtlinie implementiert wurde. Zum anderen liegt die Erkenntnis zugrunde, dass Computerprogramme im Urheberrecht eine Sonderstellung einnehmen, die es gebietet, diese (z.T.) speziellen Regeln zu unterwerfen.

Dennoch: die wesentlichen Grundprinzipien des Urheberrechts finden auch bei Software Anwendung (siehe § 69a Absatz 4 UrhG). Dies gilt v.a. für den Schöpfungsgrundsatz nach § 7 UrhG, die Urheberpersönlichkeitsrechte in den §§ 12 - 14 UrhG (z.B. das Namensnennungs- und das Veröffentlichungsrecht), den Grundsatz der Unübertragbarkeit des Urheberrechts (§ 29 Abs. 1 UrhG) und die Schutzdauer, die nach § 64 UrhG 70 Jahre nach dem Tod des Urhebers beträgt. Die Anforderungen an den Urheberrechtsschutz von Software sind sehr gering. Es bedarf bei der Programmierung lediglich eines „gewissen Spielraums“, innerhalb dessen der Entwickler eine Anordnung und Auswahl der Elemente der Programmierung frei bestimmen kann.<sup>6</sup>

## **2. Lizenzierung als vertragliche Gestattung der Softwarenutzung**

Das Softwareurheberrecht entsteht stets bei dem Urheber, in der Regel also bei dem Programmierer. Wer fremde urheberrechtlich geschützte Werke nutzt, ohne hierzu befugt zu sein, kann auf Unterlassung und – bei Verschulden – auf Schadensersatz in Anspruch genommen werden und sich auch strafbar machen.

Die Befugnis zur Nutzung kann aus gesetzlichen Bestimmungen folgen. So sind nach den §§ 69d und 69e UrhG sog. *Benutzungshandlungen* gestattet. Vor allem die für ein bloßes „Ablaufenlassen“ eines Programms notwendigen Handlungen bedürfen hier nach nicht der Zustimmung des Urhebers. Die Beschränkungen des Softwareurheber-schutzes greifen darüber hinaus jedoch nur kurz. Es gilt daher die Regel, dass jede über das Ablaufenlassen eines Programms hinausgehende Nutzungshandlung (z.B. die Vervielfältigung, Verbreitung oder öffentliche (online) Zugänglichmachung) verboten ist, die nicht explizit erlaubt wurde.

Ist eine Nutzungshandlung nicht schon per Gesetz erlaubt, bedarf es hierfür des Erwerbs von Nutzungsrechten. Diese werden durch einen Lizenzvertrag erteilt. In Bezug auf die lizenzvertraglichen Vereinbarungen, also Art und Umfang der hierdurch eingeräumten Nutzungsrechte, unterscheiden sich proprietäre und freie Softwarelizenzen elementar. Während dem Nutzer bei proprietären Lizenzmodellen regelmäßig nur ver-einzelte, u.U. auf die jeweiligen Verwendung zugeschnittene, Nutzungsrechte gegen

---

<sup>6</sup> Vgl. OLG Karlsruhe ZUM 1995, S. 143 ff. - Bildschirmmasken.

eine Lizenzgebühr erteilt werden, eröffnen freie Softwarelizenzen wesentlich mehr Freiheiten bei der Verwendung der Programme. Jedermann, der die Lizenzbestimmungen akzeptiert, kann weitreichende Nutzungsrechte erwerben.

Der Abschluss eines Open Source Lizenzvertrags wird dadurch eingeleitet, dass der Rechtsinhaber sein Programm unter eine Open Source Lizenz stellt und dies bekannt gibt. Dies geschieht, indem im Programm (entweder während des Ablaufs oder im Quellcode) und/oder auf der Webseite, von der dieses herunter geladen werden kann, darauf hingewiesen wird, dass dies unter einer freien Softwarelizenz steht. Durch diesen Hinweis macht der Rechtsinhaber ein Angebot auf Einräumung eines einfachen (aber sehr weitgehenden) Nutzungsrechts an jedermann. Er verzichtet also durch die Unterstellung unter eine freie Softwarelizenz nicht etwa nicht auf seine Urheberrechte oder seine Verwertungsbefugnis.<sup>7</sup> Vielmehr benötigt er sein Recht weiterhin, um sich gegen Verletzungen der Pflichten aus den Lizenzbestimmungen durch den Lizenznehmer zu wehren. Hat der Nutzer die Möglichkeit, von den Lizenzbestimmungen Kenntnis zu nehmen und nimmt zustimmungsbedürftige Nutzungshandlungen mit dem Programm vor, kommt zwischen ihm und dem Rechtsinhaber ein Lizenzvertrag zustande. Durch die Vornahme dieser Handlungen nimmt der Lizenznehmer den Vertrag konkludent an; ein Zugang der Annahmeerklärung bei dem Rechtsinhaber ist gemäß § 151 BGB nicht erforderlich, da die Urheber bei der Teilnahme am Open Source Modell darauf verzichten.

Die – rechtlich verbindlichen – Bedingungen dieses Vertrags ergeben sich aus den jeweiligen Bestimmungen der freien Softwarelizenz. Der Nutzer erhält hierdurch Rechte, wird aber auch mehr oder weniger weitreichenden Pflichten unterworfen.<sup>8</sup>

#### **a) Die durch Open Source Lizenzen gewährten Rechte**

Obgleich die einzelnen freien Softwarelizenzen im Detail sehr unterschiedlich sind, herrscht in Bezug auf die Rechtsgewähr wesentliche Übereinstimmung. Dies ist darauf zurückzuführen, dass alle Open Source Lizenzen auf einem einheitlichen Grundprinzip

---

<sup>7</sup> *Dreier/Schulze*, Urheberrechtsgesetz, § 69a, Rz. 11; *Metzger/Jaeger*, GRUR Int. 1999, S. 839 (842).

<sup>8</sup> Zu den Details siehe unten S. 18 ff.

basieren. Dieses liegt darin, dass die hierunter vertriebenen Computerprogramme zur ungehinderten Nutzung durch die Allgemeinheit zur Verfügung stehen sollen.

Nach der „Free Software Definition“ handelt es sich nur dann um eine freie Softwarelizenz, wenn diese dem Nutzer das Recht gewährt, das Programm zu jedem Zweck zu nutzen; dessen Funktionsweise zu untersuchen und es an die eigenen Anforderungen anzupassen (also zu bearbeiten); es beliebig weiterzugeben und es zu verbessern und diese Verbesserungen der Allgemeinheit zur Verfügung zu stellen.<sup>9</sup> Urheberrechtlich ausgedrückt bedeutet dies, dass jede freie Softwarelizenz den Lizenznehmern das Vervielfältigungsrecht (vgl. §§ 69c Nr. 1, 16 UrhG), das Verbreitungsrecht (§§ 69c Nr. 3, 17 UrhG), das Recht auf öffentliche (online) Zugänglichmachung (vgl. §§ 69c Nr. 4, 19a UrhG) und das Bearbeitungsrecht (vgl. §§ 69c Nr. 2, 23 UrhG) gewährt.<sup>10</sup> Freie Software kann also von jedem, der die Lizenzbestimmungen akzeptiert, beliebig kopiert, weitergegeben, online gestellt und verändert werden.

Das bloße „Ablaufenlassen“ des Programms wird schon von der gesetzlichen Nutzungsbefugnis des § 69d UrhG gestattet, insofern ist keine gesonderte Lizenzierung erforderlich (vgl. auch Ziffer 0 Abs. 2 GPL).

## **b) Urheberschaft und Rechtsinhaberschaft bei Open Source Software: Der richtige Lizenzgeber**

Wer ein Programm unter eine freie Softwarelizenz stellen will, muss Inhaber der exklusiven (oder „ausschließlichen“, siehe § 31 Abs. 3 UrhG) Nutzungsrechte für alle durch die Lizenz betroffenen Nutzungsarten sein. Er muss also berechtigt sein, über die Rechte auch zu verfügen. Dies ist in der Regel der Programmierer, kann aber auch eine andere Person sein. Die Frage nach dem richtigen Lizenzgeber hängt also davon ab, wer das Programm entwickelt hat und ob den Programmierern (also den Urhebern) die ausschließlichen Nutzungsrechte hieran (noch) zustehen.<sup>11</sup>

---

<sup>9</sup> S.o. S. 1 f.

<sup>10</sup> Vgl. dazu auch ausführlicher unten S. 21 ff.

<sup>11</sup> Vgl. dazu auch unten S. 81 ff.

Die Urheberrechte entstehen stets mit dem Schöpfungsakt in der Person des Urhebers. Allerdings steht nach § 69b UrhG die Verwertungsbefugnis an Programmen, die von Arbeitnehmern oder Beamten bei der Wahrnehmung ihrer Aufgaben aus dem Arbeits- oder Dienstverhältnis geschaffen wurden, für alle Nutzungsarten allein dem Arbeitgeber zu. Dem Arbeitnehmer verbleiben allein die Persönlichkeitsrechte. Wo die Grenze zwischen solchen Arbeitnehmerwerken und anderen, außerhalb der dienstlichen Aufgaben geschaffenen Werken liegt, ist eine Frage des Einzelfalls. Es kommt dabei darauf an, ob ein enger innerer Zusammenhang zwischen der Programmerstellung und den „Aufgaben“ des Arbeitnehmers besteht. Die „Aufgaben“ ergeben sich dabei primär aus dem Arbeitsvertrag sowie aus tarifvertraglichen Regelungen, der betrieblichen Funktion des Arbeitnehmers, dem Berufsbild und der Üblichkeit. Ob das Programm hingegen inner- oder außerhalb der Arbeitszeit geschaffen wurde, ist nach wohl überwiegender Auffassung irrelevant. Handelt es sich um ein „Arbeitnehmerwerk“ gehen die notwendigen Rechte per Gesetz über, der Arbeitgeber muss also für die Vornahme von Nutzungshandlungen weder die Zustimmung des Programmierers einholen noch eine Vergütung hierfür bezahlen (der Erwerb der Rechte wird durch das Gehalt abgegolten). Auch ist er allein befugt, an dem Programm Lizenzen zu erteilen. An Software, die in privater Initiative entwickelt wurde verbleiben die Rechte hingegen stets beim Arbeitnehmer.

Sind an der Softwareentwicklung mehrere Programmierer beteiligt, die zusammen aufgrund eines gemeinsamen Konzepts ein Programm schreiben, so sind sie Miturheber (siehe § 8 Absatz 1 UrhG). Den Miturhebern steht die Verwertungsbefugnis an dem Gesamtprogramm stets gemeinsam zu, wenn hier nicht wiederum der soeben geschilderte Sonderfall vorliegt, dass es sich um ein „Arbeitnehmerwerk“ handelt, an dem der Arbeitgeber die ausschließlichen Nutzungsrechte hat. Soll eine Software, die in Miturheberschaft entwickelt wurde, unter eine Open Source Lizenz oder unter eine proprietäre Lizenz gestellt werden, müssen alle Programmierer zustimmen.

Von miturheberschaftlich geschaffenen Werken zu unterscheiden sind Programmentwicklungen, bei denen mehrere Personen einzelne Bestandteile einer Software unab-

hängig voneinander geschrieben haben, die wiederum eigenständig<sup>12</sup> verwertet werden können. Die Entwickler sind hier keine Miturheber, es handelt sich nicht um ein „gemeinsames Werk“ im Sinne des § 8 UrhG, sondern um eine Werkverbindung nach § 9 UrhG, deren Bestandteile auch alleine als Open Source Software freigegeben werden können. Nach den §§ 3, 69c Ziff. 2 Satz 2 UrhG kann im Übrigen auch ein Bearbeiter Urheber sein. Entwickelt ein Programmierer den Code weiter, den ein anderer geschrieben hat, erwirbt er an der veränderten Version des Programms u.U. ein eigenes Urheberrecht.<sup>13</sup> Will man diese geänderte Version nutzen oder unter eine freie Softwarelizenz stellen, müssen sowohl der Bearbeiter, als auch der oder die Urheber des Ursprungsprogramms dem zustimmen.

### c) Rechtserwerb bei Fehlern in der „Lizenzkette“

Ein Lizenzwerb kann nur dann stattfinden, wenn der Lizenzgeber tatsächlich Inhaber aller Rechte ist, die dem Lizenznehmer eingeräumt werden sollen. Dies gilt auch dann, wenn eine Lizenz über mehrere Personen übertragen wird, z.B. ein Programmierer zunächst ausschließliche Rechte an einer Software auf eine Softwarefirma überträgt (etwa bei einer Auftragsentwicklung) und diese die Rechte dann auf ein anderes Unternehmen weiter überträgt. Man spricht dann von einer Lizenzkette. Fehlt in dieser Kette ein Glied, z.B. weil der eine der Rechtsübertragungen unwirksam war, kann das Recht nicht mehr weitergegeben werden. Die Folge ist, dass auch die späteren Lizenznehmer keine Rechte erhalten können. Eine Art „gutgläubiger Erwerb“ von urheberrechtlichen Nutzungsrechten ist nach deutschem Recht ausgeschlossen.<sup>14</sup>

In den meisten Open Source Lizenzen wird diese Problematik dadurch umgangen, dass die Rechte nicht von einem Lizenznehmer zum anderen übertragen werden, sondern jeder Lizenznehmer die Rechte direkt vom Rechtsinhaber erwirbt. Hierin liegt ein wesentliches Prinzip des auto-distributiven Verbreitungssystems Freier Software. Eine solche Regelung findet sich z.B. in Ziff. 6 GPL. Verliert beispielsweise ein Lizenznehmer seine Rechte aus der Open Source Lizenz, weil er gegen die Lizenzbestimmungen versto-

---

<sup>12</sup> Die Frage, wann von einer eigenständigen bzw. selbständigen Verwertbarkeit auszugehen ist, stößt bei Computerprogrammen auf besondere Probleme, da Software stets darauf ausgerichtet ist, mit anderen Programmen und anderen Elementen einer Datenverarbeitungsanlage zusammenzuarbeiten.

<sup>13</sup> Ob ein Bearbeiter an der veränderten Version eines Computerprogramms eigene Urheberrechte erwirbt, hängt davon ab, ob die Bearbeitung eine persönliche geistige Schöpfung darstellt, vgl. § 3 Satz 1 UrhG.

<sup>14</sup> Vgl. BGH GRUR 1959, S. 200/203 - Der Heiligenhof.

Ben hat (siehe z.B. Ziff. 4 GPL), werden die Rechte anderer Lizenznehmer hierdurch auch dann nicht beeinträchtigt, wenn sie ihr Exemplar des Programms von diesem Lizenznehmer erhalten haben (etwa sich dieses von der Webseite des anderen heruntergeladen haben).

### **III. Patentrecht**

#### **1. Grundzüge patentrechtlichen Schutzes**

Das Patentrecht ist – wie auch das Urheberrecht – ein Territorialrecht. Das bedeutet, dass Patente stets nur für einen bestimmten Raum erteilt werden (z.B. haben deutsche Patente nur in Deutschland Wirkung). Das heutige Patentsystem sieht nationale, europäische und internationale Patente vor. Europäische Patente wirken in allen Staaten, die dem Europäischen Patentübereinkommen (EPÜ) beigetreten sind und für die sie erteilt wurden (siehe Art. 2 Absatz 2 EPÜ). Sie wirken also grenzüberschreitend. Das gleiche gilt für internationale Patente, die nach Maßgabe des Patent Cooperation Treaty (PCT) erteilt werden.

Computerimplementierte Erfindungen sind - wie auch solche aus anderen Bereichen - nur dann patentfähig, wenn sie eine Lehre zum technischen Handeln enthalten.<sup>15</sup> Der Begriff der „Technik“ wird weder im EPÜ noch im PatG definiert.

Den Entscheidungen des BGH und des Europäischen Patentamts (EPA) lassen sich allgemeine Grundsätze entnehmen, aus denen man ableiten kann, dass im Wesentlichen drei – unten näher beleuchtete – Grundtypen von Programmen als patentfähig anerkannt sind.<sup>16</sup> Allerdings befindet sich die Entscheidungspraxis nach wie vor im Fluss.

Letztlich bleibt die weitere Entwicklung abzuwarten, die insbesondere auch durch die Diskussion um eine EU-Richtlinie über den Schutz computerimplementierter Erfindungen beeinflusst werden dürfte. Die in dieser Richtlinie zu regelnde europaweite Haltung gegenüber der Patentierbarkeit von „computerimplementierten Erfindungen“ wird

---

<sup>15</sup> Mes, Patentgesetz, § 1, Rz. 57.

<sup>16</sup> Vgl. die Übersicht bei *Tauchert*, GRUR 1997, S. 149 (150 ff.). Ob ein konkretes Programm wirklich patentfähig ist, kann - selbst wenn es grundsätzlich diesen Grundtypen zuzuordnen ist - nur im Einzelfall unter Prüfung der materiellrechtlichen Voraussetzungen beurteilt werden.

momentan noch so kontrovers diskutiert, dass kaum absehbar ist, ob und gegebenenfalls in welchem Maße mit Verabschiedung der Richtlinie eine Ausweitung oder sogar Einschränkung des Softwarepatentschutzes einhergehen wird.<sup>17</sup>

## 2. Grundtypen von patentierbaren Computerprogramm

### a) Programme zur Steuerung klassisch-technischer, mechanischer Einheiten

Weitgehend unbestritten ist nach geltender Rechtslage die grundsätzliche Patentierbarkeit von Programmen, die klassisch-technische, mechanische Einheiten direkt ansteuern. Nach der Entscheidung des Bundesgerichtshofs „ABS“<sup>18</sup> liegt der technische Charakter solcher Programme darin, dass diese in einen technischen Gesamtablauf eingebettet sind. Das Programm ist hier mit im klassisch-mechanischen Sinn als technisch verstandenen Einheiten verknüpft und dient deren Steuerung. Beispiele aus der Rechtsprechung sind Steuerungsprogramme für Tauchcomputer<sup>19</sup> oder das besagte automatische Bremssystem (Antiblockiersystem). In beiden Fällen steuert eine Software unmittelbar die mechanischen Teile der Technologie.

### b) Neue, erfinderische Betriebsweise einer EDV-Anlage

Der zweite Grundtypus patentierbarer Software zeichnet sich dadurch aus, dass sie einen neuen, erfinderischen Aufbau einer Datenverarbeitungsanlage lehren oder Anweisungen enthalten, die Anlage auf eine neue, bisher nicht übliche und auch nicht nahe liegende Weise zu benutzen.<sup>20</sup> Der Bundesgerichtshof hat die Anforderungen für eine Patentierbarkeit solcher Programme in der Entscheidung „Seitenpuffer“ definiert.<sup>21</sup> In Rede stand hier die Erfindung eines Verfahrens, den Arbeitsspeicher einer Datenverarbeitungsanlage effektiver zu nutzen, indem die während eines Prozesses benötigten Speicherseiten möglichst im Seitenpuffer und nicht im langsameren Hauptspeicher gespeichert werden. Der BGH sah eine solche Software als „technisch“ an, da sie die Funktionsfähigkeit der Datenverarbeitungsanlage unmittelbar positiv beeinflusse. Diesem zweiten Grundtypus patentierbarer Software gehören alle Programme an, die die Steu-

---

<sup>17</sup> Siehe hierzu *Metzger*, CR 2003, S. 313 ff.

<sup>18</sup> GRUR 1980, S. 949.

<sup>19</sup> BGH GRUR 1992, S. 430.

<sup>20</sup> *Tauchert*, GRUR 1997, S. 159 (151).

<sup>21</sup> BGH GRUR 1992, S. 33.

ereinheit einer Datenverarbeitungsanlage (v.a. also die Komponenten von Betriebssystemen) unmittelbar beeinflussen.<sup>22</sup> Nicht hierunter fallen reine Anwendungsprogramme, die nicht der Steuerung der Datenverarbeitungsanlage selbst dienen.

### c) Anwendungsprogramme

Klassische (Büro-) Anwendungsprogramme galten lange Zeit nicht als patentfähig, da sie nicht der Steuerung von mechanischen oder elektronischen Einheiten dienen (also keine Beherrschung von Naturkräften erfordern). In seiner Entscheidung „Logikverifikation“ aus dem Jahr 1999 ist der BGH von dieser Linie erstmals abgewichen.<sup>23</sup> Hiernach soll es für die Technizität von Software ausreichen, wenn diese auf einer Lehre basiert, die „durch eine Erkenntnis geprägt ist, die auf technischen Überlegungen beruht“. Der Unterschied dieser Entscheidung zu der älteren Rechtsprechung liegt darin, dass es hiernach für die Patentierbarkeit von Computerprogrammen nicht (mehr) darauf ankommen soll, ob diese der Steuerung mechanischer Prozesse oder einer Datenverarbeitungsanlage dienen. Vielmehr soll es genügen, dass dem Programm technische Erkenntnisse zu Grunde liegen. Der „technische Charakter“ eines Programms ergibt sich hiernach also aus dem Umstand, dass die zugrunde liegenden Erkenntnisse dem technischen und nicht etwa dem naturwissenschaftlichen oder schöpferischen Bereich zuzuordnen sind. Nach dieser Entscheidung wird man zukünftig davon ausgehen können, dass sich die Patentierbarkeit von Computerprogrammen nicht mehr nur auf mechanische oder elektronische Steuerungsprogramme beschränkt.

In späteren Entscheidungen (unter anderem „Sprachanalyseeinrichtung“<sup>24</sup> und „Suche fehlerhafter Zeichenketten“<sup>25</sup>) wurden die Anforderungen an die Patentierbarkeit in der Folge weiter präzisiert und zum Teil wieder restriktiver gehandhabt.

Insgesamt zeigt sich hier, dass über die Ausdehnung des Technikbegriffs in der Entscheidung „Logikverifikation“ hinaus eine weitere Ausdehnung insbesondere in den Bereich des Schutzes reiner „Geschäftsmethoden“ hinein nicht zu erwarten ist. Zu nen-

---

<sup>22</sup> Tauchert, GRUR 1997, S. 159 (151).

<sup>23</sup> BGH GRUR 2000, 498, 500 mit Anmerkung *Betten*.

<sup>24</sup> BGH GRUR 2000, 1007 mit Anmerkung *Betten*.

<sup>25</sup> BGH GRUR 2002, 143 – Suche fehlerhafter Zeichenketten

nen ist insoweit insbesondere die Entscheidung „elektronischer Zahlungsverkehr“<sup>26</sup>. Dort entschied der BGH, dass über den Vorschlag hinaus, einen Computer zur Lösung eines bestimmten Problems einzusetzen, weitere Voraussetzungen erforderlich sind. Technizität liege nur dann vor, wenn der Patentanspruch darüber hinaus Anweisungen enthält, denen ein konkretes technisches Problem zugrunde liegt, so dass bei der Prüfung auf erfinderische Tätigkeit eine Aussage darüber möglich ist, ob eine Bereicherung der Technik vorliegt, die einen Patentschutz rechtfertigt.

### 3. Mögliche Problemfelder

Immer häufiger wird heutzutage davor gewarnt, dass eine Ausweitung der Patentvergabepraxis auf Computerprogramme zu Gefahren für die Entwicklung und Distribution von Open Source Software führen würde. Auch in der GPL wird hierauf explizit hingewiesen. In der Einleitung zum Lizenztext heißt es: „Finally, any free program is threatened constantly by software patents.“ Die im Folgenden angesprochenen Punkte zeigen auf, welche Problemkreise im Bereich der Freien Software diskutiert werden.

#### a) „Proprietäre“ Patentanmeldungen auf Freie Software

Für die Programmierer Freier Software kann die Gefahr entstehen, dass ein Dritter ein Softwarepatent auf eine Erfindung anmeldet, die bereits im Open Source Bereich gemacht und die bereits veröffentlicht wurde. Sie ist mit der Veröffentlichung bereits Stand der Technik.

Da eine Erfindung nur zum Patent angemeldet werden kann, wenn sie „neu“ ist, kann theoretisch ein Dritter kein Patent erlangen auf Verfahren, die bereits zuvor im Bereich Freier Software gemacht wurden. In der Praxis kann sich aber das Problem ergeben, dass die Patentämter dennoch für eine solche Erfindung ein Patent an einen Dritten erteilen, weil sie keine Kenntnis von der bereits bestehenden Freien Software haben, insbesondere weil der Stand der Technik im Bereich Open Source möglicherweise noch nicht hinreichend dokumentiert ist.

---

<sup>26</sup> BGH, Urteil vom 29.04.2004- I ZR 233/01 – Elektronischer Zahlungsverkehr.

Hier wird man allerdings zu berücksichtigen haben, dass solche Probleme unzureichender Dokumentation durch vergleichsweise einfache Schritte auf Seiten der Open Source Entwickler beseitigt werden können. Dies kann etwa geschehen (und geschieht bereits) durch den Einsatz von CVS Systemen, die den Zeitpunkt der Implementierung einzelner Funktionalitäten exakt rückverfolgbar machen und durch die kurze Beschreibung der implementierten Funktionalitäten und der zugrundeliegenden Verfahren in entsprechenden Datenbanken etc.

Zu dieser Problemstellung ist allerdings anzumerken, dass die Gefahr einer „Annektierung“ von Computerprogrammen durch fremde Softwarepatente nicht spezifisch Open Source Software betrifft. Auch die Entwickler „proprietärer“ Programme sind diesem Risiko ausgesetzt.

#### **b) Das Risiko von Patentverletzungsprozessen**

Mit der zunehmenden Zahl der angemeldeten Patente, steigt die Gefahr, durch die Programmierung und Verwendung von (Open Source) Code solche Patente (unbewusst) zu verletzen. Hier ist zu befürchten, dass dies gerade kleinere Softwarefirmen oder freie Programmierer behindert, da diese kaum die Möglichkeiten haben, vor der Codeentwicklung und -verwendung eine kosten- und zeitaufwändige Patentrecherche durchzuführen oder jedenfalls ein eigenes Patentportfolio als Basis für gegenseitige Lizenzen mit anderen Unternehmen einbringen zu können.

Auch hierbei handelt es sich allerdings nicht um ein Open Source spezifisches Problem. Die durch fortschreitende Patentierung von Computerprogrammen herbeigeführte Rechtsunsicherheit beeinträchtigt die gesamte Softwarebranche unabhängig von dem für die jeweilige Software gewählten Lizenzmodell. Eine effektive Lösung für das Problem der durch Softwarepatente ausgelösten Unübersichtlichkeit der Rechtslage wurde bislang allerdings noch nicht gefunden.

#### **c) Patente im System der Open Source Lizenzierung**

Schließlich wird befürchtet, dass bei einer Ausweitung der Patentierungsmöglichkeiten für Software auf Freie Programme zunehmend Patente von Rechtsinhabern (Softwarefirmen, Entwickler) aus dem Kreis der Open Source Entwickler angemeldet werden und

diese Patente als Druckmittel eingesetzt werden, um die Software unter restriktiveren Lizenzbedingungen zu vertreiben. Diese Befürchtung betrifft besonders den Fall, dass Bearbeiter von Freier Software auf Verfahren Patente anmelden, die sie in Weiterentwicklungen der Open Source Software implementiert haben.

Durch die Open Source Lizenzen wird eine Anmeldung von Patenten auf Verfahren, die im weiterentwickelten Code implementiert werden, nicht ausdrücklich untersagt. Die GPL beispielsweise enthält keine Regelung, die ein derartiges Verbot konkret aussprechen würde.<sup>27</sup>

Gleichzeitig enthalten die Lizenzen jedoch vielfach auch keine Regelungen zu der Frage, in welchem Umfang den anderen Open Source Entwicklern und Nutzern eine Lizenz an dem Patent zu erteilen ist. Anders verhält es sich unter anderem mit der Mozilla Public Licence (§ 2.1(b) und 2.1(d) MPL) als auch mit den für den deutschen Rechtsraum konzipierten freien Softwarelizenzen,<sup>28</sup> die jeweils ausdrückliche Kollisionsklauseln für Softwarepatente enthalten. Diese Lizenzen verbieten zwar nicht, Patente anzumelden. Sie verpflichten jedoch den Bearbeiter, jedem Nutzer eine Lizenz einzuräumen, welche die freie Nutzung des Programms in dem durch die Lizenz ansonsten gestatteten Umfang (also freie Vervielfältigung, Verbreitung und öffentliche Wiedergabe) gewährleistet. Erhält ein Bearbeiter mithin ein Patent auf seine Weiterentwicklung, muss er nach diesen Lizenzen allen Nutzern neben urheberrechtlichen auch patentrechtliche Lizenzen einräumen.

Im Ergebnis führen Patente auf Weiterentwicklungen von Freier Software zu keinen besonderen Problemen, weil der Patentinhaber bei der Weitergabe des so weiterentwickelten Programms zugleich sein Patent lizenzieren muss<sup>29</sup>. Außerhalb des Codes, der einer Copyleft-Lizenz unterstellt ist, kann er aber das Patent unbeschränkt nutzen. Denn

---

<sup>27</sup> Vgl. *Jaeger/Metzger*, Open Source Software, S. 122. Gleiches gilt für die Mozilla Public Licence (MPL) und die Netscape Public Licence, die ihren Ziff. 2.2.b) eine Anmeldung von Patenten auf Fortentwicklungen sogar ausdrücklich gestatten.

<sup>28</sup> Gemeint sind die „Bremer Lizenz für freie Softwarebibliotheken“ (verfügbar unter <<http://www.ostc.de/Bremer-Lizenz.pdf>>) und die „Deutsche Freie Software Lizenz“ (verfügbar unter <<http://www.d-fsl.de>>). Patentklauseln finden sich in § 8 Abs. 2 der „Bremer Lizenz“ und in § 3 Abs. 6 der „Deutschen Freien Software Lizenz“.

<sup>29</sup> Praktisch wurde diese Problematik in dem Fall „RTLinux“, vgl. <<http://www.gnu.org/philosophy/rtnlinux-patent.html>>.

da durch Patente Erfindungen geschützt werden, können diese auch durch vollständig anderen (proprietären) Code implementiert werden. Entsprechendes gilt für Software unter Non-Copyleft-Lizenzen, die dadurch „proprietär“ werden kann.<sup>30</sup>

#### **IV. *Praktische Hinweise: Was ist bei Programmierung, Vertrieb und Verwertung von Open Source Software zu berücksichtigen, um Patent- und Urheberrechtsverletzungen zu vermeiden?***

Will man ein Programm unter eine Open Source Lizenz stellen, ist es in erster Linie erforderlich, über die ausschließlichen, inhaltlich, zeitlich und räumlich unbeschränkten Nutzungsrechte zu verfügen. Dies ist dann der Fall, wenn man selbst alleiniger Urheber der Software ist oder Arbeitgeber des Urhebers oder sich explizit die ausschließlichen Nutzungsrechte an dem Programm abtreten lässt. Ist man Entwickler in einem Team, müssen sich alle gemeinsam für die Unterstellung des Programms unter die Open Source Lizenz entscheiden. Soll einer der Entwickler alleine als Lizenzgeber auftreten, müssen ihm die anderen Mitentwickler zunächst ihre Rechte übertragen.

Um sich gegen die nachträgliche Anmeldung von Softwarepatenten durch Dritte zu schützen, ist es ratsam, den eigenen Code zu veröffentlichen.<sup>31</sup> Da nur solche Erfindungen patentfähig sind, die noch nicht zum „Stand der Technik“ gehören und damit neu sind, kann ein Programm, das Erfindungen enthält, die schon einmal irgendwo auf der Welt veröffentlicht waren, nicht zum Patent angemeldet werden. Für eine solche Vorveröffentlichung genügt es, den Programmcode auf beliebige Weise (etwa im Internet, in öffentlich zugänglichen Mailinglisten, in Zeitschriften, Büchern, Vorträgen, Sendungen oder öffentlich geführten Gesprächen) der Öffentlichkeit zugänglich zu machen.<sup>32</sup> Dabei ist darauf zu achten, dass die Publikation mit Zeitpunkt der Veröffentlichung beweiskräftig dokumentiert und auf eine Weise vorgenommen wird, die es dem Patentamt bei einer Recherche ermöglicht, hiervon Kenntnis zu nehmen. Vor diesem Hintergrund ist die Veröffentlichung in einem Medium, in dem (möglichst sämtliche) Open Source Softwareentwicklungen laufend dokumentiert würden, vorteilhaft.

---

<sup>30</sup> Vgl. unten S. 23.

<sup>31</sup> Vgl. Jaeger/Metzger, Open Source Software, S. 114.

<sup>32</sup> Busse-Keukenschrijver, Kommentar zum PatG, 6. Auflage 2003, § 3, Rz. 76.

Um der Gefahr von unbewussten Patentverletzungsprozessen effektiv entgegenzuwirken, kann ein sicherer Schutz – sofern finanzierbar – nur durch eine ausführliche Patentrecherche hergestellt werden.<sup>33</sup> Dies empfiehlt sich insbesondere bei solchen Programmen, die den o.g. „Grundtypen“ patentierbarer Software zugeordnet werden können. Bei diesen ist die Wahrscheinlichkeit eines entgegenstehenden Patents höher, als bei anderen Programmarten.

### **Teil 3. Lizenzrechtliche Aspekte der Freien Software**

#### ***I. Einführung***

Open Source Lizenzen gestatten es jedermann, der die Lizenzbedingungen akzeptiert, die Software zu bearbeiten, sowie diese in unveränderter oder bearbeiteter Form zu vervielfältigen, zu verbreiten und öffentlich zugänglich zu machen. Anders als bei sonstigen Softwareentwicklungs- und Softwarevermarktungskonzepten erhält damit jedermann die Möglichkeit, die Software nicht nur zur Erfüllung eigener softwarespezifischer Aufgaben zu benutzen, sondern auch „Entwicklungs- und Vertriebsrechte“ zu erwerben. Um eine Bearbeitung durch jedermann auch tatsächlich zu ermöglichen, wird zugleich der Quelltext der Software offengelegt.

Das Konzept Freier Software beruht dabei weder auf der Verwendung einer einzigen Lizenz für alle Programme, noch auf der detaillierten und bindenden Regelung sämtlicher im Zusammenhang mit der Lizenzierung von Computerprogrammen auftretenden Fragen. Vielmehr werden in der „Open Source Definition“ und der „Free Software Definition“ allein bestimmte Rahmenbedingungen vorgegeben, die Voraussetzung dafür sein sollen, dass ein Computerprogramm als Open Source Software bzw. Freie Software bezeichnet werden kann.<sup>34</sup> Innerhalb dieses Rahmens verbleiben Gestaltungsspielräume.

Die bestehenden Gestaltungsspielräume haben dazu geführt, dass eine Vielzahl sehr unterschiedlicher Lizenzen erstellt wurden, die teilweise spezifische Klauseln enthalten,

---

<sup>33</sup> Recherchen zum Stand der Technik werden auch von den Patentämtern durchgeführt.

<sup>34</sup> Vgl. dazu bereits oben S. 1 f.

die auf die Besonderheiten bestimmter Projekte abgestimmt sind. Gleichzeitig haben sich jedoch bestimmte „Familien“ von Open Source Lizenzen herausgebildet, die sich insbesondere danach systematisieren lassen, wie in den Lizenzen der Vertrieb modifizierter Versionen des Computerprogramms geregelt wird. Die in der Praxis wichtigsten Lizenzfamilien sind dabei die „Lizenzen ohne Copyleft-Effekt“, die „Lizenzen mit strengem Copyleft-Effekt“ sowie „Lizenzen mit beschränktem Copyleft-Effekt“:

- Lizenzen ohne Copyleft-Effekt zeichnen sich dadurch aus, dass sie dem Lizenznehmer alle Freiheiten einer Open Source Lizenz einräumen und für Bearbeitungen der Software keine Bedingungen hinsichtlich des zu verwendenden Lizenztyps enthalten. Bearbeitungen können insbesondere auch proprietär lizenziert werden.
- Bei Lizenzen mit strengem Copyleft-Effekt wird der Lizenznehmer verpflichtet, von der ursprünglichen Software abgeleitete Werke stets unter den Bedingungen der Ursprungslizenz weiterzuverbreiten. Dieses Modell, wonach auch Weiterentwicklungen stets „frei“ bleiben müssen, wird als „Copyleft“ bezeichnet.
- Lizenzen mit beschränktem Copyleft-Effekt gewähren dem Nutzer bestimmte „Ausnahmen“ von der strengen Bindung an das Copyleft, die eine Kombination mit Softwarekomponenten unter abweichenden Lizenzbedingungen erlaubt.

## **II. Allgemeine Hinweise**

### **1. Rechte aus der Open Source Lizenz**

Dem Lizenznehmer einer Open Source Lizenz werden unterschiedliche Rechte gewährt, insbesondere die Rechte, das Programm zu vervielfältigen, zu bearbeiten und in unveränderter oder veränderter Version zu vertreiben. Urheberrechtlich handelt es sich dabei um die Einräumung eines einfachen (nichtexklusiven) Nutzungsrechts gem. § 31 Abs. 2 UrhG.<sup>35</sup>

---

<sup>35</sup> Allg. Ansicht, vgl. nur *Schiffner*, Open Source Software, S. 154; *Metzger/Jaeger*, GRUR Int. 1999, 839, 842 f.; *Spindler*, VSI-Gutachten, S. 27; *ders.* in: *Spindler (Hrsg.)*, Rechtsfragen bei Open Source, Kap. C, Rz. 23; *Koch*, CR 2000, 333, 335.

Dieses einfache Nutzungsrecht erhält der Lizenznehmer mit Abschluss der Open Source Lizenz stets direkt vom Urheber, und zwar auch dann, wenn der Nutzer die eigentliche Programmkopie von einem Dritten erhält. Die Open Source Lizenz stellt sich dabei als Angebot an „jedermann“ zum Abschluss eines entsprechenden Vertrages dar. Der Nutzer nimmt dieses Angebot in dem Moment durch schlüssiges Verhalten an, in dem er die Software vervielfältigt, verändert oder verbreitet. Eines Zugangs der Annahmeerklärung beim Urheber bedarf es dabei gemäß § 151 S. 1 BGB nicht.<sup>36</sup>

## 2. Wann muss die Open Source Lizenz abgeschlossen werden?

In der juristischen Diskussion ist streitig, zu welchem Zeitpunkt ein Abschluss der Open Source Lizenz erforderlich wird.<sup>37</sup> Im Ergebnis ist dabei davon auszugehen, dass der Nutzer erst in dem Moment eine Open Source Lizenz abschließen muss, in er über das schlichte Ablaufenlassen des Programms hinausgehende Handlungen vornehmen möchte. Dafür spricht zunächst der Wortlaut zahlreicher Open Source Lizenzen. So heisst es etwa in Ziffer 0 Abs. 2 GPL: *“Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted [...]”*. Hier wird also eine deutliche Abgrenzung zwischen der lizenzpflichtigen Verwendung und dem schlichten Ablaufenlassen gezogen. Zugleich bestimmt § 69d Abs. 1 UrhG, dass diejenigen Handlungen keiner Zustimmung durch den Rechtsinhaber bedürfen, die zur bestimmungsgemäßen Benutzung des Computerprogramms erforderlich sind. Damit sind die zur Benutzung erforderlichen Vervielfältigungen, zum Beispiel im Arbeitsspeicher, schon gesetzlich gestattet. Daher bedarf es für die bloße Benutzung einer Software keiner (Open Source) Lizenz; die mit dem Erwerb einhergehende Berechtigung des Erwerbers reicht regelmäßig dafür aus.

---

<sup>36</sup> Die Rechtsinhaber verzichten im Rahmen des Open Source Softwareentwicklungs- und Softwarevermarktungsmodells auf den Zugang einer entsprechenden Willenserklärung, vgl. dazu ausführlicher *Jaeger/Metzger*, Open Source Software, S. 33.

<sup>37</sup> Für eine Verpflichtung zum Abschluss der Open Source Lizenz direkt bei Erwerb der Programmkopie wohl *Spindler*, VSI-Gutachten, S. 43 f. Für eine Verpflichtung zum Abschluss der Lizenz im Falle der Vornahme von über die Programm Benutzung hinausgehenden Handlungen u.a. *Schiffner*, Open Source Software, S. 215; *Jaeger/Metzger*, Open Source Software, S. 161.

### 3. Welche Rechte werden gewährt?

Open Source Lizenzen gewähren vielfach die Rechte „to copy, to modify and to distribute“.<sup>38</sup> Unstreitig wird damit jedenfalls das Vervielfältigungsrecht (§ 16, § 69c Nr. 1 UrhG), das Verbreitungsrecht (§ 17, § 69c Nr. 3 UrhG) und das Bearbeitungsrecht (§ 23, § 69c Nr. 2 UrhG) erfasst.

#### a) Zugänglichmachung im Internet?

Ob dort, wo deutsches Recht anwendbar ist, auch das Recht zur Zugänglichmachung der Software im Internet mitgewährt wird, ist in der juristischen Diskussion teilweise bezweifelt worden. Die öffentliche Zugänglichmachung stelle ein eigenes Verwertungsrecht dar, das in der Lizenz nicht genannt werde.<sup>39</sup> Die Frage nach der Einräumung des Rechts zur öffentlichen Zugänglichmachung ist dabei von erheblicher praktischer Bedeutung, erfolgt doch der Vertrieb von Open Source Software zu einem erheblichen Teil in der Weise, dass diese zum Download im Internet bereitgestellt wird.<sup>40</sup>

Begründet wird die Ansicht, dass die Zugänglichmachung im Internet nach deutschem Recht nicht gestattet werde, damit, dass der Begriff des „Verbreitens“ nur die körperliche Weitergabe (d.h. die Weitergabe eines Datenträgers) erfassen würde, nicht hingegen die (unkörperliche) Zurverfügungstellung im oder die Übertragung über das Internet.<sup>41</sup> Die zugrundeliegende Argumentation überzeugt jedoch nicht:<sup>42</sup> Sie vernachlässigt den Umstand, dass der Begriff „distribute“ ausgelegt werden muss. Dabei ist es nicht zulässig, einen Ausdruck einer fremdsprachigen Lizenz in einer ganz bestimmten Weise zu übersetzen und aus dem Wortlaut der Übersetzung bestimmte Beschränkungen für die Reichweite der gewährten Rechte zu postulieren. Immerhin ist es ebenso möglich, den Begriff „distribution“ nicht als „Verbreitung“ sondern als „Vertrieb“ zu übersetzen. Dann können mit diesem Begriff sowohl die Verbreitungshandlungen im engeren Sinne als auch das Zugänglichmachen im Internet erfasst werden. Gerade dies entspricht auch dem Parteiwillen. So ist von den Verwendern der Open Source Lizenzen intendiert, dass

<sup>38</sup> Vgl. z.B. Ziff. 1-3 GPL.

<sup>39</sup> So insbesondere Koch, CR 2000, 333, 338.

<sup>40</sup> § 2 der Open Source Definition sieht die Bereitstellung zum Download im Internet sogar als bevorzugte Form der Quellcodeüberlassung an, wenn dieser nicht gemeinsam mit dem Objektprogramm vertrieben wurde.

<sup>41</sup> Koch, CR 2000, 333, 338.

<sup>42</sup> So die ganz überwiegende Auffassung, vgl. z.B. Omselsin: Schertz/Omsels (Hrsg.), S. 141, 158, Jaeger/Metzger, Open Source Software, S. 33, wohl auch Wandtke/Bullinger-Grützmaker, Urheberrechtsgesetz, § 69c, Rz. 61.

gerade auch die Bereitstellung zum Download möglich sein soll, da dies dem üblichen Vertriebsweg entspricht.<sup>43</sup> Hinzu kommt schließlich, dass nach US-amerikanischem Verständnis, welches den Lizenzen bei Erstellung regelmäßig zugrunde lag, der Begriff „to distribute“ gerade auch die unkörperliche Verwertung erfasst.<sup>44</sup>

### **b) Ist Vermietung gestattet?**

Von der Einräumung des Rechts „to distribute“ ist nach vorzugswürdiger Ansicht auch das Vermietrecht miterfasst.<sup>45</sup> Dies ist für die Praxis durchaus von Bedeutung, wird doch Open Source Software teilweise – z.B. als Teil einer Datenverarbeitungsanlage – zeitlich befristet gegen Entgelt überlassen. Dabei spricht insbesondere der Zweck der Lizenz, den Vertrieb der Open Source Software möglichst einfach und weitgehend zu gestatten, für eine Auslegung, die unter den Begriff „distribution“ auch eine Einräumung des Vermietrechts als Teil des Verbreitungsrechts fasst.<sup>46</sup>

### **c) Ist Application Service Providing gestattet?**

Problematischer erscheint demgegenüber die Nutzbarmachung von Open Source Software im Wege des sogenannten Application Service Providing (ASP). Hier ist zu unterscheiden: Open Source Produkte, die vollständig in jüngerer Zeit geschrieben wurden, sind innerhalb eines Geschäftsmodells „Application Service Providing“ wohl ohne weiteres einsetzbar. Entsprechende Handlungen sind dem Recht der öffentlichen Zugänglichmachung zuzuordnen. Dieses Recht wird – wie bereits oben ausgeführt wurde – durch die Befugnisse „to distribute“ gewährt.

Bei älteren Open Source Produkten besteht hingegen ein Sonderproblem. § 31 Abs. 4 UrhG bestimmt, dass die Einräumung von Nutzungsrechten für noch nicht bekannte „Nutzungsarten“ unwirksam ist. Nutzungsart ist dabei die wirtschaftlich-technisch selbst-

---

<sup>43</sup> Bei der GPL ergibt sich dies z.B. aus Ziffer 3 am Ende: „If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.“

<sup>44</sup> Vgl. dazu ausführlicher *Jaeger/Metzger*, Open Source Software, S. 33.

<sup>45</sup> Für die Zulässigkeit einer Vermietung *Jaeger/Metzger*, Open Source Software, S. 33 f.; zweifelnd demgegenüber u.a. *Spindler*, VSI-Gutachten, S. 45 f.; *ders.* in: *Spindler (Hrsg.)*, Rechtsfragen bei Open Source, Kap. C, Rz. 80 f.; ablehnend *Koch*, CR 2000, 333, 338.

<sup>46</sup> Weitere Argumente bei *Jaeger/Metzger*, Open Source Software, S. 33 f.

ständige und abgrenzbare Art und Weise der Nutzung.<sup>47</sup> Viele ordnen Application Service Providing als eine solche eigenständige Nutzungsart ein und heben hervor, dass diese jedenfalls bis Mitte der neunziger Jahre nicht bekannt gewesen sei.<sup>48</sup> Daher könne bei Open Source Software, die vorher geschrieben wurde, eine diesbezügliche Rechtseinräumung nicht stattgefunden haben, solange der betreffende Urheber diese nicht ausdrücklich nachhole.<sup>49</sup> Die Nutzbarmachung für Zwecke des Application Service Providing ohne eine solche nachgeholtene Rechtseinräumung sei in diesen Fällen unzulässig.

Hier ist allerdings zu berücksichtigen, dass entsprechende Fragen in der Vergangenheit in der Praxis weder bei Freier Software noch bei proprietären Programmen Relevanz erlangt haben.

### **III. Rechte und Pflichten wichtiger Open Source Lizenzen**

#### **1. BSDartige Lizenzen**

Eine bedeutsame Gruppe innerhalb der Open Source Lizenzen sind die „BSDartigen“ Lizenzen. Sie gehören zur Familie der Lizenzen ohne Copyleft-Effekt. Die erste BSD-Lizenz stammt aus dem Jahr 1989. Mittlerweile liegen verschiedene Varianten dieser Lizenz vor.<sup>50</sup> Sie hat zudem als Vorbild für zahlreiche ähnliche Softwarelizenzen gedient, z.B. die Apache License,<sup>51</sup> die X Window System License,<sup>52</sup> die W3C Software Notice and License<sup>53</sup> und die OpenLDAP Public License.<sup>54</sup>

---

47 Vgl. zum Begriff der Nutzungsart aus der Rechtsprechung BGH GRUR 1986, 62, 65 – GEMA-Vermutung I, BGH GRUR 1990, 669, 671 – Bibelreproduktion, BGH GRUR 1992, 310, 311 – Taschenbuch-Lizenz, BGH GRUR 1997, 464, 465 – CB-infobank II, aus der Literatur z.B. *Ulmer*, Urheber- und Verlagsrecht, § 84 I 3. (S. 362 f.), *Schri-cker/Schricker*, Vor §§ 28 ff., Rz. 52.

48 *Bettinger/Scheffelt*, CR 2001, 729, 735. Eingehender zur Frage des Application Service Providing bei Open Source Software *Jaeger/Metzger*, Open Source Software, S. 34, *Spindler*, VSI-Gutachten, S. 46.

49 Dieses Problem gilt entsprechend auch für proprietäre Programme, die alte Sourcecode-Bestandteile enthalten.

50 4.4 BSD Copyright, <<http://www.de.freebsd.org/copyright/license.html>>; FreeBSD Copyright (Modifizierte BSD-Lizenz), <<http://www.de.freebsd.org/copyright/freebsd-license.html>>; BSD License (Original), <<http://www.xfree86.org/3.3.6/COPYRIGHT2.html#6>>.

51 Apache License (v. 2.0), <<http://www.apache.org/licenses/LICENSE-2.0.html>>.

52 X Window System License, <<http://www.x.org/Downloads/terms.html>>.

53 W3C Software Notice and License, <<http://www.w3.org/Consortium/Legal/copyright-software.html>>.

54 OpenLDAP Public License (v. 2.7), <<http://www.openldap.org/software/release/license.html>>.

## a) Rechte

Wie bei allen Open Source Lizenzen werden dem Lizenznehmer – wie bereits oben ausführlicher dargestellt – bestimmte Rechte zur Nutzung der Software erteilt: Er darf diese vervielfältigen, verändern und unveränderte sowie veränderte Versionen off- und online vertreiben. BSD-spezifische Besonderheiten bestehen hier nicht.

## b) Pflichten

Die zentralen Pflichten des Lizenznehmers unterscheiden sich danach, ob der Nutzer allein den Vertrieb unveränderter Vervielfältigungsstücke im Objektcode oder Sourcecode anstrebt, oder ob er Weiterentwicklungen an der Software vornehmen und die modifizierten Programme vertreiben möchte. Weiter ist zu berücksichtigen, dass einzelne BSDartige Lizenzen besondere Lizenzbedingungen aufstellen.<sup>55</sup>

### **(1) Pflichten beim Vertrieb unveränderter Versionen**

Für den Vertrieb unveränderter Programmversionen sehen die BSDartigen Lizenzen wie alle freien Softwarelizenzen regelmäßig bestimmte Pflichten vor, die in erster Linie dem Zweck dienen, auch dem Erwerber der Programmkopie die Möglichkeit zu verschaffen, vom Open Source Modell zu profitieren. Die Pflichten unterscheiden sich geringfügig danach, ob die Software im Sourcecode oder im Objektcode weitergegeben wird:

**Verbreitung im Sourcecode.** Der Vertrieb unveränderter Programmkopien der Software wird an die Nutzerpflicht „gebunden“, jedem Erwerber der Software zugleich auch eine Kopie der entsprechenden Open Source Lizenz zu verschaffen, unter welcher die Software vertrieben wird.<sup>56</sup> Damit soll jeder Erwerber die Möglichkeit erhalten, durch Annahme des in den Lizenzen enthaltenen Angebots der Rechteinhaber zum Abschluss einer der Lizenz selbst die Rechte zum Kopieren, Bearbeiten und Verbreiten zu erhalten.

Neben der Verpflichtung zur Weitergabe der Lizenz sehen die BSD-artigen Lizenzen zugleich vor, dass bei Vertrieb des Quellcodes das Vervielfältigungsstück den enthaltenen Copyrightvermerk beibehalten muss, der auf den Ersteller der Software hinweist. In

---

<sup>55</sup> Diese besonderen Klauseln werden hier nur für ausgewählte Lizenzen dargestellt.

<sup>56</sup> Vgl. Ziffer 1 BSD License, Ziffer 1 Apache Software License, Ziffer 3 OpenLDAP Public License.

den Ziffern 1 der BSD License und der Apache Software License ist zudem bestimmt, dass auf einen Haftungsausschluss des Erstellers der Software hingewiesen werden muss. Dies ist bei der OpenLDAP Public License nicht explizit gefordert, ergibt sich aber mittelbar daraus, dass eine wörtliche Kopie der Lizenz, die dann auch den Haftungsausschluss enthält, beigefügt werden muss.

**Verbreitung im Objektcode.** Wenn die Software im Objektcode verbreitet wird, dann ist wie bei der Sourcecodeweitergabe vorgesehen, dass dem Erwerber der Software eine Kopie des Lizenztextes verschafft werden muss, dass ein Copyrightvermerk erfolgt und dass auf den Haftungsausschluss des Erstellers der Software hingewiesen wird. Besonderheiten bestehen im Hinblick auf die Art und Weise in der diese Pflichten umzusetzen sind: Lizenztext, Haftungsausschluss und Copyrightvermerk müssen in der begleitenden Dokumentation oder anderen Begleitmaterialien enthalten sein. Damit wird den Unterschieden in der Art der Verwendung von Quell- und Objektcode Rechnung getragen.

## ***(2) Pflichten bei der Bearbeitung***

Es gehört zum Selbstverständnis des Open Source Modells, dass jedermann die Bearbeitung der Software für eigene Zwecke ohne Einschränkungen möglich sein soll. Die reine interne Bearbeitung wird daher nicht an irgendwelche Pflichten geknüpft. Pflichten sollen vielmehr erst in dem Moment zu beachten sein, in dem die Bearbeitung nicht mehr allein zu eigenen Zwecken genutzt wird, sondern veröffentlicht oder weitergegeben wird.

Wann eine Veröffentlichung oder Verbreitung vorliegt, kann anhand des Öffentlichkeitsbegriffs im Sinne des § 15 Abs. 3 UrhG beurteilt werden, der trotz der grundsätzlich vorrangigen Auslegung der Lizenzen, die insoweit aber keine weiteren Hinweise enthalten, hilfsweise heranzuziehen ist. Öffentlich ist danach jede der genannten Handlungen, wenn sie für eine Mehrzahl von Personen bestimmt ist, es sei denn, dass der Kreis der Personen bestimmt abgegrenzt ist und diese Personen durch gegenseitige Beziehungen oder durch Beziehung zum Veranstalter persönlich untereinander verbunden sind.

Wo diese Grenzen zu ziehen sind, kann im Einzelfall schwer zu bestimmen sein. Es muss aber zum Beispiel davon ausgegangen werden, dass aufgrund der fehlenden persönlichen Verbundenheit der Angestellten in einem großen Unternehmen bereits eine Veröffentlichung bzw. Verbreitung vorliegen kann, wenn das bearbeitete Werk über eine Abteilung hinaus in dem Unternehmen angeboten oder zur Verfügung gestellt wird.<sup>57</sup> Bei einem Test oder einer Modifikation der Software innerhalb eines überschaubaren Entwicklungsteams entstehen hingegen noch keine vertraglichen Pflichten.

### **(3) Pflichten beim Vertrieb bearbeiteter Versionen**

Anders als bei den Lizenzen mit einem strengen Copyleft-Effekt sehen BSDartige Lizenzen für den Fall des Vertriebs bearbeiteter Versionen keine Pflichten zur Offenlegung des modifizierten Quellcodes und zur Open-Source-Weiterlizenzierung vor. Der Bearbeiter soll vollständig selbst entscheiden können, in welcher Form und mit welcher Lizenz er seine Bearbeitung vertreiben möchte. Er kann entscheiden, in welchem Umfang er den Quellcode offen legt und mit welchem Inhalt er Dritten Nutzungsrechte an seinem eigenen (Bearbeiter-) Urheberrecht einräumt. Er kann die bearbeitete Software insbesondere auch „proprietär“ vertreiben.

Allerdings muss der Bearbeiter, da die BSDartigen-Lizenzen ausdrücklich auch für den Vertrieb in „modifizierter Form“ gelten, darauf hinweisen, dass die Software auf Code der jeweiligen Lizenz basiert. Zugleich sind insoweit die entsprechenden Verpflichtungen einzuhalten, die auch beim Vertrieb unveränderter Versionen gelten: Copyrightvermerk und Lizenz der „Basissoftware“ sowie ein Haftungsausschluss müssen beigelegt werden.

Schließlich sehen die meisten BSDartigen Lizenzen teilweise sehr unterschiedliche Regelungen vor, die den Umgang mit den Namen der Originalurheber beim Vertrieb veränderter Programmversionen regeln. Diese werden im Folgenden dargestellt.

### **(4) Besonderheiten bei den BSD-Lizenzen**

---

<sup>57</sup> Dies ist streitig, vgl. dazu vertiefend *Jaeger/Metzger*, *Open Source Software*, S. 39 f.; a.A. *Free Software Foundation*, FAQ, <<http://www.gnu.org/licenses/gpl-faq.html#InternalDistribution>>.

BSD-Lizenzen werden in unterschiedlichen Versionen verwendet, die sich insbesondere hinsichtlich der folgenden Klauseln unterscheiden können:

**Werbeklausel.** Die ursprüngliche Version<sup>58</sup> der BSD-Lizenz enthielt in Ziffer 3 eine besondere „Werbeklausel“: „All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by the University of California, Lawrence Berkeley Laboratory.“

Diese Klausel wurde in der Folge auch von anderen Entwicklern unter Abänderung des Referenznamens eingesetzt, so dass teilweise die Werbung für bestimmte Zusammenstellungen von Programmen (z.B. Betriebssysteme) schwierig wurde, da mehrere Dutzend Hinweise auf enthaltene Software erfolgen mussten. Wegen der Kritik an der damit verbundenen schwierigen praktischen Handhabbarkeit, wurde in neueren Lizenzversionen<sup>59</sup> die Werbeklausel gestrichen. Allerdings ist stets zu prüfen, ob nicht möglicherweise eine Software verwendet wird, die unter einer älteren Lizenzversion steht. Ob dies der Fall ist, ergibt sich jeweils aus den beigefügten Lizenztexten.

**Schutzklauseln.** Eine Reihe von BSD-Lizenzversionen enthalten eine Klausel, die ausdrücklich die Nennung des Originalautors „zur Förderung des Vertriebs von Bearbeitungen“ des Computerprogramms untersagt, solange nicht eine ausdrückliche schriftliche Erlaubnis erteilt wird.<sup>60</sup> Zweck dieser Klausel ist es, sicherzustellen, dass nicht der Originalautor durch das Werben für qualitativ schlechte Änderungen der Software durch Dritte in ein falsches Licht gerückt wird.

### ***(5) Besondere Pflichten bei den Apache-Lizenzen***

**Werbeklauseln.** In älteren Versionen (1.0) enthielt die Apache-Lizenz eine Werbeklausel, die derjenigen der früheren BSD-Lizenzversionen ähnelt. Da sich die Apache Software Foundation die Urheberrechte Dritter, die Beiträge zu Apache geleistet haben,

---

<sup>58</sup> BSD License (Original), <<http://www.xfree86.org/3.3.6/COPYRIGHT2.html#6>>.

<sup>59</sup> FreeBSD Copyright, <<http://www.de.freebsd.org/copyright/freebsd-license.html>>.

<sup>60</sup> „Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.“

übertragen lässt,<sup>61</sup> sind die aktuellen Apache-Versionen stets unter neueren Lizenzversionen ohne Werbeklausel erhältlich.

**Schutzklauseln.** Wie die BSD-Lizenzen enthält auch die Apache-Lizenz bestimmte Schutzklauseln. Anders als beim BSD-Copyright dienen diese aber nicht in erster Linie dem Schutz des Urhebers oder Copyrightinhabers sondern dem Schutz der Bezeichnung „Apache“ im geschäftlichen Verkehr. So sieht die Lizenz vor, dass die Namen „Apache“ und „Apache Software Foundation“ nicht verwendet werden dürfen, um den Vertrieb von veränderten Versionen der Software zu bewerben oder zu fördern, solange nicht eine ausdrückliche schriftliche Einwilligung der Apache Software Foundation erteilt wurde. Weiter wird bestimmt, dass veränderte Versionen von Softwareprodukten, die unter der Apache Lizenz stehen, nicht die Bezeichnung „Apache“ tragen dürfen oder einen Namen, der die Bezeichnung „Apache“ enthält, ohne dass die Apache Software Foundation vorher ausdrücklich und schriftlich eine entsprechende Einwilligung erteilt hat.

### c) Kombination mit Softwarebestandteilen unter anderen Lizenzen

Bearbeitete Versionen einer Software, die unter einer BSDartigen Lizenz steht, können vom Bearbeiter ohne Beschränkungen unter den von ihm ausgewählten Bedingungen lizenziert werden; es sind lediglich gewisse untergeordnete Modalitäten zu berücksichtigen. Für die Zusammenfügung von Code, der unter BSDartigen Lizenzen steht, mit Code, der unter anderen (proprietären oder freien) Softwarelizenzen steht, gilt daher:

Stellt sich das Ergebnis als ein einziges Computerprogramm dar,<sup>62</sup> handelt es sich um eine Bearbeitung der BSDartigen Software. Dieses Ergebnis, also die Bearbeitung, darf – wie soeben ausgeführt – nach den Bestimmungen der BSD-Lizenz nach Belieben lizenziert werden; es sind lediglich die Hinweise auf die „Basissoftware“ zu beachten.<sup>63</sup> Gleichzeitig ist das Schaffensergebnis jedoch auch regelmäßig eine Bearbeitung des (unter anderen Lizenzen stehenden) anderen Codes. Insoweit gilt, dass bei einer Ver-

---

<sup>61</sup> Vgl. <<http://www.apache.org/licenses/cla-corporate.txt>>.

<sup>62</sup> Umfangreichere Ausführungen zur Abgrenzung zwischen einem einzelnen Programm und mehreren Programmen befinden sich z.B. unten S. 32 ff. bei den Ausführungen zur GPL, da die Abgrenzung bei den Lizenzen mit einem strengen Copyleft zu größeren Schwierigkeiten führt.

<sup>63</sup> Siehe dazu oben S. 26.

bindung auch dessen Lizenzbedingungen zu berücksichtigen sind. Soll die Software daher beispielsweise mit proprietärer Software eines Drittanbieters verbunden werden, so stellt sich die Frage, ob von Seiten des Drittanbieters alle Rechte, die hierzu erforderlich sind, eingeholt wurden. Soll die Software mit anderer Freier Software verbunden werden, stellt sich die Frage, ob auch deren jeweilige Softwarelizenz eine Verbindung zulässt und welche Bedingungen sie dafür statuiert.

Stellt sich das Ergebnis als zwei Programme dar, indem die Kombination zwischen den jeweiligen Codebestandteilen nicht derart eng ist, dass bei technisch-wirtschaftlicher Gesamtbetrachtung lediglich ein Programm vorliegt, so kann das proprietäre oder unter einer anderen freien Lizenz stehende Programm selbstverständlich auch weiterhin unter seinen jeweiligen Lizenzen vertrieben werden. Für die Software, die unter einer BSDartigen Lizenz steht, sind hingegen deren Bedingungen weiterhin zu berücksichtigen.

## 2. Die GNU General Public License (GPL)

Die GNU General Public License (GPL)<sup>64</sup> ist mit Abstand die wichtigste und verbreitetste Lizenz für Freie Software. Sie ist zugleich der „Prototyp“ der Lizenzen mit einem strengen Copyleft-Effekt.<sup>65</sup> Der GPL sind unter anderem der Linux-Kernel und die GNU-Bestandteile (Emacs, GNU Compiler etc.) unterstellt. Derzeit liegt die GPL in der Version 2.0 vom Juni 1999 vor.

### a) Rechte

Der Lizenznehmer erhält die Rechte, die Software zu verändern, zu vervielfältigen und zu vertreiben. Dabei erfassen die Vertriebsrechte sowohl die Weitergabe der Software auf Datenträgern, als auch die Bereitstellung der Software zum Download im Internet.<sup>66</sup>

---

<sup>64</sup> GNU General Public License (GPL) (v. 2.0), <<http://www.fsf.org/copyleft/gpl.html>>.

<sup>65</sup> Weitere Lizenzen, die ein strenges Copyleft enthalten, sind u.a. die Affero Public License, die Open Software License, die Apple Public Source License, die Common Public License und die IBM Public License. Eine ausführliche Liste der Lizenzen, geordnet nach Lizenztypen, findet sich unter <[http://www.ifross.de/ifross\\_html/lizenzcenter.html](http://www.ifross.de/ifross_html/lizenzcenter.html)>.

<sup>66</sup> Vgl. dazu bereits ausführlich oben S. 21 ff.

## b) Pflichten

Wie alle Open Source Lizenzen gewährt die GPL nicht nur Rechte, sondern verbindet die Rechtseinräumung auch mit Pflichten. Diese Pflichten unterscheiden sich danach, ob unveränderte oder veränderte Versionen der Software weitergegeben werden sollen, und danach, ob ein Vertrieb im Sourcecode oder im Objektcode erfolgt. Die rein interne Bearbeitung der Software ohne Weitergabe ist nicht an bestimmte Pflichten geknüpft.<sup>67</sup>

### ***(1) Pflichten beim Vertrieb unveränderter Versionen***

**Sourcecode.** Beim Vertrieb unveränderter Versionen muss der Lizenznehmer einen Copyright-Hinweis (copyright notice) und einen Gewährleistungs- und Haftungsausschluss<sup>68</sup> anbringen,<sup>69</sup> bereits existierende Hinweise auf die Lizenz und auf das Fehlen von Haftung bestehen lassen und eine Kopie des Lizenztextes mitgeben.

Die Hinweise werden dabei in der Praxis regelmäßig in die einzelnen Source-Files eingefügt. Die Weitergabe der Kopien des Lizenztextes geschieht in den Fällen der schlichten Sourcecodeweitergabe zumeist in den „COPYING“-Dateien, die den jeweiligen Softwarebestandteilen beigelegt sind. Die Beilegung des Lizenztextes ist von wesentlicher Bedeutung, da ansonsten Kunden und Dritte nicht von ihren Rechten Kenntnis nehmen können. Es ist darauf zu achten, dass immer eine klare Zuordnung getroffen werden kann, welcher Lizenz ein Softwarebestandteil unterstellt ist.

**Objektcode.** Bei der Weitergabe der Software im Objektcode sind zunächst sämtliche Pflichten zu berücksichtigen, die auch für die Weitergabe im Sourcecode gelten. Die Mitlieferung der Lizenz erfolgt dabei regelmäßig in einer eigenen Datei oder aber in Papierform. Zusätzlich ist bei der Weitergabe im Objektcode zu beachten, dass die Software auch im Quellcode bereitgehalten werden muss, um die Möglichkeit zu schaffen, dass die Software auch tatsächlich verändert werden kann. Hierfür sieht die GPL insgesamt vier alternative Möglichkeiten vor:

---

<sup>67</sup> Vgl. zur Abgrenzung zwischen (interner) Bearbeitung und Vertrieb bereits oben S. 25 f.

<sup>68</sup> Die GPL bezieht die Verpflichtung auf den Vermerk zwar nur auf die „Warranty“, also Gewährleistung, und nicht auf die Liability, Haftung. Allerdings bezieht sich die Überschrift „NO WARRANTY“ auf die Ziffern 11 (Warranty) und 12 (Liability) der GPL, so dass auf beide Ausschlüsse hingewiesen werden sollte.

<sup>69</sup> Die Pflicht zur Anbringung dieses Hinweises gilt unabhängig davon, dass der in der GPL enthaltene vollständige Haftungs- und Gewährleistungsausschluss nach deutschem Recht unwirksam ist.

- Mitlieferung des vollständigen Quelltextes auf einem üblichen Datenträger (z.B. CD-ROM);
- schriftliches, zumindest drei Jahre gültiges Angebot an jedermann zur Lieferung des vollständigen Quelltextes auf einem üblichen Datenträger zu einem Preis, der die eigenen Kosten dafür nicht übersteigen darf;
- für den Fall, dass die Weitergabe nichtkommerziell erfolgt und das Programm auch schon ohne Quelltext und nur mit einem schriftlichen Angebot erworben wurde, reicht es aus, wenn dieses Angebot an die Erwerber weitergegeben wird (so dass sich diese an den ursprünglichen Anbieter wenden können);
- für den Fall, dass das Programm über das Internet vertrieben wird, genügt es, wenn auch der Quelltext auf derselben Website zum Download angeboten wird.

## **(2) (Zusätzliche) Pflichten beim Vertrieb veränderter Versionen**

Wird GPL-Software modifiziert, entstehen zusätzlich zu den bereits genannten Pflichten, die auch bei der Nutzung unveränderter Programmversionen bestehen, weitere Pflichten, wenn die modifizierte Software weitergegeben werden soll.

So müssen veränderte Versionen einen auffälligen Vermerk mit Hinweis auf die Modifikation und das Datum der Modifikation enthalten. Diese Kennzeichnung ist auch in anonymer oder pseudonymer Form möglich. In der Praxis geschieht sie durch entsprechende Eintragungen in den Header. Es genügt, wenn der Änderungsvermerk im Sourcecode enthalten ist, da jeder, der die Software im Objectcode weitergibt, zumindest verpflichtet ist, den Sourcecode anzubieten. So ist sichergestellt, dass der Empfänger die Möglichkeit hat, von dem Änderungsvermerk Kenntnis zu nehmen.

Weiter ist das sogenannte Copyleft zu berücksichtigen: Wer die Software oder einen Teil davon verändert und das so veränderte Programm weitergibt oder veröffentlicht, muss die Software gem. Ziffer 2b) GPL insgesamt unter der GPL lizenzieren.<sup>70</sup> Dabei müssen jedermann lizenzgebührenfrei dieselben Nutzungsrechte eingeräumt werden, die auch der Bearbeiter von dem oder den ursprünglichen Urhebern unter der GPL erhalten hat.

---

<sup>70</sup> Dazu ausführlich unten c).

Das Prinzip ist so einfach wie klar: Zusätzliche oder andere Lizenzbedingungen sind nicht zulässig, die Weitergabe darf nur unter der GPL erfolgen.

In etwas umständlicher Ausdrucksweise fordert die GPL schließlich in Ziffer 2c), dass besonderen Hinweispflichten genügt werden muss, wenn ein Programm bei der Ausführung „interaktiv Kommandos einliest“. Die nach Ziffer 2c) GPL erforderlichen Angaben umfassen Urhebervermerke, den Hinweis auf den Gewährleistungsausschluss, sofern nicht Gewährleistung explizit eingeräumt wird, den Hinweis darauf, dass das Programm unter den Bedingungen der GPL weiterverbreitet werden darf, und den Ort, an dem sich der Text der GPL findet, also etwa der Dateiname. Diese Informationen müssen beim Start des Programms eingeblendet werden. Ausnahmen gelten dort, wo die Ursprungsversion entsprechende Hinweise nicht enthielt. In diesen Fällen müssen entsprechende Hinweise auch nicht in der Bearbeitung erfolgen.

### **c) Kombination mit Softwarebestandteilen unter anderen Lizenzen**

Die Basisregelung für die Nutzung von GPL-Programmen mit Erweiterungen, Änderungen und hinzugefügten Softwaremodulen findet sich in Ziffer 2 b) GPL, die die folgende Verpflichtung enthält, die auch als „Copyleft“ bezeichnet wird:

*“You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.”*

Für die Kombination mit Software unter anderen Lizenzen gilt die gleiche Regel wie für die Hinzufügung neuer Codebestandteile: Eine Weitergabe muss insgesamt unter den Bedingungen der GPL erfolgen, wenn das Schaffensergebnis Bestandteile eines GPL-Programms enthält oder von diesem „abgeleitet“ („*derived*“) ist. Handelt es sich demgegenüber um mehrere unabhängige Programme, muss zwar das GPL-Programm nach wie vor unter der GPL vertrieben werden, die damit kombinierten Programme können jedoch unter ihren eigenen (proprietären oder freien) Lizenzen verbreitet werden.

Das entscheidende Kriterium, ob die Kombination von Softwarebestandteilen insgesamt unter der GPL zu vertreiben ist, liegt damit in dem Begriff „*derivative work*“. Wenn

hinzugefügte Softwarebestandteile gemeinsam mit dem GPL-Code kein solches „*derivative work*“ darstellen, können erstere auch unter anderen Lizenzen weitergegeben werden, ohne gegen die Lizenzbestimmungen der GPL zu verstoßen.

Wann ein „*derivative work*“ im Sinne der GPL vorliegt ist, ist im Wesentlichen eine Frage der Auslegung. Ausgangspunkt ist dabei Ziffer 2 GPL, 2. Hauptabsatz, in dem als Kriterium angeführt wird, dass selbständige Softwaremodule nicht unter der GPL lizenziert werden müssen, wenn sie als *eigenständige Werke* weitergegeben werden: *“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.**”* Danach genügt es also nicht, dass die Softwarebestandteile eigenständig sind, auch deren Verbreitung muss „als eigenständige Werke“ erfolgen.

Gleichzeitig regelt jedoch Ziffer 2, letzter Hauptabsatz: *“[...] mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.”* Allein auf eine Verbreitung auf einem oder mehreren Medien kommt es daher bei der Unterscheidung nicht an.

Vor diesem Hintergrund ergibt die Unterscheidung in Ziffer 2, Abs. 2 der GPL dann einen Sinn, wenn sie dahingehend ausgelegt wird, dass vermieden werden soll, dass GPL-Software mit selbständigen Softwaremodulen in einer Form verbunden wird, die es dem Nutzer unmöglich macht, die GPL-Bestandteile ohne weiteres als eigenständige Teile zu erkennen und zu nutzen. Wann dies der Fall ist, ist von einer Gesamtbetrachtung nach inhaltlichen und formalen Kriterien abhängig.<sup>71</sup> Im Einzelfall kann dies eine schwierige Auslegungsfrage bedeuten, die nicht immer zweifelsfrei möglich ist.

Die technische Ausgestaltung ermöglicht dabei – da der Programmierer zur Umsetzung der softwaretechnischen Aufgaben hier vielfach verschiedenste Möglichkeiten hat – nur in Grenzfällen bereits eine eindeutige Zuordnung, im übrigen ist sie allenfalls indiziell.

---

<sup>71</sup> Vgl. dazu *Jaeger/Metzger*, Open Source Software, S. 42.

So ist es anerkannt, dass Systemaufrufe von Anwendungsprogrammen an den Linux-Kernel nicht dazu führen, dass das Anwendungsprogramm ein „*derivative work*“ des Kernels wird.<sup>72</sup> Auch werden Pipes, Sockets und Kommandozeilenargumente als übliche Kommunikationsmittel zwischen selbständigen Softwarebestandteilen angesehen,<sup>73</sup> während bei Änderungen des GPL-Sourcecodes oder dem Vertrieb in einem Executable stets ein „*derivative work*“ vorliegt und demzufolge dann die gesamte Software nur unter der GPL weitergegeben werden darf.

Im Übrigen ist es grundsätzlich eine Frage des Einzelfalls, ob und wann von einem oder mehreren Programmen auszugehen ist. Im Folgenden werden daher lediglich einzelne generalisierende und beispielhafte Betrachtungen angestellt.

### **(1) Kernelmodule**

Eine in der Praxis sehr umstrittene Frage besteht darin, wann ein Kernelmodul als „abgeleitet“ betrachtet werden kann. Hier gibt es keine einheitliche Antwort. Sofern die Kernelfunktionen nur insoweit genutzt werden, dass die eigenen Softwaremodule mit dem Kernel über eine Schnittstelle „kommunizieren“, kann ein unabhängiges Softwaremodul vorliegen. Entscheidend ist dafür im Regelfall die Frage, ob das Kernelmodul funktional eigenständig ist. Dafür spielen vielfache Aspekte eine Rolle. Wird ein Gerätetreiber eigens für Linux entwickelt, soll er also dessen integraler Bestandteil werden, sprechen gute Gründe dafür, von einem „*derivative work*“ auszugehen.<sup>74</sup> Entsprechendes gilt, wenn das Kernelmodul technisch weitergehend in den Kernel eingebunden ist, etwa mittels „Hooks“, oder dessen „Infrastruktur“ mehr als nur oberflächlich nutzt, so dass auf diese Weise offenbar wird, dass das Kernelmodul nicht unabhängig ist. Es existieren allerdings auch Kernelmodule, die älter sind als Linux. Dort liegt es auf der Hand, dass sie als funktional eigenständig anzusehen sind, da sie nicht nur gemeinsam mit Linux sondern auch mit anderen Betriebssystemen vertrieben werden können. Entspre-

---

72 Dies wird in der Kernellizenz durch einen Zusatz von *Linus Torvalds* zur GPL auch explizit hervorgehoben: „*NOTE! This copyright does \*not\* cover user programs that use kernel services by normal system calls - this is merely considered normal use of the kernel, and does \*not\* fall under the heading of "derived work". Also note that the GPL below is copyrighted by the Free Software Foundation, but the instance of code that it refers to (the Linux kernel) is copyrighted by me and others who actually wrote it.*“ Diesem Hinweis dürfte nur deklaratorische Bedeutung zukommen.

73 So auch die FAQ der FSF, <<http://www.fsf.org/licenses/gpl-faq.html#MereAggregation>>.

74 In der Praxis wird es von den Kernelentwicklern vielfach geduldet, dass reine Binärkernelmodule in Linux eingefügt werden, um dem Betriebssystem damit weitere Hardware zu erschließen. Dies heißt keinesfalls, dass dieses Vorgehen auch rechtlich zulässig ist oder eine Selbstbindung nach sich zieht.

chendes gilt für Kernelmodule, die zwar nicht älter sind als Linux, aber auch mit anderen Unix-Versionen ablauffähig sind und auf diese Weise ihre Eigenständigkeit zeigen. Es ist zulässig, solche Kernelmodule unter einer proprietären Lizenz zu verbreiten, wenn sie auch formal klar vom Linux-Kernel getrennt werden.

## **(2) Plugins**

Wie für den Bereich der Kernelmodule kann auch für Plugins keine einheitliche Aussage getroffen werden. Ein Plugin ist dann nicht als „*derivative work*“ anzusehen, das nur unter der GPL verbreitet werden darf, wenn es zum einen eine selbständige Funktionalität enthält und technisch von dem Hauptprogramm klar getrennt ist. Dies ist für Einbindung mittels „*fork and exec*“ allgemein anerkannt.<sup>75</sup> Für andere Techniken, etwa Einbindung als Bibliothek oder über eine spezielle Schnittstelle kann dies nicht allgemeingültig festgestellt werden.

Gerade bei Plugins kommt der inhaltlichen Bewertung des Verhältnisses zum „Hauptprogramm“ besondere Bedeutung zu, da der Begriff „Plugin“ ein breites Feld von Softwareimplementierungen umfasst. So kann die Schnittstelle für Plugins technisch dafür vorgesehen sein, durch die Einbindung von Plugins und die damit verbundene modulare Struktur eine bessere Performance oder eine einfachere Weiterentwicklung zu ermöglichen. In diesem Fall spricht mehr dafür, ein neues Plugin als „*derivative work*“ anzusehen. Hat die Schnittstelle für Plugins hingegen die Funktion, externe oder funktional vollständig eigenständige Bestandteile mit dem Hauptprogramm ablaufen zu lassen, ist davon auszugehen, dass der Copyleft-Effekt nicht eingreift.

## **(3) Bibliotheken**

Für die Beurteilung der Frage, ob Programme, die unter der GPL lizenzierte Programm-bibliotheken verwenden, ebenfalls nur unter der GPL weiterverbreitet werden dürfen, ist zum einen auf die Funktion der Bibliothek, zum anderen auf die Form der Verbreitung abzustellen.<sup>76</sup> Sofern Bibliothek und zugreifendes Programm in einem Executable vertrieben werden, liegt stets ein einheitliches Programm vor, das insgesamt unter die

---

75 Vgl. die Auffassung der FSF, die regelmäßig der „strengsten Auslegung“ folgt: <<http://www.fsf.org/licenses/gpl-faq.html#GPLPluginsInNF>>.

76 Dazu näher Jaeger/Metzger, Open Source Software, S. 43 ff.

GPL gestellt werden muss. Bei dem statischen Verlinken wird der Sourcecode der Bibliothek mit dem Sourcecode des zugreifenden Programms verbunden, so dass in jedem Fall ein abgeleitetes Werk entsteht.

Bei dynamischem Verlinken kommt es wesentlich auf inhaltliche Kriterien an. Sofern Programmbibliotheken verwendet werden, die eine strukturelle Einheit mit dem zugreifenden Programm bilden, liegt in der Regel ein „*derivative work*“ vor. Dies dürfte insbesondere für sog. Frameworks der Fall sein, wenn das zugreifende Programm für die Bibliotheken des Frameworks geschrieben wurde. Handelt es sich hingegen um Funktionsbibliotheken, die eingebunden werden, um einer Applikation eine zusätzliche Funktionalität zu verleihen, und dabei die Applikation nicht im Hinblick auf die Bibliothek geschrieben wurde, kann von eigenständigen Programmteilen ausgegangen werden. Dies gilt insbesondere dann, wenn die Bibliotheken austauschbar sind und rein zur Ersparnis von Programmieraufwand für eine Standardroutine dienen. Es muss aber darauf hingewiesen werden, dass diese Frage umstritten ist und daher ein rechtliches Restrisiko verbleibt. In jedem Falle ist es erforderlich, dass der Nutzer die Bibliothek formal klar abgegrenzt erhält – also nicht (nur) in einem Executable – und das zugreifende Programm auch mit einer weiterentwickelten Form der Bibliothek arbeiten kann.

#### ***(4) Lizenzkompatibilität: Kombination von GPL-Software mit Software unter anderen Lizenzen***

**BSDartige-Lizenzen.** Grundsätzlich kann Software unter BSDartigen Lizenzen problemlos mit GPL-Software kombiniert werden. Sofern dabei ein abgeleitetes Werk entsteht, muss dann die gesamte Software bei der Weitergabe der GPL unterstellt werden. Eine Ausnahme vom Grundsatz der problemlosen Kombinierbarkeit gilt dann, wenn die Software unter der originalen BSD-Lizenz oder Apache Software License v. 1.0 lizenziert wurde, die jeweils eine Werbeklausel enthalten.<sup>77</sup> Da die Werbeklausel auch bei der Verbreitung veränderter Programmversionen zu beachten ist, würde sich daraus die Verpflichtung ergeben, bei einer Verbindung eines GPL-Programms mit BSD-Software unter der originalen BSD-Lizenz, die gesamte Software der GPL zu unterstellen *und* einen Werbehinweis vorzunehmen. Nach Ansicht der Free Software Foundation sind die

---

<sup>77</sup> Dazu siehe oben S. 26 f.

originale BSD-Lizenz bzw. die Apache Software License 1.0 deshalb nicht GPL-kompatibel, d.h. eine solche Softwareverbindung verstoße gegen die GPL.<sup>78</sup> Dies wird nicht näher begründet, aber naheliegend ist der Bezug zu Ziffer 6 der GPL, wonach es dem Lizenznehmer nicht gestattet ist, bei der Weitergabe der GPL-Software zusätzliche Bedingungen aufzustellen.<sup>79</sup> Fasst man die Pflicht, einen Werbehinweis vorzunehmen, als eine solche zusätzliche Bedingung auf – was durchaus auch anders gesehen werden kann –, so gelangt man zur Unvereinbarkeit von GPL-Software mit Software unter der originalen BSD-Lizenz in einem „*derivative work*“. Da in der Praxis mittlerweile ganz überwiegend Lizenzen ohne Werbeklausel verwendet werden, sind entsprechende Problemkonstellationen in der Praxis allerdings äußerst selten.

**Copyleft-Lizenzen.** Probleme mit der Kombination von Softwarebestandteilen unter anderen Lizenzen können grundsätzlich dort auftreten, wo diese anderen Softwarebestandteile unter einer Copyleft-Lizenz stehen. In diesem Fall stehen sich zwei Lizenzen gegenüber, die versuchen, die Bearbeitung der Software unter ihr ausschließliches Lizenzregime zu ziehen und gleichzeitig die Vereinbarung weiterer Lizenzbedingungen zu verbieten. Einige Lizenzen sehen hier Klauseln vor, die eine Nutzung der Kombination von Softwarebestandteilen unter der GPL gestatten; die andere Lizenz tritt in diesem Fall aufgrund ausdrücklicher Regelung zurück, um eine Kompatibilität der Softwarebestandteile zu ermöglichen. So regelt Ziffer 3 LGPL: *“You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library.”* Und in der Deutschen Freien Software Lizenz heißt es in Ziffer 3 Abs. 3: *„Wenn Sie das Programm oder einen Teil hiervon - verändert oder unverändert - zusammen mit einem anderen Programm verbreiten oder öffentlich zugänglich machen, das unter der GNU General Public License (GPL) lizenziert wird, darf das Programm auch unter den Bedingungen der GPL genutzt werden, sofern es mit dem anderen Programm ein "derivative work" im Sinne der GPL bildet. Dabei sollen die Hinweise auf diese Lizenz entfernt und durch einen Hinweis auf die GPL ersetzt werden.“* Wo andere Copyleft-Lizenzen solche Klauseln jedoch nicht enthalten, ist eine Kombinierbarkeit mit GPL-Software regelmäßig ausgeschlossen, wenn dadurch ein „*derivative work*“ entstehen würde. Dies gilt namentlich für die Mozilla Public License (MPL).

---

<sup>78</sup> Vgl. dazu <<http://www.fsf.org/licenses/license-list.html#OriginalBSD>>.

<sup>79</sup> *“You may not impose any further restrictions on the recipients' exercise of the rights granted herein.”*

### 3. Die GNU Lesser General Public License (LGPL)

Die Lesser General Public License (LGPL)<sup>80</sup> ist eine Lizenz, die speziell für die Lizenzierung von Programmbibliotheken entwickelt wurde.<sup>81</sup> Sie gehört zur Familie der Lizenzen mit beschränktem Copyleft-Effekt. Mit der Lizenz sollte die Möglichkeit geschaffen werden, eine weitergehende Einbindung von freien Bibliotheken in proprietäre Programme zu ermöglichen. Derzeit liegt die Lizenz in der Version 2.1 aus dem Februar 1999 vor. Bedeutsamstes Programm unter der LGPL ist die „glibc“.

#### a) Rechte

Die durch die LGPL gewährten Rechte stimmen mit den aus der GPL gewährten Rechten überein: Der Lizenznehmer erhält die Befugnis, die Software zu verändern, zu vervielfältigen und sowohl körperlich als auch unkörperlich zu vertreiben.

Damit LGPL-Software auch unproblematisch mit GPL-Programmen verbunden werden kann, sieht Ziffer 3 der LGPL vor, dass die Lizenzierung zugleich unter der GPL und LGPL erfolgt. Der Lizenznehmer hat die Wahl, ob er die Bibliothek unter den Bedingungen der GPL oder der LGPL nutzen möchte.

#### b) Pflichten

Bei der unveränderten Weitergabe eines LGPL-Programms sind dieselben Pflichten zu beachten wie bei der Verbreitung von GPL-Software, also im Wesentlichen:

- Anbringung eines Copyrighthinweises und eines Gewährleistungs- und Haftungsausschlusses;
- Nichtentfernung bestehender Hinweise auf die Lizenz und auf das Fehlen von Haftung und Gewährleistung;
- Mitlieferung einer Kopie des Lizenztextes;
- Zugänglichmachung des Quellcodes.<sup>82</sup>

---

<sup>80</sup> GNU Lesser General Public License (LGPL) (v. 2.1), <<http://www.fsf.org/copyleft/lesser.html>>.

<sup>81</sup> Dies ergibt sich unter anderem aus der Präambel der LGPL. Frühere Versionen der LGPL wurden als „Library GPL“ bezeichnet.

<sup>82</sup> Vgl. im Einzelnen oben S. 30 f.

Weitestgehend die gleichen Pflichten wie bei der GPL gelten auch beim Vertrieb veränderter Versionen, wenn die der LGPL unterstellte Bibliothek modifiziert wurde, insbesondere ist die modifizierte Version beim Vertrieb unter der LGPL zu lizenzieren und der Quellcode zugänglich zu machen.<sup>83</sup> Hier greift der Copyleft-Effekt „streng“ ein. Abweichend ist die rechtliche Lage bei der LGPL aber hinsichtlich des Verhältnisses zwischen zugreifendem Programm und LGPL-Bibliothek, wie im Folgenden beschrieben.

### c) Kombination mit Softwarebestandteilen unter anderen Lizenzen

Bei der Kombination von LGPL-Bibliotheken mit Programmen, die auf diese Bibliothek zugreifen, dürfen diese „zugreifenden Programme“ unter einer beliebigen Lizenz mit der Bibliothek verbreitet werden, wenn einige besondere „LGPL-Pflichten“ beachtet werden. Systematisch geht die LGPL dabei in Ziffer 5 davon aus, dass dann, wenn das zugreifende Programm mit der LGPL-Bibliothek zu einem Executable verbunden wird, stets ein „*derivative work*“ entsteht. Abweichend von der Grundregel, dass alle „*derivative works*“ wiederum der LGPL unterstellt werden müssen, sehen die Ziffern 6 und 7 der LGPL sodann Sonderregelungen vor.

#### **(1) Pflichten aus dem beschränkten Copyleft der Ziffer 6 LGPL**

In Ziffer 6 wird die Erlaubnis, das zugreifende Programm unter beliebigen Lizenzbestimmungen verbreiten zu dürfen, an eine Reihe von Verpflichtungen geknüpft. Zunächst muss dem Kunden, dem die Software geliefert wird, die Veränderung des zugreifenden Programms gestattet werden und zu diesem Zweck auch ein Reverse Engineering zur Fehlerbehebung. Zudem muss jedes Vervielfältigungsstück (Datenträger mit der Software) einen deutlichen Hinweis darauf enthalten, dass eine LGPL-Bibliothek verwendet wird und dass diese Bibliothek unter den Lizenzbedingungen der LGPL genutzt werden darf. Der Lizenztext der LGPL muss mitgeliefert werden. Zeigt das zugreifende Programm auf einem Display einen Urhebervermerk an, dann muss dabei auch ein Urhebervermerk für die LGPL-Bibliothek eingeblendet werden sowie ein Hinweis darauf, wo der Lizenztext der LGPL zu finden ist.

---

<sup>83</sup> Der einzige Unterschied besteht darin, dass keine der Ziffer 2c der GPL entsprechende Regelung – Hinweispflichten beim interaktiven Einlesen von Kommandos – in der LGPL enthalten ist. Dies ist wohl der Eigenart von Programmbibliotheken geschuldet.

Neben den genannten Pflichten muss zugleich *eine* der folgenden Bedingungen erfüllt werden, deren Ziel es ist, sicherzustellen, dass Programmierer die LGPL-Bibliothek an ihre Bedürfnisse anpassen können und dennoch die weitere Verwendung mit dem zugreifenden Programm möglich bleibt:

- Mitlieferung des Sourcecodes der Bibliothek sowie des Objectcodes *oder* Sourcecodes des zugreifenden Programms in einer Weise, die es dem Kunden erlaubt, die Bibliothek zu verändern und mit dem zugreifenden Werk neu zu verlinken, so dass ein neues Executable erstellt werden kann.
- Abgabe eines 3 Jahre gültigen Angebots zur Lieferung der in der ersten Alternative aufgeführten Materialien gegen ein Entgelt, das nicht höher sein darf als die Kosten für die Herstellung der gelieferten Materialien. Sofern die Software über das Internet vertrieben wurde, dürfen auch die in der ersten Alternative aufgeführten Materialien über das Internet auf der gleichen Website zugänglich gemacht werden.
- Verwendung eines geeigneten „shared-library-Mechanismus“ zum Linken mit der Bibliothek. „Geeignet“ ist ein solcher Mechanismus, der erstens während der Laufzeit eine im Computersystem des Benutzers bereits vorhandene Kopie der Bibliothek benutzt, anstatt Bibliotheksfunktionen in das ausführbare Programm zu kopieren, und der zweitens auch mit einer veränderten Version der Bibliothek, wenn der Benutzer eine solche installiert, richtig funktioniert, solange die veränderte Version schnittstellenkompatibel mit der Version ist, mit der das zugreifende Programm erstellt wurde

## **(2) Pflichten aus dem beschränkten Copyleft der Ziffer 7 LGPL**

Ziffer 7 LGPL gestattet, dass in einer Bibliothek LGPL-Funktionseinheiten und proprietäre Funktionseinheiten in einer gemeinsamen Bibliothek kombiniert werden, ohne dass die proprietären Funktionseinheiten bei der Weitergabe unter die LGPL gestellt werden müssen. Dafür müssen die folgenden Voraussetzungen erfüllt sind:

- Die getrennte Verbreitung der LGPL-Bestandteile und der proprietären Bestandteile muss gestattet sein.

- Es muss ein deutlicher Hinweis darauf erfolgen, dass eine LGPL-Bibliothek verwendet wird und wo die LGPL-Bestandteile aufgefunden werden können.
- Es müssen die LGPL-Anteile zusätzlich in isolierter Form als Kopie unter den Lizenzbedingungen der LGPL beigelegt werden.

#### 4. Die Common Public License (CPL)

Die Common Public License (CPL) wurde von IBM entwickelt und ist die Nachfolgerin der „IBM Public License“. Die CPL ist eine Copyleft-Lizenz, soll aber ihrer Intention nach einfacher als die GPL die Kombination mit Softwarebestandteilen unter anderen Lizenzbedingungen gestatten.<sup>84</sup> Zu den Charakteristika der CPL gehören die Sonderregelungen für die Lizenzierung von Patenten und der Umstand, dass in einer Rechtswahlklausel der Ziffer 7 das Recht des US-Staates New York vorgesehen ist.

##### a) Rechte

Jeder „Contributor“, d.h. sowohl der ursprüngliche Entwickler als auch die Rechtsinhaber aller Weiterentwicklungen, räumt dem Lizenznehmer ein einfaches Nutzungsrecht zur Vervielfältigung, Verbreitung, öffentlichen Zugänglichmachung und Bearbeitung der Software ein. Dabei bedient sich die Lizenz der Wortwahl des § 106 US Copyright Act, um zu zeigen, dass alle möglichen Nutzungsarten erfasst werden sollen. Daher sind auch Nutzungen der öffentlichen Wiedergabe wie „publicly perform“ aufgeführt, denen bei der Softwarenutzung keine besondere Bedeutung zukommt.

Eine Besonderheit der CPL ist der Umstand, dass auch die Unterlizenzierung explizit gestattet wird. Damit ist es möglich, dass bei der Weitergabe keine Direktlizenzierung durch die Rechtsinhaber vorgenommen wird, sondern der Lizenznehmer selbst an Dritte weiterlizenziert. Dann ist er alleiniger Lizenzgeber gegenüber diesem Dritten und muss auch alleine für die Rechtseinräumung einstehen. Der Nachteil gegenüber der unter der CPL ebenfalls möglichen Direktlizenzierung besteht darin, dass bei der Weitergabe durch mehrere Unterlizenznehmer „Rechteketten“ entstehen. Ist eine Unterlizenzierung in der Kette unwirksam – etwa wegen eines Mangels beim Vertragsabschluss – so sind alle nachfolgenden Lizenzierungen ebenfalls gegenstandslos, weil

---

<sup>84</sup> <<http://www-106.ibm.com/developerworks/library/os-cplfaq.html>>.

dann keine Rechte zur weiteren Unterlizenzierung mehr existieren, die übertragen werden können.<sup>85</sup> Bei der Direktlizenzierung werden die Rechte hingegen immer von den Rechtsinhabern erworben, so dass sich eine unwirksame Lizenzierung nur in dem Verhältnis zu einem Lizenznehmer auswirkt.

Die Patentlizenz ermöglicht die Verwendung eines Patents, soweit dies erforderlich ist, um das Programm mit einem Entwicklungsbeitrag des Patentinhabers zu nutzen. Die Patentlizenz gilt jedoch nicht für den Fall, dass der Entwicklungsbeitrag des Patentinhabers mit anderen Softwarekombinationen als dem CPL-Programm genutzt werden soll. Damit soll verhindert werden, dass eine Patentlizenz „erschlichen“ wird, indem der Entwicklungsbeitrag in vollständig anderen Programmen eingesetzt wird, die nicht der CPL unterstehen.

## **b) Pflichten**

Die CPL unterscheidet hinsichtlich der Pflichten nach kommerziellem Vertrieb und nicht-kommerzieller Weitergabe.

### **(1) Pflichten beim nicht-kommerziellen Vertrieb**

Die CPL verbietet in Ziffer 3 die Änderung und Streichung von Urhebervermerken und verlangt zusätzlich, dass diejenigen, die einen Entwicklungsbeitrag leisten, sich als Urheber identifizieren. Anonyme Beiträge sind damit nicht gestattet. Grundsätzlich sieht der Schutz der Urheberpersönlichkeitsrechte in Deutschland vor, dass der Urheber darüber entscheiden kann, ob er mit Namen genannt werden möchte (§ 13 UrhG). Allerdings kann über dieses Recht in Grenzen verfügt werden.<sup>86</sup> Ob derjenige, der CPL-Programme weiterentwickelt und vertreibt, nach deutschem Urheberrecht die Verpflichtung eingehen kann, seinen Namen nennen zu müssen, ist zumindest fraglich. Da der Urheber allerdings erst im Zeitpunkt des eigenen Vertriebs aus der Anonymität treten muss und konkret darüber entscheidet, dass und in welchem Umfang er über sein

---

<sup>85</sup> Vgl. dazu oben S. 10f.

<sup>86</sup> Vgl. Metzger, Rechtsgeschäfte über das Droit moral im deutschen und französischen Urheberrecht, S. 43.

Namensnennungsrecht verfügt, lässt sich eine Vereinbarkeit mit deutschem Urheber- und AGB-Recht durchaus begründen.<sup>87</sup>

**Beim Vertrieb im Sourcecode** muss eine Kopie des Lizenztextes beigefügt werden, und entsprechend dem Copyleft müssen „Contributions“ unter der CPL lizenziert werden.

**Für den Vertrieb im Objektcode** sieht die CPL die Besonderheit vor, dass dieser auch unter einer eigenen Lizenz des Lizenznehmers weiterverbreitet werden darf, wenn die folgenden Voraussetzungen erfüllt sind:

- Die Lizenzbedingungen dürfen der CPL nicht widersprechen
- Die Gewährleistung wird wirksam für alle Rechtsinhaber ausgeschlossen
- Die Haftung wird wirksam für alle Rechtsinhaber ausgeschlossen
- Es muss ein Hinweis erfolgen, dass alle von der CPL abweichenden Lizenzbedingungen nur den Anbieter betreffen, nicht aber die übrigen Rechtsinhaber
- Der Sourcecode der Software muss von dem Anbieter in einer üblichen Form zugänglich gemacht werden (und dann auch unter der CPL lizenziert werden)

Diese komplizierten Voraussetzungen dürften diese Vertriebsform unter einer eigenen Lizenz eher als Ausnahme erscheinen lassen.

## **(2) Pflichten beim kommerziellen Vertrieb**

Neben den Pflichten, die auch den nicht-kommerziellen Vertrieb betreffen, sieht Ziffer 4 CPL zusätzlich eine besondere Freistellungsklausel vor. Danach muss derjenige, der ein Programm unter der CPL in einem kommerziellen Produkt vertreibt, die anderen Rechtsinhaber von Haftungsansprüchen Dritter freistellen, die aus dem kommerziellen Vertrieb resultieren. Ansprüche Dritter wegen der Verletzung von Rechten an geistigem Eigentum sind allerdings davon ausgenommen. Mit der Klausel soll sichergestellt werden, dass die Rechtsinhaber nicht deswegen haften, weil beim kommerziellen Vertrieb gewisse Zugeständnisse bei Haftung und Gewährleistung gegenüber Kunden gemacht

---

<sup>87</sup> Vgl. dazu allgemein Metzger, Rechtsgeschäfte über das Droit moral im deutschen und französischen Urheberrecht, S. 219 f. Durch die Rechtswahlklausel der CPL kann die Anwendbarkeit der zwingenden Regelungen des deutschen Urheberrechts und von Verbraucherschutzregelungen nicht umgangen werden.

werden. Die Klausel ist sehr auf das US-amerikanische Haftungssystem zugeschnitten und in Deutschland nur von beschränkter Bedeutung, da Haftungszugeständnisse nur im Rahmen der Vertragsbeziehung zwischen dem Anbieter und dem Endkunden rechtliche Wirkungen entfalten.

### **c) Kombination mit Softwarebestandteilen unter anderen Lizenzen**

Die CPL enthält einen strengen Copyleft-Effekt, der sich allerdings in seiner Ausgestaltung von der Regelung in der GPL unterscheidet. So stellt die CPL darauf ab, was als „Contribution“ anzusehen ist und damit dem Copyleft unterfällt. In der Definition der Ziffer 1 CPL wird dabei auf „changes to the Program“ und „additions to the Program“ abgestellt, zugleich der Begriff aber wie folgt eingeschränkt:

*„Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.“*

Der Copyleft-Effekt greift also dann nicht ein, wenn kumulativ zwei Bedingungen erfüllt sind. Es muss ein eigenständiges Softwaremodul vorliegen und dieses Softwaremodul darf kein von dem Ursprungsprogramm „abgeleitetes“ Werk darstellen.

#### **(1) Eigenständige Softwaremodule**

Mit dem Element der eigenständigen Softwaremodule dürfte eine Abgrenzung von solchen „additions“ bezweckt sein, die innerhalb eines Programms bzw. innerhalb einer Datei stattfinden. Sinnvollerweise ist dabei auf den Sourcecode abzustellen, da so die Eigenständigkeit für den Programmierer bewertet wird und nicht für den Rechner.

#### **(2) Kein abgeleitetes Werk**

Fraglich ist, was unter „abgeleitet“ („derivative“) zu verstehen ist. Anders als die GPL enthält die CPL keine weiteren Auslegungshilfen. In der englischen Rechtssprache wird mit „derivative work“ eine Bearbeitung eines Werkes bezeichnet, so dass im Anwendungsbereich des deutschen Urheberrechtsgesetzes der gesetzliche Begriff der Bearbeitung in § 69c Nr. 2 UrhG zu Grunde zu legen ist. Jedoch gibt es keine klar umrissene Definition dessen, was als „Bearbeitung“ im Bereich Software anzusehen ist, insbeson-

dere gibt es nur wenig Rechtsprechung zu dieser Frage. Daher ist auf allgemeine Grundsätze des Urheberrechts abzustellen, um zu umschreiben, was als Bearbeitung verstanden werden kann.

Zweck des Bearbeitungsrechts ist es, dem Urheber eines Werkes die Rechtsmacht darüber einzuräumen, ob sein Werk von Dritten verändert werden darf. Diese Rechtsmacht besteht nur insoweit, wie das eigene Werk von solchen Änderungen unmittelbar betroffen ist. Dies ist für Codeänderungen innerhalb eines Softwaremoduls zweifellos der Fall. Schwierig und in der Rechtswissenschaft noch nicht geklärt ist die Frage, ob und in welchen Fällen die Kombination von Softwaremodulen dazu führt, dass das eine Modul als Bearbeitung eines anderen Moduls anzusehen ist. Außerhalb des Softwareurheberrechts werden Bearbeitungen von „Werkverbindungen“ i.S.d. § 9 UrhG dadurch unterschieden, dass bei Werkverbindungen selbständig verwertbare Teile vorliegen, während bei der Bearbeitung der eine Teil vom anderen Teil abhängig und nicht selbständig verwertbar ist. Überträgt man dies auf das Softwareurheberrecht, könnte man danach unterscheiden, ob ein Softwaremodul ohne den der CPL unterstellten Programmbestandteil sinnvoll verwertbar ist und funktional von diesem abhängt. Im Einzelfall können sich damit schwierige Abgrenzungsfragen ergeben.

## 5. Die Mozilla Public License (MPL)

Die Mozilla Public License (MPL)<sup>88</sup> gehört zur Familie der Lizenzen mit einem beschränktem Copyleft-Effekt. Sie ist die Lizenz, unter der der Web-Browser „Mozilla“, verbreitet wird. Insgesamt weist die MPL gegenüber anderen Lizenzen eine deutlich höhere Regelungsdichte auf. Dies erklärt sich unter anderem daraus, dass die Lizenz entwickelt wurde, um ein ursprünglich proprietäres Produkt, den Netscape Browser, in Open Source Software zu überführen. Da der Netscape Browser auch Fremdsoftware enthielt, war eine Lizenzierung unter der GPL nicht möglich. Netscape ließ daher von Lizenzjuristen eine neue Open Source Lizenz entwickeln.

---

<sup>88</sup> Mozilla Public License (v. 1.1), <<http://www.mozilla.org/MPL/MPL-1.1.html>>.

## a) Rechte

Die höhere Regelungsdichte der MPL hat zur Folge, dass schon der Umfang der Rechtseinräumung durch die Rechtsinhaber sehr viel ausführlicher geregelt ist. Dabei wird unterschieden nach Rechtseinräumungen des „*Initial Developers*“ und des „*Contributors*“.

**Rechtseinräumung durch den Initial Developer.** Der Nutzer erhält das „weltweite, lizenzgebührenfreie einfache Nutzungsrecht, die Software zu benutzen, zu vervielfältigen, zu verändern, zugänglich zu machen, zu verbreiten und Unterlizenzen zu vergeben“ (vgl. Ziffer 2.1. (a) MPL). Soweit durch die Vervielfältigung und Verbreitung des Programms Patentrechte betroffen sind, so ist dem Nutzer auch die Benutzung des Patents gestattet (vgl. Ziffer 2.1. (b) MPL). Dies gilt gem. Ziffer 2.1. (d) MPL allerdings nur für diejenigen Patentansprüche, die durch die Nutzung des unveränderten Codes betroffen sind. Einer Verbreitung veränderter Versionen von Mozilla können deshalb durchaus patentrechtliche Verbotsrechte von Netscape entgegenstehen. Hinter dieser Klausel steht die folgende Überlegung: In dem Moment, in dem die Software freigegeben wird, hat der Rechtsinhaber den Überblick über seine eigenen Patentansprüche und kann allen Nutzern die Verwendung seiner Patente einräumen. Zugleich soll aber keine allgemeine Patentlizenz an jedermann eingeräumt werden, die es Dritten gestattet, Patente des Rechtsinhabers in anderem Zusammenhang als dem unveränderten Quellcode zu nutzen.

**Rechtseinräumungen durch den Contributor.** Die Rechtseinräumungen bei späteren Modifikationen sind in Ziffer 2.2 MPL geregelt. Sie entsprechen weitgehend denen des „*Initial Developers*“. Die MPL unterscheidet wohl deshalb zwischen „*Initial Developers*“ und „*Contributors*“, um innerhalb der Formulierungen herausstellen zu können, dass sowohl die ursprünglichen Urheberrechte als auch die Bearbeiterurheberrechte lizenziert werden.

## b) Pflichten

Im Hinblick auf die Verpflichtungen ähnelt die MPL den Lizenzen mit strengem Copyleft-Effekt. Deutliche Beschränkungen des Copyleft ergeben sich – wie weiter unten

dargestellt wird –, wenn es um den Vertrieb eines Gesamtprogramms (*Larger Work*) geht, das aus mehreren selbständigen Teilen besteht.

**Zugänglichmachung des Sourcecodes.** Greift das Copyleft, muss der Sourcecode zugänglich gemacht werden. Hierfür sieht die MPL in Ziffer 3.2. bestimmte Regelungen vor, die von denen der GPL abweichen. Der Bearbeiter hat die Möglichkeit, den Sourcecode gemeinsam mit dem Objektcode auf demselben Datenträger zu vertreiben. Er kann aber auch den Zugang zum Sourcecode sicherstellen, indem er diesen über einen „üblichen elektronischen Verbreitungsmechanismus“ (z.B. Website) zumindest für 12 Monate zur Verfügung stellt; bei Weiterentwicklung dieser Modifikationen reicht ein Zeitraum von 6 Monaten.

Beim Vertrieb der MPL-Software im Objektcode muss gemäß Ziffer 3.6. MPL ein deutlicher Hinweis darauf erfolgen, wo der Sourcecode zugänglich ist und dass dieser unter der Mozilla Public License steht. Dieser Hinweis muss dabei auch in der Dokumentation oder anderen (körperlichen oder elektronischen) Begleitmaterialien enthalten sein.

**Dokumentation und Lizenzvermerk.** In Ziffer 3.3 MPL findet sich die Verpflichtung, vorgenommene Veränderungen (*Modifications*) in nachvollziehbarer Weise in einer beigefügten Datei zu dokumentieren. Enthalten sein müssen danach eine Beschreibung der vorgenommenen Änderung und das Datum der Änderung. Darüber hinaus enthält Ziffer 3.3 bestimmte Hinweispflichten auf den Initial Developer: *“You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.”*

Beim Vertrieb der Software ist zudem in jeder Datei ein Lizenzvermerk aufzunehmen, wie er am Ende des Lizenztextes als „Exhibit A“ als Muster beigefügt ist. In diesem wird unter anderem auf den Haftungsausschluss hingewiesen, vgl. Ziffer 3.5. (i.V.m. Ziffer 3.6.) MPL.

**Dual Licensing.** Ziffer 13 der MPL sieht explizit die Möglichkeit des Dual Licensing, also der gleichzeitigen Lizenzierung unter mehreren Lizenzen vor. Wird beispielsweise Code unter der MPL und GPL lizenziert, darf der Code auch in ein GPL-Programm übernommen werden. Ansonsten kann MPL-Code nicht mit GPL-Programmen verbunden werden, da nicht gleichzeitig die sich gegenseitig ausschließenden Vertragsbedingungen der GPL und MPL befolgt werden können.

**Mitlieferung der Lizenz.** Gemäß Ziffer 3.1. MPL ist beim Vertrieb des Quellcodes – unabhängig davon, ob es sich um modifizierte oder nicht modifizierte Versionen handelt – ein Exemplar der Lizenz beizufügen. Zweck dieser Weitergabeverpflichtung ist es, jedem Erwerber ein Exemplar der Lizenz zu verschaffen, damit dieser durch Annahme des darin enthaltenen Angebots zum Abschluss einer Open Source Lizenz selbst die Rechte zum Kopieren, Bearbeiten und Verbreiten erhalten kann.

Ziffer 3.4. MPL sieht zudem bestimmte Hinweispflichten vor, wenn Rechte Dritter betroffen sein können. So muss in einer „LEGAL“ genannten Datei darauf hingewiesen werden, wenn die Rechte zur Lizenzierung von Modifikationen aus einem Vertrag mit Dritten hergeleitet werden.<sup>89</sup>

**Ausnahmen für den Objektcode bei der MPL-Lizenzierung.** Unter bestimmten engen Umständen – unter anderem einer Haftungsfreizeichnung gegenüber den an der Vorversion beteiligten Urhebern – kann der Objektcode auch unter einer anderen Lizenz als der MPL verbreitet werden. Dabei werden allerdings die Rechte am Sourcecode nicht berührt; dieser ist nach wie vor unter die MPL zu stellen und der entsprechende Sourcecode zugänglich zu machen.

### c) Kombination mit Softwarebestandteilen unter anderen Lizenzen

Soweit das Copyleft eingreift, dürfen modifizierte Versionen nur insgesamt unter der MPL vertrieben werden. Eine sehr eng begrenzte Ausnahme gilt – wie soeben dargestellt – nur für den Vertrieb des Objektcodes. Allerdings handelt es sich bei der MPL um eine Lizenz, die einen lediglich beschränkten Copyleft-Effekt hat. Es sind „Ausnahmen“

---

<sup>89</sup> Vgl. dazu die Anmerkungen in der kommentierten MPL, <<http://www.mozilla.org/MPL/MPL-1.1-annotated.html>>.

von einer strengen Bindung an das Copyleft vorgesehen. So gestattet Ziffer 3.7. den Nutzern, MPL-Software gemeinsam mit anders lizenzierter Software als Teil eines Ganzen zu verbreiten („*Larger Works*“), soweit alle Pflichten hinsichtlich des MPL-Teils erfüllt werden.<sup>90</sup>

Mit den Besonderheiten beim Vertrieb von „*Larger Works*“ korrespondiert der Begriff der „*Modifications*“, der in Ziffer 1.9. MPL ausdrücklich definiert wird und der insbesondere dafür entscheidend ist, wann die Copyleft-Pflicht für Bearbeitungen eingreift und wann nicht. Die Vorschrift bestimmt dabei, dass Modifikationen immer dann vorliegen sollen, wenn entweder Hinzufügungen oder Streichungen in bestehenden Dateien vorgenommen werden oder wenn Dateien hinzugefügt werden, die MPL-Code enthalten. Eine Modifikation i. S. d. MPL ist es daher nicht, wenn der Software eine eigene selbständige Datei hinzugefügt wird. Diese darf dann nach Belieben des Urhebers, also auch proprietär, als Teil eines „*Larger Work*“ vertrieben werden. Anders als bei der GPL erfolgt eine Unterscheidung also nur nach formalen Kriterien.

#### **IV. Nachträgliche Änderungen einer Open Source Lizenz**

Zahlreiche Open Source Lizenzen, insbesondere Lizenzen mit strengem oder beschränktem Copyleft-Effekt, sehen vor, dass die jeweilige Lizenz von Zeit zu Zeit angepasst werden kann. Dies kann notwendig werden, wenn sich die Anforderungen an die Open Source Lizenz verändern<sup>91</sup>. Die möglichen Gründe für die Notwendigkeit von Anpassungen einer Lizenz sind vielfältig. So ist denkbar, dass sich die äußeren (z.B. technischen) Umstände so fortentwickeln, dass das Lizenzmodell nicht mehr allen Möglichkeiten der Nutzung Rechnung trägt. Auch kann sich herausstellen, dass einzelne Aspekte der Lizenz an den Bedürfnissen von Lizenzgebern und/oder Lizenznehmern vorbeigehen.

Die verschiedenen Lizenzen übertragen dabei die „Lizenzhoheit“, also die Entscheidungsbefugnis, wann und mit welchem Inhalt neue Versionen der Lizenz veröffentlicht

---

<sup>90</sup> Die MPL ist in diesem Fall weniger streng als die GPL, die den äußerlich als ein Produkt zusammengefassten Vertrieb an sich selbständiger Werke als einheitliches Ganzes nur dann gestattet, wenn das Gesamtprogramm unter der GPL verbreitet wird.

<sup>91</sup> Dass solche Fälle vorkommen, zeigt sich an der GPL, die momentan in der 2. Fassung gilt. Version 3 befindet sich zurzeit in der Erstellung.

werden sollen, auf sehr unterschiedliche Träger. Teilweise sind einzelne Organisationen zur Herausgabe neuer Versionen befugt.<sup>92</sup> In anderen Fällen haben sich die Inhaber der Rechte an der Erstversion entschlossen, eigene Lizenzen zu verwenden, in denen sie sich selbst zur Herausgabe neuer Lizenzen ermächtigen, um langfristigen Einfluss auf die Lizenzierungspraxis ausüben zu können.<sup>93</sup> Einen Zwischenweg bietet neuerdings die Deutsche Freie Software Lizenz, bei der ein Lizenzrat über neue Versionen entscheidet.<sup>94</sup>

Unterschiedlich ist auch die Art und Weise der Mechanismen, mit der neue Lizenzversionen Geltung erlangen sollen. Zahlreiche Open Source Lizenzen sehen vor, dass der Rechtsinhaber wählen kann, ob die Software nur mit einer bestimmten oder aber mit jeder Version der Lizenz genutzt werden darf, die auf die Veröffentlichungsversion folgt.<sup>95</sup> Im letzteren Fall kann der Lizenznehmer dann diejenige Lizenzversion wählen, die als für ihn günstigste erscheint. Andere Lizenzen versuchen einen vollständigen Wechsel auf die jeweils aktuellste Lizenzversion zu erreichen.<sup>96</sup>

Die in den Open Source Lizenzen enthaltenen Klauseln zum Wechsel der Lizenzversion können je nach Ausgestaltung zu verschiedensten rechtlichen Problemen führen. So kann etwa dort eine unangemessene Benachteiligung des Nutzers vorliegen, wo diesem nachträglich Rechte abgeschnitten werden können.<sup>97</sup> Schwierigkeiten bereiten sämtliche Klauseln in den Open Source Lizenzen, die einen automatischen Wechsel auf spätere Lizenzversionen ermöglichen sollen, auch vor dem Hintergrund des § 31 Abs. 4 UrhG.<sup>98</sup> Nach dieser Vorschrift ist die Einräumung von Nutzungsrechten für noch nicht bekannte Nutzungsarten ebenso unwirksam wie die Verpflichtung dazu. Dies soll den Urheber vor der Vergabe von Rechten schützen, deren Umfang und wirtschaftliche Bedeutung er noch nicht absehen kann. Allein aufgrund einer Lizenzänderungsklausel

<sup>92</sup> Z.B. der GPL erfolgt die Entscheidung über neue Lizenzversionen durch die Free Software Foundation.

<sup>93</sup> Dies gilt beispielsweise für die MPL, bei welcher der Netscape Communications Corporation die Befugnisse zustehen, neue Lizenzen herausgeben.

<sup>94</sup> Vgl. dazu <<http://www.d-fsl.de>>.

<sup>95</sup> So z.B. § 9 GPL.

<sup>96</sup> Insb. § 9 D-FSL, <<http://www.d-fsl.de>>.

<sup>97</sup> Dies ist dort nicht der Fall, wo – wie etwa bei der GPL – die Befugnisse des Nutzers stets nur erweitert werden, indem er die volle Entscheidungsfreiheit erhält, welche Lizenzversion er befolgen möchte.

<sup>98</sup> Ob diese Regelung die weitere Reform des Urheberrechts überstehen wird, ist allerdings fraglich, vgl. dazu Referentenentwurf für ein Zweites Gesetz zur Regelung des Urheberrechts in der Informationsgesellschaft, <<http://www.bmj.bund.de/media/archive/760.pdf>>.

kann der Rechtsinhaber daher heute noch nicht die Entscheidung treffen, dass ein Dritter diese Nutzungsarten in spätere Lizenzen schreiben darf. Sollen später auftretende Nutzungsarten (nach-) lizenziert werden, bedarf es vielmehr einer bewussten konkreten Entscheidung des Urhebers, die bei der "any later version"-Klausel für den Lizenzwechsel gerade nicht erforderlich ist.

#### **V. *Praktische Hinweise: Was ist bei der Auswahl einer Open Source Lizenz zu berücksichtigen?***

Die Auswahl der Lizenzbedingungen stellt eine für die Zukunft des jeweiligen Open Source Projektes wesentliche Entscheidung dar. Sie ist daher mit besonderer Sorgfalt zu treffen.

Wahlmöglichkeiten bestehen dabei von vornherein nur dort, wo entweder die eigene Software erstmals unter eine Open Source Lizenz gestellt wird, oder aber dort, wo die für die jeweilige Software geltenden Lizenzen einen Wechsel der Lizenz beim Vertrieb von Modifikationen zulassen (z.B. BSD).

Soweit Wahlmöglichkeiten bestehen, ist zunächst zu prüfen, welche Lizenzfamilie für die Durchsetzung der verfolgten Zwecke am Besten geeignet erscheint. Während Non-Copyleft Lizenzen im Wesentlichen einer schnellen Standardisierung dienen können, zielen die Copyleft-Lizenzen auf einen in allen Phasen offenen Entwicklungsprozess. Software, die unter einer Copyleft-Lizenz steht, kann nicht „proprietär“ werden. Lizenzen mit beschränktem Copyleft-Effekt können dort sinnvoll sein, wo Kompromisse erforderlich sind.

In einem weiteren Schritt ist zu entscheiden, ob eine international verbreitete Lizenz (zumeist U.S.-amerikanischen Ursprungs) gewählt werden soll, oder ob möglicherweise eher (u.U. zusätzlich) auf eine deutsche oder europäische Lizenz zurückgegriffen werden sollte. Letztere berücksichtigen die Besonderheiten des deutschen und europäischen Urheber-, Vertrags- und Verbraucherschutzrechts häufig sehr viel besser. Sie bieten jedoch – umgekehrt – keine optimale Gestaltung der Rechtsbeziehungen für die internationalen Märkte (insb. für den wichtigen U.S.-amerikanischen Markt) und können unter Umständen Akzeptanzprobleme im Ausland haben.

Wichtig ist auch die „Lizenzkompatibilität“, also die Frage, ob und unter welchen Voraussetzungen die Software mit Software unter anderen Lizenzen kombiniert werden können soll. Hier lassen insbesondere die Lizenzen mit einem strengen Copyleft-Effekt vielfach nur sehr beschränkte Spielräume zu.

Schließlich ist auch die Frage nach der Kontrolle über die Herausgabe neuer Versionen der Lizenz von wichtiger Bedeutung für die Lizenzierungsentscheidung. Hier gibt es deutliche Unterschiede, in welchem Umfang der Urheber nach der Open Source Lizenzierung noch Einfluss auf spätere Versionen der Lizenz nehmen kann. Während bei zahlreichen Lizenzen die Entscheidung über neue Versionen zentral an einzelne Organisationen übertragen werden (z.B. GPL), haben sich in anderen Fällen (z.B. MPL) die Rechteinhaber der ursprünglichen Version entschlossen, eigene Lizenzen zu verwenden, um langfristigen Einfluss auf die Lizenzierungspraxis ausüben zu können.

## **Teil 4. Vertragliche Beziehungen bei der Nutzung von Freier Software**

### ***I. Überblick über die vertraglichen Beziehungen bei der Nutzung Freier Software***

Im Zusammenhang mit Open Source Software können Verträge mit sehr unterschiedlichen Vertragsinhalten geschlossen werden. So ist es beispielsweise möglich, dass der Nutzer lediglich ein oder mehrere Exemplare der Software benötigt, die er zur schlichten Programmbenutzung einsetzen möchte.<sup>99</sup> Bei dieser Software kann es sich um ein Massenprodukt handeln, sie kann aber auch individuell hergestellt oder angepasst worden sein. Weiter ist denkbar, dass der Nutzer auch die Rechte erwerben möchte, Veränderungen an der Software vorzunehmen, sowie die Software zu vervielfältigen, zu verbreiten und Dritten öffentlich zugänglich zu machen. Schließlich ist es ohne weiteres

---

<sup>99</sup> Der zur Verwendung des Programms Berechtigte darf die Software gemäß § 69d UrhG in bestimmungsgemäßem Umfang benutzen.

möglich, dass Service- oder Pflegeverträge hinsichtlich der Software getroffen werden sollen.

Diese unterschiedlichen Verträge müssen nicht zwingend zur selben Zeit und mit jeweils demselben Vertragspartnern abgeschlossen werden. Es sind vielmehr Verträge mit unterschiedlichen Beteiligten denkbar.

Die nachfolgende Darstellung verschafft einen Überblick über die verschiedenen, im Zusammenhang mit Open Source Software möglichen Verträge. Zugleich wird eine Zuordnung der Verträge zu bestimmten Vertragstypen (z.B. Kauf, Schenkung) vorgenommen. Diese Zuordnung ist von hohem praktischem Interesse. Sie hat zunächst Bedeutung für den Umfang der Pflichten der beteiligten Vertragsparteien: Das Gesetz hält für die unterschiedlichen Vertragstypen sogenannte „dispositive“ Vorschriften bereit, die eingreifen, wenn die Parteien Regelungen einzelner Punkte unterlassen haben oder die getroffenen Regelungen nicht wirksam sind. Fragen der Haftung und Gewährleistung sind sehr unterschiedlich geregelt. Darüber hinaus kennt das Gesetz zahlreiche zwingende Vorschriften für einzelne Vertragstypen. Bedeutung hat die Zuordnung einzelner Verträge zu einem bestimmten Vertragstyp schließlich auch im Bereich des Rechts der Allgemeinen Geschäftsbedingungen sowie für Haftung und Gewährleistung. Je nach Vertragstyp können unterschiedliche Regelungen zulässig sein.

## ***II. Verträge über die Überlassung der Software zur Benutzung***

Häufig wird man eine Software allein aus dem Grunde erwerben wollen, um diese zur Erfüllung der eigenen privaten oder unternehmensinternen Aufgaben einzusetzen. Eigene Änderungen an der Software oder eine Weitergabe an Dritte sollen nicht stattfinden. Als Vertragspartner des Kunden kommt in diesen Fällen neben den Rechtsinhabern jedermann in Betracht, der mit den Rechtsinhabern eine Open Source Lizenz abgeschlossen hat. Denn aufgrund einer solchen Lizenz erwirbt man Vertriebsrechte und darf die Software vervielfältigen, verbreiten und öffentlich zugänglich machen. Jeder Lizenznehmer einer Open Source Lizenz ist daher zur Erfüllung eines Vertrages über die Überlassung der Software in der Lage; er kann zum „Distributor“ der Software werden.

Die vertragstypologische Einordnung der Verträge über die Überlassung der Software zur Benutzung ist zunächst davon abhängig, ob dem Erwerber die Software – was der Regelfall ist – endgültig überlassen werden soll, oder ob eine lediglich zeitweise Überlassung erfolgt.<sup>100</sup> Weiter kommt es darauf an, ob eine Gegenleistung durch den Nutzer zu erbringen ist und worin diese besteht. Die in der Praxis häufigsten Fälle der Überlassung von Open Source Software sind die dauerhafte Überlassung der Software gegen Einmalentgelt (z.B. Erwerb einer freien Betriebssystemdistribution) und die dauerhafte kostenlose Überlassung der Open Source Software (z.B.: Download via Internet).

### 1. Dauerhafte Überlassung gegen einmalige Bezahlung

Eine Überlassung von Open Source Software gegen Entgelt ist ohne weiteres zulässig. Zwar ist in der juristischen Diskussion vielfach behauptet worden, Open Source Software dürfe nur unentgeltlich vertrieben werden. Entsprechende Ansichten beruhen jedoch auf einer unrichtigen oder unvollständigen Erfassung der Tatsachengrundlagen. Sie stützen sich zumeist auf Ziffer 1 der Open Source Definition.<sup>101</sup> Diese Klausel untersagt jedoch allein, dass für die Lizenzierung der Vervielfältigungs- und Verbreitungsrechte Lizenzgebühren verlangt werden. Die Überlassung der Software, die ein davon zu trennendes Geschäft darstellt, wird hingegen nicht erfasst: Zwar erwirbt man die Rechte zum Vertrieb kostenlos; das bedeutet jedoch nicht, dass man die Software nur kostenlos vertreiben darf. Die Software darf vielmehr zu jedem erzielbaren Preis angeboten und veräußert werden.<sup>102</sup>

Erfolgt die Überlassung der Open Source Software entgeltlich, handelt es sich regelmäßig um einen Vertrag, der dem Vertragstyp „Kauf“ (§§ 433 ff. BGB) zugeordnet werden kann: Der Vertragstyp „Kauf“ stellt sachgerechte Regelungen für reine Umsatzgeschäfte zur Verfügung. Die Rechtsprechung hat daher seit Ende der achtziger Jahre Fallgruppen herausgearbeitet, in denen Kaufvertragsrecht auf den Softwareüberlassungsvertrag –

---

<sup>100</sup> Ob eine Vermietung von Open Source Software zulässig ist, ist zunächst von der Ausgestaltung der Lizenz im Einzelnen abhängig. Wo lediglich vom Recht „to distribute“ gesprochen wird, ist die Zulässigkeit der Vermietung streitig, im Ergebnis aber zu bejahen, vgl. *Jaeger/Metzger*, Open Source Software, S. 33 f.

<sup>101</sup> “The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.”

<sup>102</sup> *Stallman*, <<http://www.gnu.org/philosophy/selling.html>>: “You can charge nothing, a penny, a dollar, or a billion dollars. It’s up to you and the marketplace, so don’t complain to us if nobody wants to pay a billion dollars for a copy.”

zumindest entsprechend – Anwendung findet. Dies ist im Wesentlichen dann der Fall, wenn es sich um die reine Überlassung von Software auf Dauer und gegen Zahlung eines Einmalentgelts handelt.<sup>103</sup> Unbeachtlich ist hingegen, ob die Software auf einem Datenträger oder im Wege der elektronischen Datenübermittlung überlassen wurde.<sup>104</sup>

## 2. Dauerhafte unentgeltliche Überlassung

Open Source Software wird vielfach auch unentgeltlich überlassen, z.B. indem sie kostenlos zum Download im Internet bereitgestellt wird. Solche Verträge, die auf die kostenlose dauerhafte Überlassung der Software gerichtet sind, sind rechtlich dem Vertragstyp Schenkung zuzuordnen. Eine Schenkung im Sinne des § 516 BGB setzt voraus, dass sich beide Parteien darüber einig sind, dass eine Zuwendung aus dem Vermögen des Schenkers unentgeltlich erfolgt.<sup>105</sup>

### III. Verträge über die Einräumung von Entwicklungs- und Vertriebsrechten

Von den Verträgen über die Überlassung der Software sind die Verträge über die Einräumung der Entwicklungs- und Vertriebsrechte zu unterscheiden: Teilweise soll die Software nicht lediglich intern eingesetzt werden, sondern vielmehr verändert und/oder an Dritte weitergegeben werden. In diesen Fällen reicht ein schlichter Erwerb der Software aufgrund dessen die Software bestimmungsgemäß benutzt werden darf (§ 69d UrhG) nicht. Es ist vielmehr erforderlich, dass ein Vertrag geschlossen wird, durch den die erforderlichen Rechte eingeräumt werden. Vertragspartner entsprechender Verträge sind der jeweilige Lizenznehmer und die Rechtsinhaber selbst. Der Inhalt des Vertrages wird durch die jeweilige Open Source Lizenz bestimmt.<sup>106</sup>

Bei der vertragstypologischen Einordnung der Verträge kann unterschieden werden zwischen den sog. Non-Copyleft-Lizenzen und den Copyleft-Lizenzen. Während Non-Copyleft-Lizenzen wohl einheitlich als unentgeltliche Verträge eingeordnet werden, ist

---

<sup>103</sup> U.a. BGH CR 1988, 124, 126 f.; BGH CR 1993, 203, 204; BGH CR 1993, 681, 683; BGH CR 2000, 207, 208. Eine umfassende Auswertung der Entscheidungen findet sich bei *Schneider*, Hdb. des EDV-Rechts, Kap. D, Rz. 166 ff.

<sup>104</sup> Vgl. nur *Pres*, Urheberrechtliche Softwarelizenzverträge, S. 170; *Schneider*, Hdb. des EDV-Rechts, Kap. N, Rz. 53.

<sup>105</sup> Zur schenkungsrechtlichen Einordnung vgl. ausführlich *Jaeger/Metzger*, Open Source Software, S. 141 f.; ähnlich *Spindler* in: *Spindler*, Rechtsfragen bei Open Source, S. 154 ff., allerdings ohne deutliche Trennung zwischen Überlassungs- und Lizenzvertragsgeschäft.

<sup>106</sup> Lediglich in diesem Vertragsverhältnis spielt die Open Source Lizenz eine Rolle. Hingegen betrifft sie das schlichte Erwerbsgeschäft in der Regel nicht.

dies bei Copyleft-Lizenzen streitig. Auch hier erfolgt vielfach eine Zuordnung zu den unentgeltlichen Verträgen. Zum Teil werden diese Arten von Lizenzen jedoch auch als teilweise „entgeltliche“ Verträge eingestuft, mithin als ein solche, bei denen bestimmte Leistungen im Gegenseitigkeitsverhältnis stehen. Denn hier wird der Lizenznehmer immerhin verpflichtet, die eigenen Modifikationen beim Vertrieb wiederum „Open Source“ zu lizenzieren und den Quellcode offen zu legen.

## 1. Non-Copyleft-Lizenzen

Wohl unstrittig stellen Non-Copyleft-Lizenzen „unentgeltliche“ Verträge dar: Die in allen Open Source Lizenzen begründeten Verpflichtungen des Lizenznehmers, beim Vertrieb unveränderter Versionen einen Copyrightvermerk, einen Haftungsausschluss und die Open Source Lizenz beizufügen, sowie in näher bezeichneter Art und Weise einen Zugang zum Quellcode sicherzustellen, beziehen sich ausschließlich auf „vertriebsgestaltende“ Umstände des erst durch die Einräumung der Nutzungsrechte ermöglichten Vertriebs. Vor diesem Hintergrund ist davon auszugehen, dass die Einräumung der Vertriebsrechte unentgeltlich erfolgt. Denn die Vertriebsmodalitäten, die allein Inhalt und Umfang der Ausübung der vom Lizenzgeber eingeräumten Rechte regeln, bilden kaum ein gegenüber der Rechtseinräumung gleichwertiges Leistungsobjekt. Und da sich die Pflichten des Lizenznehmers ausschließlich auf die spätere Ausübung der vom Lizenzgeber einzuräumenden Rechte beschränken, kann schwerlich davon gesprochen werden, dass der Lizenzgeber die Verpflichtung zur Nutzungsrechtseinräumung nur deshalb eingeht, weil er die Einhaltung der Vertriebsmodalitäten möchte.<sup>107</sup> Die Einhaltung dieser Verpflichtungen ist also gerade keine „Gegenleistung“.<sup>108</sup>

Um welche Art von unentgeltlichem Vertrag es sich bei den Non-Copyleft-Lizenzen allerdings genau handelt, ist durchaus streitig. Vielfach wird davon ausgegangen, es liege eine Schenkung (unter Auflage) oder einen schenkungsähnlicher Vertrag vor.<sup>109</sup> Andere gehen eher von einem urheberrechtlichen Nutzungsvertrag eigener Art aus, auf den

---

<sup>107</sup> Vgl. nur *Deike*, CR 2003, 9, 14 f.

<sup>108</sup> Obwohl die Parteien grundsätzlich die Freiheit haben, darüber zu bestimmen, welche Pflichten in einem Gegenseitigkeitsverhältnis stehen sollen, gibt es bestimmte Pflichten, die sich von vornherein zur Einbindung in ein Austauschverhältnis nicht eignen, *Gernhuber*, Schuldverhältnis, § 13 IV 1 in Fn. 37.

<sup>109</sup> *Jaeger/Metzger*, Open Source Software, S. 139 ff.; in diese Richtung auch *Spindler/Wiebe*, CR 2003, 873, 879; *Deike*, CR 2003, 9, 14 f.

einzelne gesetzliche Regelungen bekannter unentgeltlicher Verträge (insb. Leihe) Anwendung finden können.<sup>110</sup>

In der Praxis führen die verschiedenen Ansichten allerdings in der Regel nicht zu wesentlichen Unterschieden. Insbesondere ist man sich einig darüber, dass hinsichtlich der Haftung und Gewährleistung des Lizenzgebers eine weitgehende Beschränkung besteht.<sup>111</sup>

## 2. Copyleft-Lizenzen

Während bei den Non-Copyleft-Lizenzen Einigkeit darüber besteht, dass es sich um unentgeltliche Verträge handelt, ist dies bei den Copyleft-Lizenzen nicht der Fall. Verbreitet wird zwar auch hier davon ausgegangen, dass es sich wie bei den Non-Copyleft-Lizenzen um einen in jeder Hinsicht unentgeltlichen Vertrag handle. Die vertragsrechtliche Einordnung hat dann ebenso wie bei den Non-Copyleft-Lizenzen zu erfolgen.<sup>112</sup> Dies wird damit begründet, dass die Verpflichtungen aus dem Copyleft nur „nachhängend“ sind, d.h. nicht in unmittelbarem Zusammenhang mit der Zuwendung selbst stehen, sondern erst bei der Vornahme zusätzlicher Handlungen entstehen.<sup>113</sup>

Eine Gegenansicht nimmt jedoch an, dass es sich lediglich hinsichtlich des Vertragsteils „Vertrieb unveränderter Versionen“ um einen unentgeltlichen Vertrag handelt. Demgegenüber stünden die Pflicht zur Einräumung der Rechte zum Vertrieb modifizierter Versionen einerseits und die durch den Vertrieb der Modifikationen bedingten Verpflichtungen zur Offenlegung des Quellcodes und zum Open Source Angebot an jedermann andererseits in einem Gegenseitigkeitsverhältnis.<sup>114</sup> Eine Unentgeltlichkeit dieses Vertragsteils komme daher nicht in Betracht. Es bestehe ein besonderes „Motivationsverhältnis“ beider Parteien für die eigene Verpflichtung zur Leistung. Die vom Lizenzgeber zu erteilende Befugnis, veränderte Versionen der Software vertreiben zu dürfen werde allein deshalb gewährt, „damit“ und „weil“ auch der Lizenznehmer dem Lizenzgeber und jedem Dritten wiederum dieselben Befugnisse und Möglichkeiten an dem von ihm

---

<sup>110</sup> Schulz, Dezentrale Softwareentwicklungs- und Softwarevermarktungskonzepte, Kapitel 5, C IV. 2.

<sup>111</sup> Vgl. dazu unten S. 71 f.

<sup>112</sup> Vgl. nur Jaeger/Metzger, Open Source Software, S. 143.

<sup>113</sup> Metzger/Jaeger, GRUR Int. 1999, 839, 847.

<sup>114</sup> Schulz, Dezentrale Softwareentwicklungs- und Softwarevermarktungskonzepte, Kapitel 5, C I. 3.

geschaffenen Code, über den er tatsächlich verfügen kann und der ihm (bearbeiter-) urheberrechtlich zugewiesen ist, verschaffe.

Solange keine gerichtliche Klärung der Frage existiert, kann noch keine eindeutige Zuordnung getroffen werden. Für die Praxis ist dieser Streit dabei jedoch insbesondere deshalb von Bedeutung, da hinsichtlich entgeltlicher Vertragsteile ein abgemilderter Haftungsmaßstab in der Regel nicht in Betracht kommt.<sup>115</sup>

#### ***IV. Erstellungs-, Anpassungs- und Wartungsverträge***

Neben den schlichten Erwerbsgeschäften und den Geschäften über die Einräumung von Vertriebs- und Entwicklungsrechten können zahlreiche weitere Vertragskonstellationen im Umfeld Freier Software auftreten. Insbesondere wird Open Source Software vermehrt im Kundenauftrag vollständig neu erstellt oder aber bestehende Software an spezifische Kundenbedürfnisse angepasst. Hinzu kommen begleitende Service- und Supportabreden.

##### **1. Neuprogrammierung einer Open Source Software**

Programmierung von Open Source Software braucht nicht stets „im stillen Kämmerlein“ zu beginnen. Dem Vertrieb von Open Source Software und der Einräumung von Rechten zur Vervielfältigung, Verbreitung, öffentlichen Zugänglichmachung und Bearbeitung kann vielmehr vorausgehen, dass die Software im Auftrag eines Kunden hergestellt wird.

Hinsichtlich der Rechteübertragung sind bei der Neuprogrammierung von Open Source Software im Kundenauftrag im Wesentlichen zwei Konstellationen denkbar. Entweder wird vereinbart, dass der Auftragnehmer, der die Software im Kundenauftrag programmiert, diese selbst unter eine Open Source Lizenz stellt. Oder aber der Auftraggeber lässt sich zunächst die ausschließlichen Nutzungsrechte übertragen und stellt die Software dann selbst unter eine Open Source Lizenz.<sup>116</sup> Dabei ist es in allen Fallkonstellationen auch möglich, zu vereinbaren, dass die Software lediglich „auch“ unter eine Open

---

<sup>115</sup> Vgl. dazu im Einzelnen unten S. 72 f.

<sup>116</sup> Je nach Konstellation wird dann entweder der Auftragnehmer oder der Auftraggeber Vertragspartner von Dritten, die einen Open Source Lizenzvertrag (Vertrag über die Einräumung von Entwicklungs- und Vertriebsrechten) abschließen möchten.

Source Lizenz gestellt wird und daneben eine proprietäre Vermarktung zulässig bleibt (sog. Dual Licensing)<sup>117</sup>.

Schwierig ist bei der vertragstypologischen Einordnung der Softwareerstellungsverträge vor allem die Abgrenzung in der Anwendung von Kauf- (§ 651 BGB; §§ 433 ff. BGB) oder Werkvertragsrecht (§§ 631 ff. BGB). Diese Unterscheidung ist wichtig z.B. im Hinblick auf Mängelrügepflichten im kaufmännischen Verkehr und auf das Erfordernis einer Abnahme durch den Besteller.

Wird ein bestimmtes Entgelt vereinbart, so ist der Vertrag über die Erstellung von Open Source Software im Kundenauftrag nach einer gewichtigen Ansicht als Werkvertrag im Sinne des § 631 BGB einzuordnen.<sup>118</sup> Der Auftragnehmer verpflichtet sich zur Herstellung eines bestimmten immateriellen Werkes, der Besteller zur Entrichtung einer Vergütung. Die Einordnung als Werkvertrag ist allerdings nicht unbestritten, seit aufgrund des Schuldrechtsmodernisierungsgesetzes zum 1. Januar 2002 in § 651 BGB die Regelung eingefügt wurde, dass sich Verträge über die Lieferung herzustellender oder zu erzeugender beweglicher Sachen im Wesentlichen nach Kaufrecht richten. Denn vielfach wird davon ausgegangen, dass es sich bei Software um eine bewegliche Sache im Sinne dieser Vorschrift handelt.<sup>119</sup> Insoweit bestehen auch keine Unterschiede zwischen Open Source Software und herkömmlicher Software.

Insgesamt bietet es sich hier zum gegenwärtigen Zeitpunkt im Hinblick auf die keineswegs eindeutige Rechtslage an, die Verpflichtungen der Parteien detailliert zu regeln und etwa eine (aus dem Werkvertragsrecht bekannte) Abnahme ausdrücklich zu vereinbaren sowie im unternehmerischen Verkehr mögliche Mängelrügefristen abzubedingen oder nach den spezifischen Bedürfnissen zu regeln.

## 2. Anpassung an kundenspezifische Bedürfnisse

Bei der entgeltlichen Anpassung bereits bestehender Open Source Software an die kundenspezifischen Bedürfnisse sind verschiedene Fallgruppen zu unterscheiden.

---

<sup>117</sup> Vgl. zum Dual Licensing ausführlicher *Jaeger/Metzger*, Open Source Software, S. 79 ff.

<sup>118</sup> Wie hier u.a. *Palandt/Sprau*, § 651, Rz. 5 m.w.N.

<sup>119</sup> Vgl. ausführlicher zu dieser Problematik *Thewalt*, CR 2002, 1 ff.

Erfolgt die Lieferung einer Software gemeinsam mit vom Lieferanten eingepflegten Änderungen, so stellen sich zunächst dieselben Abgrenzungsprobleme wie bei der Neuerstellung von Individualsoftware. Es kommt darauf an, ob man Software als bewegliche Sache einordnet. In diesem Fall ist über § 651 BGB im Wesentlichen Kaufrecht anwendbar. Ansonsten kommt die Anwendung von Werkvertragsrecht in Betracht. Letzteres gilt allerdings dann nicht, wenn es sich nur um kleinere Änderungen handelt, die gegenüber dem reinen Umsatzgeschäft zurücktreten. In diesen Fällen ist wohl unstreitig Kaufrecht anwendbar.

Wird die Software vom Kunden zur Verfügung gestellt, so ist in der Regel Werkvertragsrecht anwendbar.<sup>120</sup>

Insgesamt ist auch hier aufgrund der bestehenden Unsicherheiten in der Rechtsanwendung zu empfehlen, dass die Vertragsparteien die gegenseitigen Verpflichtungen ausführlich regeln und insbesondere auch Abreden zu möglichen Mängelrügepflichten und Abnahmeerfordernissen treffen.

### **3. Service- und Supportverträge**

Im Zusammenhang mit Open Source Software werden vielfach entgeltliche Service- und Supportleistungen angeboten. Auch wenn der Inhalt der jeweiligen Leistungspflichten dabei von der Abrede im Einzelfall abhängig ist, lassen sich die Pflichten grob wie folgt unterscheiden:

- Behebung von bestehenden oder auftretenden Mängeln einer Software (Fehlerbeseitigung)
- Aktualisierung der Software je nach Fortgang der Weiterentwicklung oder Vereinbarung künftiger Verbesserungen (Update/Aktualisierung)
- Beratung im Umgang mit dem Computerprogramm (Beratung)

---

<sup>120</sup> Vgl. zu einer möglichen Systematik in der Abgrenzung von Kauf und Werkvertragsrecht insb. *Schneider*, Hdb. des EDV-Rechts, Kap. H, Rz. 3.

Je nachdem, ob nach der Parteiabsprache die Leistungen eher erfolgsbezogen ausgestaltet sind, oder ob sie sich auf die gewissenhafte Durchführung der Maßnahmen beschränken, haben diese im ersten Falle einen eher werkvertraglichen (§§ 631 ff. BGB) im zweiten einen eher dienstvertraglichen (§§ 611 ff. BGB) Charakter. Diese Abgrenzung ist wichtig, da im ersteren Falle ein konkretes Leistungsergebnis geschuldet wird.

Innerhalb der Grundtypen Fehlerbeseitigung und Update/ Aktualisierung spricht dabei viel dafür, bei Fehlen abweichender konkreter Abreden grundsätzlich von einer weitgehenden Erfolgsbezogenheit auszugehen und die vereinbarten Leistungen als werkvertraglich zu qualifizieren.<sup>121</sup> Demgegenüber ist der Fall der schlichten Beratung eher als dienstvertragliche Leistung einzuordnen.<sup>122</sup>

#### **V. Grenzen der Aufspaltbarkeit in entgeltliche und unentgeltliche Vertragsteile**

Im Hinblick auf eine interessengerechte Vertragsgestaltung stellt sich bei Open Source Software die äußerst praxisrelevante Frage, ob es möglich ist, Verträge in der Weise zu gestalten, dass die Parteien verabreden, die Software solle unentgeltlich überlassen werden, es werde aber ein zeitlich befristeter entgeltlicher Vertrag über die Fehlerbeseitigung geschlossen. Ziel ist es, eine (aufgrund der unentgeltlichen Überlassung) gemilderte Haftung und Gewährleistung für die Software zu erreichen, gleichzeitig aber ein „maßgeschneidertes“ Servicepaket anzubieten. Problematisch ist dies insbesondere deshalb, da hier die Gefahr besteht, dass durch eine Aufspaltung in Einzelabreden die kaufrechtliche Mängelgewährleistung ausgehebelt wird.<sup>123</sup> In diesem Zusammenhang können folgende Fallgruppen unterschieden werden:

Erhält der Erwerber nicht die Möglichkeit, das Geschäft über die unentgeltliche Überlassung der Software zur Nutzung isoliert vom Servicevertrag abzuschließen, sondern besteht von vornherein allein ein Angebot zum gleichzeitigen Abschluss von Überlassungs- und Servicevereinbarung, so gestaltet sich das Geschäft bei wirtschaftlicher Be-

---

<sup>121</sup> Für den Bereich Update/Aktualisierung ebenso *Schneider*, Hdb. des EDV-Rechts, Kap. D, Rz. 258.

<sup>122</sup> *Schneider*, Hdb. des EDV-Rechts, Kap. D, Rz. 263.

<sup>123</sup> Ähnliche Probleme können hinsichtlich der Frage einer möglichen Aufspaltung von (unentgeltlicher) Softwareüberlassung und (entgeltlicher) Anpassung an die kundenspezifischen Bedürfnisse entstehen.

trachtung als kaufähnlich. Die Software wird dauerhaft überlassen und die davon nicht trennbare Mängelgewährleistung, die auch beim Kauf einen wesentlichen preisbildenden Faktor darstellt, wird nur gegen Entgelt gewährt. Die Aufspaltung in die einzelnen verbundenen Abreden ist in diesen Fällen nicht zulässig. Denn sie kann hier regelmäßig nur dem Zweck dienen, die besonderen AGB- und Verbraucherschutzrechtlichen Vorschriften des Kaufrechts zu umgehen.

Hatte der Erwerber hingegen tatsächlich die Möglichkeit, mit demselben Vertragspartner einen isolierten Vertrag über eine kostenlose Überlassung der Programmkopie zu schließen, dann erscheint fraglich, ob das Geschäft insgesamt als kaufähnliche Gestaltung zu beurteilen ist. Hier wird man die weitere Diskussion abwarten müssen. Zwar besteht auch insoweit vordergründig die soeben geschilderte Nähe zum Kauf. Die enge Verbindung zwischen beiden Abreden ist jedoch nicht ohne Alternativen, sondern abhängig von der freien Entscheidung des Erwerbers. Würde man auch in diesem Fall die Vertragsgestaltung als kaufähnlich einordnen, könnten Wertungswidersprüche entstehen. Denn da Open Source Software dezentral verbreitet wird, ist es für den Erwerber in der Regel ohne weiteres möglich, die Software von einem Dritten zu beschaffen und sodann (problemlos) einen maßgeschneiderten schlichten Service- und Supportvertrag mit dem Vertragspartner zu schließen; diese Möglichkeit wäre ihm hingegen beim Erwerb „aus einer Hand“ verbaut. Im Ergebnis wird man daher in diesem Fall die Möglichkeit einer Aufspaltung in zwei unabhängige Verträge befürworten müssen.

#### ***VI. Praktische Hinweise: Was ist bei der vertraglichen Gestaltung von Programmierung, Vertrieb und Verwendung von Open Source Software zu berücksichtigen?***

Die Gestaltung von Verträgen ist stets von unterschiedlichen Faktoren abhängig. Neben die Regelungen, welche die primären wirtschaftlichen Ziele betreffen, treten regelmäßig Abreden über die Risikoverteilung bei Problemen in der Durchführung oder Abwicklung der Verträge. Gleichzeitig ist bei der Ausgestaltung von Verträgen zu berücksichtigen, dass das Gesetz für verschiedene Arten von Verträgen unterschiedliche Regelungen bereit halten kann. Insgesamt kann es sich anbieten, folgende Merkposten zu berücksichtigen:

*Welches primäre wirtschaftliche Ziel wird verfolgt?* – Zunächst geht es um die Frage, ob die schlichte Überlassung von Software, die Einräumung von Nutzungsrechten oder aber sonstige Leistungen (Erstellung/ Wartung/ Pflege) begehrt werden. Gleichzeitig ist zu klären, ob eine bestimmte Gegenleistung erbracht werden soll (z.B. Zahlung eines bestimmten Entgelts).

*Welchen Vertragstyp hält das Recht zur Erreichung des wirtschaftlichen Ziels bereit?* – In einem zweiten Schritt wird geklärt, welcher Vertragstyp auf das jeweilige Geschäft Anwendung findet. Damit wird der Rahmen der gesetzlichen Regelungen festgelegt, die vertragstypspezifisch Anwendung finden.

*Können, sollten und dürfen abweichende Absprachen getroffen werden?* – Insoweit geht es zunächst um die Frage, ob schon das Gesetz selbst sachgerechte Regelungen zur Verfügung stellt, um einen Interessenausgleich zwischen den Vertragsparteien sicherzustellen, oder ob abweichende Regelungen getroffen werden sollten. In einem weiteren Teilschritt ist dann zu klären, ob derartige abweichende Regelungen gesetzlich zulässig sind.

*Ist der Vertrag in der abschließenden Gestalt rechtlich wirksam?* – Eine abschließende Kontrolle der rechtlichen Wirksamkeit des Vertrages muss zwingend erfolgen, soll auch der wirtschaftliche Erfolg des Geschäfts sichergestellt werden. Unwirksame Regelungen im Bereich der Haftung und Gewährleistung etwa können schnell zu massiven wirtschaftlichen Ausfällen führen.

## **Teil 5. Haftung und Gewährleistung**

### ***I. Allgemeines / Unterschiede zu proprietär verwerteten Programmen***

#### **1. Begrifflichkeiten: Haftung und Gewährleistung**

Einer der wesentlichen Punkte in der Bewertung der Risiken der Vertragsparteien bei Abschluss und Durchführung von Verträgen im Open Source Sektor ist die Frage nach

der Haftung und Gewährleistung. Unter Gewährleistung versteht man das Einstehenmüssen für die Vertragsgemäßheit des Programms, also insbesondere dafür, dass das Programm einerseits nicht mit Fehlern behaftet ist und dass andererseits der Benutzung des Programms keine Rechte Dritter entgegenstehen. Im ersteren Falle spricht man von Sachmängelhaftung, im letzteren Falle von Rechtsmängelhaftung.<sup>124</sup> Der Begriff der Haftung ist demgegenüber weiter. Er erfasst zunächst das vertragliche Einstehenmüssen für Schäden, die sich an sonstigen Rechtsgütern des Vertragspartners ergeben, etwa Schäden an Hardware oder anderer Software (vertragliche Haftung). Mit Haftung bezeichnet man darüber hinaus auch jedes sonstige außervertragliche Einstehenmüssen für Schäden (außervertragliche Haftung). Haftung in einem weiteren Sinne beschreibt schließlich ganz allgemein jedes zivilrechtliche Einstehenmüssen einschließlich der Gewährleistung.

## **2. Gewährleistungs- und Haftungsmaßstäbe**

Für Haftung und Gewährleistung können – je nach Ausgestaltung der einzelnen vertraglichen Beziehungen – unterschiedliche gesetzliche Maßstäbe gelten. So können die Voraussetzungen insbesondere nach dem zugrundeliegenden Vertragstyp variieren. Dabei greifen die vielfach für unentgeltliche Verträge geltenden Haftungserleichterungen teilweise auch auf die außervertragliche Haftung durch.<sup>125</sup>

Ein gegenüber den jeweiligen gesetzlichen Regelungen schärferer oder milderer Haftungs- und Gewährleistungsmaßstab kann sich aus vertraglichen Vereinbarungen ergeben. Die Parteien sind innerhalb gewisser Grenzen dazu befugt, Abreden über das Einstehenmüssen für Schäden und Mängel zu treffen. Werden diese Vereinbarungen nicht wirksam getroffen, etwa weil man sich außerhalb der gesetzlich zulässigen Vereinbarungen befindet, gilt wiederum der gesetzliche Haftungsmaßstab.

---

<sup>124</sup> Seit der Schuldrechtsreform 2002 sind die früher sehr unterschiedlichen Rechtsfolgen der Sach- und Rechtsmängelhaftung angeglichen worden, vgl. *Palandt/Putzo*, § 435, Rz. 1.

<sup>125</sup> Vgl. dazu nur *Palandt/Sprau*, Einf. v. § 823 Rz. 5.

### 3. „Beschaffenheit“ und Sachmängelgewährleistung

Neben dem jeweiligen, in einer bestimmten Vertragsbeziehung geltenden Haftungs- und Gewährleistungsmaßstäben kann das „Einstehenmüssen“ auch durch weitere Faktoren beeinflusst werden.

Zu diesen Faktoren zählt im Bereich der Haftung für Sachmängel der Software die jeweilige „Beschaffenheit“ oder „Tauglichkeit“ des Vertragsgegenstandes. Eine Sachmängelhaftung kommt von vornherein nicht in Betracht, wenn der Vertragsgegenstand die vertraglich vereinbarte Beschaffenheit<sup>126</sup> bzw. Tauglichkeit<sup>127</sup> aufweist. Der Vertragsgegenstand ist dann frei von Sachmängeln. Den Ansatzpunkt für eine Beschaffenheitsvereinbarung bildet dabei bei Individualsoftware regelmäßig das Pflichtenheft, bei Standardsoftware die jeweiligen Produkt- und Leistungsbeschreibungen.<sup>128</sup>

Fehlt eine Beschaffenheits- oder Tauglichkeitsvereinbarung, liegt ein Mangel dann nicht vor, wenn sich die Sache für die jeweilige nach dem Vertrag vorausgesetzte Verwendung eignet, sonst wenn sie sich für die gewöhnliche Verwendung eignet und eine Beschaffenheit aufweist, die üblich ist und die vom Vertragspartner erwartet werden kann.

Da sich die Frage, ob ein Mangel vorliegt, in erster Linie nach der vertraglich vereinbarten Beschaffenheit richtet, bilden entsprechende vertragliche Abreden eine zentrale Möglichkeit für die Schaffung einer sachgerechten Risikoverteilung. Allerdings sind die Gestaltungsmöglichkeiten hier nicht unbegrenzt: So wird man davon auszugehen haben, dass etwa der Verkauf funktionsfähiger Software durch Distributoren an Endverbraucher zu Marktpreisen unter der Bezeichnung „nichtstabile Entwicklerversion“ nicht dazu führt, dass der Verkäufer nicht für die Lauffähigkeit der Software einstehen muss. Entsprechende Bezeichnungen dienen allein einer Umgehung des gesetzlichen Haftungsmaßstabes und sind daher – das folgt beim Verbrauchsgüterkauf aus § 475

---

<sup>126</sup> Zum Begriff der „Beschaffenheit“, der nach der Schuldrechtsreform noch keine in jeder Hinsicht festen Konturen entwickeln konnte, *Palandt/Putzo*, § 434, Rz. 9 ff.; *Bamberger/Roth/Faust*, BGB, § 434, Rz. 13 ff.

<sup>127</sup> Vgl. zur „Tauglichkeit“ *Palandt/Weidenkaff*, § 536, Rz. 16 ff.; *Bamberger/Roth/Ehlert*, BGB, § 536, Rz. 15 ff.; auch zum Verhältnis der Begriffe Tauglichkeit und Beschaffenheit.

<sup>128</sup> Ausführlicher dazu *Marly*, Softwareüberlassungsverträge, Rz. 847 ff.

BGB – unwirksam.<sup>129</sup> Anderes gilt beispielsweise dort, wo Alpha-Versionen auf einer Entwicklerseite zum Download angeboten werden.

Dort, wo eine Beschaffenheit nicht vereinbart wurde, gilt die Eignung für die vertraglich vorausgesetzte oder die gewöhnliche Verwendung als Maßstab. Hier ist teilweise vorgeschlagen worden, eine Differenzierung zwischen proprietärer Software und Open Source Software vorzunehmen. Da Open Source Software vielfach auch im privaten Bereich entwickelt werde, seien von vornherein keine gleichermaßen hohen Anforderungen zu stellen.<sup>130</sup> Doch dürfte eine derart pauschale Begünstigung beim Vertrieb von Freier Software nicht greifen: Veräußert ein Händler ein Textverarbeitungsprogramm an einen Endverbraucher, so kann Letzterer erwarten, dass dieses die zentralen Funktionen auch beherrscht. Dies gilt unabhängig davon, ob es sich um ein proprietäres oder ein Open Source Programm handelt. Anderes kann bei der Weitergabe von Entwicklerversionen oder bei der Weitergabe im privaten Bereich gelten. Der Maßstab, der anzulegen ist, ist daher stets vom Einzelfall abhängig; eine pauschale Gegenüberstellung von Open Source Software und proprietärer Software kann nicht erfolgen.

#### **4. Unterschiede zu proprietär verwerteten Programmen**

Unterschiede im Haftungs- und Gewährleistungsmaßstab gegenüber proprietär verwerteten Programmen ergeben sich in erster Linie daraus, dass im Open Source Bereich Geschäfte in einem sehr viel größeren Umfang unentgeltlich abgewickelt werden und die Beteiligten insoweit von den – unten näher konkretisierten – Haftungsmilderungen profitieren.

Soweit entgeltliche Verträge zugrunde liegen, bestehen regelmäßig keine wesentlichen Unterschiede zwischen Open Source Software und proprietären Programmen. Insbesondere können nicht umfassend von vornherein „automatisch“ geringere Anforderungen an die Open Source Software gestellt werden.

---

<sup>129</sup> Vergleichbar hat es z.B. das OLG Oldenburg, ZGR 2004, 75 f. als unwirksam angesehen, einen funktionsfähigen Gebrauchtwagen an einen Endverbraucher zum Marktpreis als „Bastlerauto“ zu verkaufen.

<sup>130</sup> In diese Richtung insb. *Schiffner*, Open Source Software, S. 246 f.

## **II. Gewährleistung und Haftung in den jeweiligen Vertragsverhältnissen**

### **1. Gewährleistung und Haftung innerhalb der Überlassungsverträge**

Diejenigen Vertragsbeziehungen, die allein der Überlassung der Software zur schlichten Benutzung der Programme durch den Erwerber dienen, sind regelmäßig als Kauf oder als Schenkung einzuordnen, abhängig davon, ob die Überlassung entgeltlich oder unentgeltlich erfolgen soll.<sup>131</sup>

#### **a) Anwendung von Kaufrecht**

Dort, wo Open Source Software gegen Entgelt veräußert wird, richten sich Haftung und Gewährleistung regelmäßig nach Kaufrecht.

Hinsichtlich der Gewährleistung bedeutet das gemäß der Grundregel des § 433 Abs. 1 Satz 2 BGB: „Der Verkäufer hat dem Käufer die Sache frei von Sach- und Rechtsmängeln zu verschaffen.“ Er muss dem Käufer dafür einstehen, dass der Vertragsgegenstand keinen Sachmangel aufweist und dass keine Rechte Dritter an dem Vertragsgegenstand bestehen, die die Nutzung behindern (Rechtsmängel). Liegen Sach- oder Rechtsmängel vor, dann kann der Käufer Nacherfüllung verlangen. Er kann sich dabei grundsätzlich aussuchen, ob er den Mangel beseitigt haben möchte, oder ob ein neuer Vertragsgegenstand geliefert werden soll (§ 439 Abs. 1 BGB).

Schlägt die Nacherfüllung fehl,<sup>132</sup> dann hat der Käufer die Auswahl unter mehreren Ansprüchen. Ihm steht ein Rücktrittsrecht gemäß §§ 440, 323 BGB zu. Hierfür bedarf es keiner Fristsetzung, das heißt, der Erwerber kann nach einer fehlgeschlagenen Nacherfüllung ohne weitere Formalitäten zurücktreten und kann das gegebenenfalls bereits gezahlte Entgelt zurückverlangen. Als Alternative steht dem Erwerber das Recht zu, am Vertrag zwar festzuhalten, den Kaufpreis aber gemäß § 441 BGB zu mindern. „Mindern“ bedeutet dabei, darauf zu bestehen, weniger zu bezahlen, da der Gegenstand aufgrund des Mangels weniger Wert ist. Wurde bereits bezahlt, ist grundsätzlich die Differenz zwischen gezahltem Kaufpreis und gemindertem Kaufpreis vom Verkäufer herauszugeben.

---

<sup>131</sup> Vgl. dazu bereits ausführlich oben S. 53 ff.

<sup>132</sup> Eine Nachbesserung gilt nach dem erfolglosen zweiten Versuch als fehlgeschlagen, wenn sich nicht insbesondere aus der Art der Sache oder des Mangels oder den sonstigen Umständen etwas anderes ergibt, § 440 S. 2 BGB.

Schließlich hat der Käufer auch einen Anspruch auf Schadensersatz auf Grundlage der §§ 280, 440 BGB.<sup>133</sup> Voraussetzung für einen solchen Anspruch ist – anders als bei den zuvor genannten Ansprüchen – allerdings, dass den Verkäufer ein Verschulden, also der Vorwurf der Fahrlässigkeit oder des Vorsatzes trifft. Dies Verschulden wird jedoch vermutet. Das heißt, es ist die Aufgabe des Verkäufers, sich zu entlasten, indem er beweist, dass er den Fehler des Programms weder kannte noch hätte erkennen können.

Für die vertragliche Haftung des Verkäufers gilt, dass dieser für verursachte Schäden an sonstigen Rechtsgütern des Erwerbers einzustehen hat. Der Käufer kann Ersatz seines Schadens verlangen; Anspruchsgrundlage ist § 280 BGB.

Die Vertragsparteien können in einem bestimmten Umfang abweichende Absprachen zu diesen gesetzlichen Regelungen treffen. Dies gilt jedoch nicht unbeschränkt. Während bei individualvertraglichen Vereinbarungen zwischen Unternehmern weitreichende Gestaltungsfreiheit besteht, besteht ein deutlich engerer Spielraum bei Verwendung Allgemeiner Geschäftsbedingungen (AGB) sowie in den Fällen, in denen Verbraucher an dem Geschäft beteiligt sind. Insbesondere die folgenden sind Grenzen zu berücksichtigen:

- Ein Ausschluss der Haftung für Vorsatz ist gemäß § 276 Abs. 3 BGB grundsätzlich nicht möglich.
- Erfolgt der Vertrieb (z.B. kommerzieller Distributor) von einem Unternehmer an einen Verbraucher, können die Mängelgewährleistungsansprüche mit Ausnahme des Schadensersatzanspruchs nicht beschränkt werden, § 475 BGB.
- In Allgemeinen Geschäftsbedingungen können Haftungs- und Gewährleistungsbeschränkungen nur beschränkt vereinbart werden, vgl. insb. § 309 Nr. 7, 8 BGB.<sup>134</sup>

---

<sup>133</sup> Dieser Anspruch kann auch im Falle eines Rücktritts geltend gemacht werden. Rücktritt und Schadensersatz schließen sich nicht aus, § 325 BGB.

<sup>134</sup> Die einzelnen begrenzenden Regelungen können sich überschneiden oder ergänzen.

## b) Anwendung von Schenkungsrecht

Ist auf den Überlassungsvertrag wegen der unentgeltlichen Überlassung Schenkungsrecht anwendbar, gilt im Kern folgendes:

Für Sach- und Rechtsmängel, d.h. dafür, dass zum einen die Software die voraussetzende Beschaffenheit aufweist und zum anderen nicht Rechte Dritter einer Verwendung entgegenstehen, hat der Schenker nur dann einzustehen, wenn er den Mangel arglistig verschwiegen hat, §§ 523, 524 BGB. Das Gesetz sieht in diesen Fällen eine Schadensersatzpflicht vor. Voraussetzung der Arglist ist dabei unter anderem, dass der Schenker das Vorliegen eines Mangels kennt oder zumindest für möglich hält. Arglistiges Verschweigen setzt zudem voraus, dass eine Aufklärungspflicht besteht.<sup>135</sup> Der Beschenkte muss eine vorherige Aufklärung über den jeweiligen Mangel erwarten dürfen.

Für die Haftung im Hinblick auf die sonstigen Rechtsgüter des Erwerbers ist bei kostenloser Überlassung die Vorschrift des § 521 BGB maßgeblich. Danach hat der Schenker nur Vorsatz und grobe Fahrlässigkeit zu vertreten. Gegenüber dem allgemeinen Haftungsrecht ist der Schenker damit privilegiert. Falls es durch das Programm zu einer Verletzung der sonstigen Rechtsgüter des Erwerbers kommt, so haftet er nur, wenn er diese Rechtsverletzung wissentlich und willentlich herbeiführt oder seine Sorgfaltspflichten in besonders schwerem Maße verletzt hat.

Auch bei der Haftung und Gewährleistung nach Schenkungsrecht können abweichende vertragliche Abreden getroffen werden. Doch bestehen auch in diesen Fällen Grenzen. Insbesondere kann nicht die Haftung für vorsätzliches Verhalten ausgeschlossen werden, § 276 Abs. 3 BGB. Sollen Regelungen in Allgemeinen Geschäftsbedingungen getroffen werden, sind zudem insbesondere die Grenzen der §§ 309 Nr. 8, 307 Abs. 2 Nr. 1 BGB zu berücksichtigen, wonach unter anderem Gewährleistungsansprüche nicht insgesamt oder bezüglich einzelner Teile ausgeschlossen werden dürfen.

---

<sup>135</sup> Palandt/Putzo, § 442, Rz. 18.

## 2. Gewährleistung und Haftung innerhalb von Open Source Lizenzverträgen

Von den Verträgen, die die Überlassung der Software zum Gegenstand haben, sind diejenigen über die Einräumung der Vertriebs- und Entwicklungsrechte zu unterscheiden.<sup>136</sup>

Aus dieser Trennung der verschiedenen Verträge voneinander folgt eine erste wesentliche Schlussfolgerung für die vertragliche Haftung und Gewährleistung im Open Source Bereich: Denn für die vertragliche Haftung und Gewährleistung bedeutet das Bestehen verschiedener Vertragsverhältnisse, dass der Lizenzgeber dem Lizenznehmer im Wesentlichen (nur) für den Bestand und Erhalt der Vertriebs- und Entwicklungsrechte zu haften hat.<sup>137</sup> Die tatsächliche und rechtliche Eignung der Software für die schlichte Programmbenutzung ist hingegen eine Frage der Gewährleistung und Haftung zwischen den Vertragsparteien des entgeltlichen oder unentgeltlichen Geschäftes über die Überlassung der Software zur Benutzung und ist abhängig von den insoweit getroffenen Vereinbarungen.

### a) Unwirksamkeit bestehender Haftungs- und Gewährleistungsausschlüsse

Haftungs- und Gewährleistungsklauseln in Open Source Software Lizenzen, insbesondere diejenigen U.S.-amerikanischen Ursprungs, sehen in der Regel einen vollständigen Haftungs- und Gewährleistungsausschluss vor. Diese weitreichenden Ausschlüsse sind nach deutschem Recht grundsätzlich unwirksam.<sup>138</sup>

Bei den Open Source Lizenzen handelt es sich um sog. „Allgemeine Geschäftsbedingungen“, also um Vertragsbedingungen, die für eine Vielzahl von Verträgen vorformuliert wurden und von der einen Vertragspartei der anderen gestellt werden, § 305

---

<sup>136</sup> Vgl. dazu bereits oben S. 55.

<sup>137</sup> Daneben kommt in bestimmtem Umfang eine Haftung für die „Beschaffenheit des Immaterialgutes“ in Betracht. Bei den Entwicklungs- und Vertriebslizenzen hinsichtlich gewerblicher Schutzrechte bestehen mittlerweile feste Konturen einer Haftung für Mängel der Beschaffenheit, die sich im Vertragsprodukt niederschlagen. Auch bei softwareurheberrechtlichen Vertriebs- und Entwicklungsverträgen kann eine solche Pflicht zum Entstehenmüssen für die Tauglichkeit des Vertragsproduktes bestehen, vgl. nur *Pres*, Urheberrechtliche Softwarelizenzverträge, S. 201 f.; auch *Geissler/Pagenberg* in: *Lehmann (Hrsg.)*, Rechtsschutz und Verwertung, Kap. XIV, Rz. 51.

<sup>138</sup> Anderes gilt dort, wo die Lizenzen speziell an die deutsche Rechtsordnung angepasst wurden, so z.B. § 7 D-FSL (Deutsche Freie Software Lizenz, <<http://www.d-fsl.de>>).

BGB.<sup>139</sup> Daher sind die Vorschriften für Allgemeine Geschäftsbedingungen zu berücksichtigen; der Gesetzgeber hat diese besonderen Vorschriften geschaffen, um den in der Regel schwächeren Vertragspartner des Verwenders der Geschäftsbedingungen vorzu- einseitigen Klauseln zu schützen. Nach § 309 Nr. 8 b) aa) BGB können Gewährleistungsansprüche nicht insgesamt oder bezüglich einzelner Teile ausgeschlossen werden.<sup>140</sup> Auch die Haftung darf in Allgemeinen Geschäftsbedingungen nur sehr beschränkt begrenzt werden. So sind Ausschlüsse für grobfahrlässiges Verhalten stets und für fahrlässiges Verhalten bei Körper- und Gesundheitsschäden ausgeschlossen, § 309 Nr. 7 BGB. Nach der allgemeinen Regel in § 276 Abs. 3 BGB kann zudem die Haftung für Vorsatz überhaupt nicht ausgeschlossen werden.

Der Unwirksamkeit der Klauseln steht nicht entgegen, dass vielfach der Ausschluss nur soweit gehen soll, wie es das anwendbare Recht gestattet („TO THE EXTENT PERMITTED BY APPLICABLE LAW“; „UNLESS REQUIRED BY APPLICABLE LAW“.) Entsprechende Bestimmungen („salvatorische Klauseln“) verstoßen gegen das für allgemeine Geschäftsbedingungen geltende Transparenzgebot des § 307 Abs. 1 Satz 2 BGB.<sup>141</sup>

Folge ist, dass aufgrund der Unwirksamkeit der Klauseln der gesetzliche Gewährleistungs- und Haftungsmaßstab Anwendung findet.

## **b) Gesetzlicher Haftungs- und Gewährleistungsmaßstab bei Non-Copyleft-Lizenzen**

Open Source Lizenzen ohne Copyleft-Klausel stellen nach wohl einheitlicher Auffassung unentgeltliche Verträge dar.<sup>142</sup> Dabei ist streitig, ob eher schenkungsrechtliche Vorschriften oder aber eher leihrechtliche Vorschriften Anwendung finden. Für die Frage nach dem Umfang von Haftung und Gewährleistung hat dies jedoch grundsätzlich keine Auswirkungen, da in beiden Fällen im Wesentlichen identische Ergebnisse erzielt werden:

---

<sup>139</sup> Das ist wohl unbestritten, vgl. nur *Spindler*, VSI-Gutachten, S. 34; *ders.* in: *Spindler (Hrsg.)*, Rechtsfragen bei Open Source, Kap. C, Rz. 45; *Koch*, CR 2000, 333, 339; *Omsels* in: *Schertz/Omsels (Hrsg.)*, FS Hertin, S. 147; *Jaeger/Metzger*, Open Source Software, S. 147.

<sup>140</sup> Vgl. dazu im genauer *Jaeger/Metzger*, Open Source Software, S. 150.

<sup>141</sup> Vgl. nur *Palandt/Heinrichs*, Vor § 307, Rz. 13.

<sup>142</sup> Vgl. dazu oben S. 56 f.

Da zentraler Vertragsgegenstand des Open Source Lizenzvertrages die Einräumung von Rechten ist, steht im Mittelpunkt gewährleistungsrechtlicher Fragen das Entstehenmüssen für Rechtsmängel; Sachmängelgewährleistung spielt daneben in der Regel keine Rolle.<sup>143</sup> Für Rechtsmängel, also insbesondere dafür, dass der Lizenzgeber Inhaber der lizenzierten Nutzungsrechte ist und Rechte Dritter einer Lizenzierung nicht entgegenstehen, hat dieser sowohl nach schenkungs- als auch nach leihrechtlichen Vorschriften nur dann einzustehen, wenn er den Mangel arglistig verschwiegen hat, §§ 523 Abs. 1, 600 BGB.

Im Übrigen ist die Haftung des Lizenzgebers auf Vorsatz und grobe Fahrlässigkeit beschränkt, §§ 521, 599 BGB. Er haftet daher nur, wenn er wissentlich handelt oder aber seine Sorgfaltspflichten in besonders schwerem Maße verletzt.

### **c) Gesetzlicher Haftungs- und Gewährleistungsmaßstab bei Copyleft-Lizenzen**

Anders als bei den Non-Copyleft-Lizenzen besteht bei den Copyleft-Lizenzen keine Einigkeit darüber, ob es sich um in jeder Hinsicht unentgeltliche Verträge handelt. Vielmehr wird teilweise davon ausgegangen, dass die Einräumung der Rechte zum Vertrieb bearbeiteter Versionen und die Verpflichtung zur Open Source Lizenzierung und Quellcodeoffenlegung beim Vertrieb bearbeiteter Versionen (Copyleft) Pflichten darstellen, die in einem Gegenseitigkeitsverhältnis stehen.<sup>144</sup> Dies hat wichtige Auswirkungen für die Gewährleistung und Haftung:

Geht man von einer Unentgeltlichkeit der Copyleft-Lizenzen aus, gelten dieselben Haftungs- und Gewährleistungsregeln wie bei den Non-Copyleft-Lizenzen. Demgegenüber gilt dann, wenn man die Copyleft-Lizenzen als Geschäfte einordnet, deren Pflichten teilweise in einem Gegenseitigkeitsverhältnis stehen, im Bereich dieser Pflichten ein u.U. verschärfter Gewährleistungs- und Haftungsmaßstab. Dies würde bedeuten, dass eine Rechtsmängelgewährleistung für alle Fälle des Vertretenmüssens besteht und der Lizenzgeber für jegliches Verschulden haftet. Konkret bedeutet dies, dass der Lizenzgeber bei Einräumung der Rechte zum Vertrieb veränderter Versionen für den Bestand des

---

<sup>143</sup> Vgl. aber oben Fn. 137.

<sup>144</sup> Vgl. dazu bereits oben S. 57.

Rechts und dafür, dass Rechte Dritter nicht entgegenstehen, auch schon bei Fahrlässigkeit haftet. Hier wird man die weitere juristische Diskussion abwarten müssen.

### **3. Gewährleistung und Haftung bei Erstellung und Änderung von Freier Software**

#### **a) Erstellung von Software im Kundenauftrag**

Wird Software gegen Entgelt im Kundenauftrag erstellt, ist Werkvertragsrecht oder nach anderer Auffassung im Wesentlichen Kaufvertragsrecht anwendbar.<sup>145</sup>

Unabhängig davon, ob es sich um einen Kauf- oder Werkvertrag handelt, kann der Erwerber bei einem mangelhaften Werk zunächst Nacherfüllung verlangen, wobei Unterschiede in der Ausübung je nach vertragstypologischer Einordnung der Verträge bestehen.<sup>146</sup> Scheitert die Nacherfüllung, stehen dem Besteller Minderung, Rücktritt und – bei Verschulden der anderen Vertragspartei – Schadensersatzansprüche zu. Ob darüber hinaus ein Recht des Kunden besteht, den Mangel selbst zu beseitigen und Ersatz der hierzu erforderlichen Aufwendungen zu verlangen, ist davon abhängig, ob man den Vertrag als Werkvertrag ansieht und nicht als einen Vertrag, auf den Kaufrecht Anwendung findet. Hier sollten aufgrund der bestehenden Unsicherheiten präzise Regelungen in den Vertrag aufgenommen werden.

Für die vertragliche Haftung des Verkäufers gilt, dass dieser für schuldhaft verursachte Schäden an sonstigen Rechtsgütern des Erwerbers einzustehen hat. Eine fahrlässige Herbeiführung des Schadens ist hinreichend.

Die Vertragsparteien können abweichende Absprachen zu diesen gesetzlichen Regelungen treffen. Grenzen bilden insoweit die zwingenden gesetzlichen Regelungen, insbesondere die verbraucherschützenden Vorschriften in den §§ 305 ff. BGB, die lediglich begrenzt Beschränkungen der Gewährleistung zulassen.

---

<sup>145</sup> Vgl. dazu bereits oben S. 58 f.

<sup>146</sup> In welcher Form diese geleistet wird – Beseitigung des Mangels oder Lieferung eines neuen mangelfreien Werks – darf bei einem Kaufvertrag der Käufer, bei einem Werkvertrag hingegen der Unternehmer entscheiden, vgl. §§ 439, 635.

## **b) Anpassung von Open Source Software**

Auch bei der entgeltlichen Anpassung von Software an die spezifischen Kundenbedürfnisse besteht eine Gewährleistung und Haftung nach werk- oder kaufvertraglichen Grundsätzen, abhängig von der jeweiligen Einordnung des Geschäftes.

Besondere Probleme können hier hinsichtlich der Frage entstehen, ob Mängel der Anpassungsleistung das Recht auch zum Rücktritt von der Lieferung der Software geben (und umgekehrt), wenn beides „aus einer Hand“ geliefert wird. Mit anderen Worten: Kann derjenige, der die Lieferung einer Software und deren Anpassung in Auftrag gibt, auch die Software zurückgeben, wenn deren Anpassung scheitert? Kann auch die Anpassungsleistung rückabgewickelt werden, wenn sich die Software als mangelhaft erweist? Hier wird es vielfach von den Verhältnissen des Einzelfalls abhängen, ob es sich um einen wirtschaftlich einheitlichen Vertragsgegenstand handelt (Gegenstand des Vertrages ist die „angepasste Software“); dann ist ohne weiteres ein Rücktritt vom gesamten Vertrag denkbar. Handelt es sich um unterschiedliche Vertragsgegenstände, kommt es darauf an, ob der Gläubiger an der teilweisen Leistung „kein Interesse hat“, § 323 Abs. 5 BGB. Auch dann ist ein Gesamtrücktritt möglich.

## **III. Außervertragliche Haftung**

Eine Haftung kommt nicht nur innerhalb der bestehenden vertraglichen Beziehungen in Betracht. Daneben sieht das Gesetz vielmehr auch außervertragliche Haftungstatbestände vor, die unter Umständen einschlägig sein können. Zu beachten sind hier insbesondere die verschuldensunabhängige Haftung nach dem Produkthaftungsgesetz und die verschuldensabhängige Haftung nach dem allgemeinen Deliktsrecht.

### **1. Produkthaftung**

Bei der Produkthaftung handelt es sich um eine verschuldensunabhängige Garantief Haftung.

#### **a) Wer haftet?**

Die Haftung nach dem ProdHaftG trifft gem. § 1 Abs. 1 S.1 ProdHaftG den „Hersteller“. Hersteller ist gemäß § 4 ProdHaftG unter anderem derjenige, der das Endprodukt, einen Grundstoff oder ein Teilprodukt hergestellt hat. Als Hersteller gilt auch jeder, der sich

durch das Anbringen seines Namens, seiner Marke oder eines anderen unterscheidungskräftigen Kennzeichens als Hersteller ausgibt. Als Hersteller kommt also neben dem Programmierer insbesondere auch der Distributor in Frage.

Die Haftung ist gemäß § 1 Abs. 2 Nr. 3 ProdHaftG ausgeschlossen, wenn der Hersteller die Software „weder für den Verkauf oder eine andere Form des Vertriebs mit wirtschaftlichem Zweck hergestellt noch im Rahmen seiner beruflichen Tätigkeit hergestellt oder vertrieben hat.“ Ob dies beim Vertrieb freier Programme der Fall ist, kann nicht pauschal festgestellt werden. Wird von einem Distributor eine Standarddistribution für den Einzelhandel hergestellt, so handelt es sich um eine Herstellung mit wirtschaftlicher Zielsetzung und im Rahmen der beruflichen Tätigkeit. In diesem Fall bestehen somit grundsätzlich direkte Ansprüche nach dem ProdHaftG gegen den Hersteller. Wird Code in der Freizeit programmiert, besteht hingegen keine Haftung nach dem Produkthaftungsgesetz. Zwischen diesen eindeutigen Fällen sind jedoch eine Reihe weiterer Konstellationen denkbar, bei denen es jeweils einer Prüfung im Einzelfall bedarf.<sup>147</sup>

#### **b) Wofür wird gehaftet?**

Der Hersteller haftet verschuldensunabhängig für „Fehler“ im Sinne des § 3 ProdHaftG. Nach dieser Vorschrift liegt ein Fehler vor, wenn ein Produkt nicht die Sicherheit bietet, die unter Berücksichtigung aller Umstände, insbesondere seiner Darbietung, des Gebrauchs, mit dem billigerweise gerechnet werden kann, und des Zeitpunkts, in dem es in den Verkehr gebracht wurde, berechtigterweise erwartet werden kann. Ausgeschlossen sind nach § 1 Abs. 2 Nr. 4 ProdHaftG solche Fehler, die nach dem Stand der Technik im Zeitpunkt des Inverkehrbringens nicht erkannt werden konnten.

Liegt ein Fehler vor, begründet das Produkthaftungsgesetz eine Haftung nur für Personenschäden und solche Sachschäden, die an einer anderen Sache als dem fehlerhaften Produkt eintreten. Für Fehler des Produkts selbst wird hingegen nach dem Produkthaftungsgesetz nicht gehaftet; auch eine grundsätzliche Haftung für Vermögensschäden besteht nicht. Keine Haftung besteht nach diesem Gesetz, das in erster Linie den Verbraucher schützen soll, zudem für solche Sachen, die nicht privat verwendet werden,

---

<sup>147</sup> Vgl. dazu weiterführend *Jaeger/Metzger*, Open Source Software, S. 153.

vgl. § 1 Abs. 1 ProdHaftG. Unternehmerischen Schadensersatzforderungen wegen Sachschäden sieht sich der Hersteller insoweit auf der Grundlage dieses Gesetzes nicht ausgesetzt.

### c) Umfang der Haftung

Grundsätzlich hat der Hersteller den gesamten Schaden zu ersetzen. Bei Sachschäden besteht gemäß § 11 ProdHaftG jedoch eine Selbstbeteiligung von EUR 500,00. Bei Personenschäden besteht eine solche Selbstbeteiligung nicht. Hier gibt es jedoch einen Haftungshöchstbetrag: Sind Personenschäden durch ein Produkt oder gleiche Produkte mit demselben Fehler verursacht worden, so haftet der Ersatzpflichtige nur bis zu einem Höchstbetrag von 85 Millionen Euro, vgl. § 10 ProdHaftG.

## 2. Verschuldensabhängige außervertragliche Haftung

Eine außervertragliche Haftung kommt nicht nur aufgrund der besonderen Regelungen des Produkthaftungsgesetzes in Betracht. Daneben sieht das Gesetz vielmehr auch allgemeine außervertragliche Haftungstatbestände (Deliktstatbestände) vor. Da das Produkthaftungsgesetz bei Sachschäden nur private Verbraucher schützt und da es eine Selbstbeteiligung bei Sachbeschädigung vorsieht, behalten diese ihre eigene Relevanz.<sup>148</sup>

Neben der vorsätzlichen sittenwidrigen Schädigung (§ 826 BGB), deren Voraussetzungen allerdings nur selten vorliegen dürften, ist für den Softwarebereich insbesondere die Regelung des § 823 BGB interessant. Nach dieser Vorschrift ist derjenige zum Schadensersatz verpflichtet, der vorsätzlich oder fahrlässig Leben, Körper, Eigentum oder sonstige ähnlich „absolut“ geschützte Rechtspositionen widerrechtlich verletzt. Ob dabei der Datenbestand auf dem Rechner eine solche absolut geschützte Rechtsposition darstellt ist umstritten.<sup>149</sup>

---

<sup>148</sup> Grundsätzlich spielen entsprechende Ansprüche in der Praxis bisher allerdings noch eine untergeordnete Rolle, vgl. dazu *Schiffner*, Open Source Software, S. 253 f. m.w.N.

<sup>149</sup> Bejahend OLG Karlsruhe, NJW 1996, 200, 201; vgl. dazu auch *Palandt/Sprau*, § 823. Rz. 8.

### a) Haftung des Programmierers der Software

Eine Haftung kommt grundsätzlich nur dort in Betracht, wo die Verletzung der geschützten Rechtsposition vorsätzlich oder fahrlässig erfolgte. Insoweit wird man bei Programmierern den zu beachtenden Sorgfaltsmaßstab vielfach nicht zu hoch ansetzen dürfen. Die einzelnen Programmierer leisten ihre Entwicklungsbeiträge jeweils nicht mit dem Ziel, dass anschließend ein fertiges Endprodukt vorliegt (zu einem solchen wird es vielmehr erst, indem Versionskennziffern für stabile Versionen vergeben werden und/oder Distributoren und Private die Software als Endanwenderprodukte vermarkten). Innerhalb von Entwicklungszyklen muss aber die Präsentation unfertiger Lösungen vielfach in Kauf genommen werden, schon damit Dritte weitere Änderungen an der Software vornehmen können.

### b) Haftung des Distributors der Software

Deutlich weitreichender stellt sich die außervertragliche Haftung der Distributoren Freier Software dar. Diese stellen in der Regel ein fertiges Endprodukt zur Verfügung.

Hier kommt dort, wo das Produkt in industrieller Weise gefertigt und vertrieben wird, eine Haftung nach den von der Rechtsprechung entwickelten Grundsätzen der sog. Produzentenhaftung in Betracht.<sup>150</sup> Im Rahmen einer solchen Produzentenhaftung wird in verschiedener Hinsicht die Beweisführung des Verletzten erleichtert, da dieser oft nicht in der Lage ist, haftungsbegründende Umstände zu beweisen, die im Gefahren- und Verantwortungsbereich der Hersteller liegen. Gelangt ein Produkt auf den Markt und verletzt durch seine Fehlerhaftigkeit den Anspruchsteller (beides muss dieser beweisen), dann muss der Produzent beweisen, dass im Bereich seines Unternehmens die erforderliche Sorgfalt aufgewendet worden ist. Grund dafür ist, dass Vorgänge und Organisation auf Seiten des Produzenten aufgrund fehlender Einblicke durch den Anspruchsteller ansonsten zumeist nicht bewiesen werden könnten.<sup>151</sup>

Erfasst werden dabei

- Konstruktionsfehler

<sup>150</sup> Diese ist zu unterscheiden von der oben beschriebenen „Produkthaftung“; vgl. zur Produzentenhaftung auch *Schiffner*, Open Source Software, S. 253.

<sup>151</sup> Vgl. zur Produzentenhaftung die Übersicht bei *Medicus*, Bürgerliches Recht, Rz. 650 ff.

- Fertigungs- oder Kontrollfehler
- Anleitungsfehler (Instruktionsfehler)<sup>152</sup>

Insgesamt dürfen dabei aber die Sorgfaltspflichten nicht zu weitgehend festgelegt werden<sup>153</sup> und ist ein eventuelles Mitverschulden des Geschädigten stets zu berücksichtigen. Dieses Mitverschulden kann im Grenzfall derart hoch zu bewerten sein, dass eine Haftung des Distributors ausscheidet. Dies betrifft namentlich Fälle der unterlassenen Datensicherung.<sup>154</sup>

### c) Haftung von Core-Teams und Maintainern

Bislang kaum beachtet wurden Fragen der außervertraglichen Haftung von Core-Teams und Maintainern. Diese Personen oder Gruppen treffen vielfach für die Vermarktung der jeweiligen Software weitreichende Entscheidungen, insbesondere die Auswahl der Entwicklungsbeiträge und die Freigabe als „stabile Version“ bzw. die Vergabe von Versionsnummern.<sup>155</sup>

Gerade dort, wo „stabile Versionen“ herausgegeben werden, kommen jedoch Ansatzpunkte für eine außervertragliche Haftung der jeweiligen Organisationseinheiten und ihrer Mitglieder in Betracht. Der Fall ist den Montage- bzw. Assembling-Systemen vergleichbar, bei denen verschiedenen Lieferanten zusammen ein Produkt erstellen, die Aufsicht aber beim eigentlichen Hersteller verbleibt.<sup>156</sup> Dabei kommen demjenigen, der die „Zusammensetzung“ der Produkte vornimmt zwar größere Schadensabwendungspflichten als dem reinen Vertreibenden zu, nicht jedoch gleich strenge wie dem Alleinhersteller.<sup>157</sup> Allerdings können die Haftungserleichterungen aus der vertraglichen Haftung – etwa aus dem Schenkungsrecht – auch auf die außervertragliche Haftung durch-

---

<sup>152</sup> Können die Gefahren des Produkts durch Gebrauchshinweise oder Warnungen entschärft werden, ist der Hersteller gehalten, solche Hinweise zu geben. Im Softwarebereich kann dies etwa ein Hinweis auf Ungenauigkeiten bei der Rundung sein, wenn diese beim Einsatz der Software zu bestimmten Zwecken zu Risiken führen können.

<sup>153</sup> *Schiffner*, Open Source Software, S. 254.

<sup>154</sup> Im gewerblichen Anwenderbereich gehört es zu den vorauszusetzenden Selbstverständlichkeiten, dass eine zuverlässige, zeitnahe und umfassende Datenroutine die Sicherung gewährleistet. Daher hat z.B. OLG Hamm, JurPC Web-Dok. 165/2004, <<http://www.jurpc.de/rechtspr/20040165.htm>>, zu Recht einen Schadensersatzanspruch wegen überwiegenden Mitverschuldens versagt, weil eine entsprechende Routine fehlte.

<sup>155</sup> Wer jeweils zur Entscheidung über die „Reife“ eines Produktes entscheiden darf, ist innerhalb der verschiedenen Projekte unterschiedlich geregelt.

<sup>156</sup> *Spindler* in: *Spindler (Hrsg.)*, Rechtsfragen bei Open Source, Kap. E, Rz. 25.

<sup>157</sup> *Palandt/Thomas*, § 823, Rz. 216.

schlagen. Inwieweit dies für Herausgeber von „stabilen Versionen“ der Fall ist, lässt sich nicht pauschal vorhersehen.

#### **IV. *Praktische Hinweise: Was ist bei Programmierung, Vertrieb und Verwendung von Open Source Software zu berücksichtigen, um unnötige Gewährleistungs- und Haftungsrisiken auszuschließen?***

Bei Programmierung, Vertrieb und Verwendung von Open Source Software können sowohl Fragen vertraglicher Gewährleistung und vertraglicher Haftung als auch Fragen außervertraglicher Haftung auftauchen. Um sich einen Überblick über das jeweilige Risiko zu verschaffen, bietet es sich an, die folgenden Merkposten zu berücksichtigen:

*In welchem Umfang können vertragliche Gewährleistungspflichten auftreten?* – Die Frage nach der Gewährleistung, also dem Entstehenmüssen für die Vertragsgemäßheit der Leistung, stellt sich sowohl dort, wo Freie Software überlassen oder im Kundenauftrag angepasst oder erstellt wird, als auch dort, wo Nutzungsrechte an der Software eingeräumt werden, und schließlich auch dort, wo Service oder Support geleistet werden. Der Umfang der Gewährleistungspflichten bestimmt sich dabei unter anderem danach, ob eine Gegenleistung verlangt werden soll oder nicht. Er ist zudem davon abhängig, welche Beschaffenheit des Vertragsproduktes vereinbart wurde: Bei einer „Entwicklerversion“ kann kaum dieselbe Laufstabilität verlangt werden wie bei „Endanwenderversionen“.

*Können vertragliche Gewährleistungspflichten vertraglich modifiziert oder abbedungen werden?* – Abweichend von den gesetzlichen Gewährleistungsregeln können die Parteien innerhalb bestimmter Grenzen Abreden über den Umfang der Gewährleistung treffen. Entsprechende Abreden sind insbesondere dort dringend zu empfehlen, wo der Umfang der bestehenden Pflichten nach wie umstritten ist; dies gilt insbesondere für die individuelle Herstellung und Anpassung von Open Source Software im Kundenauftrag. Insgesamt ist zu berücksichtigen, dass eine Überschreitung der Grenzen, innerhalb derer Abreden getroffen werden können, zu einer Nichtigkeit entsprechender Gewährleistungsklauseln führt. An die Stelle der nichtigen Klausel tritt sodann die gesetzliche

Regelung. Nichtig sind insbesondere regelmäßig die in den Open Source Lizenzen U.S.-amerikanischen Ursprungs enthaltenen vollständigen Gewährleistungsausschlüsse.

*Welcher vertragliche Haftungsmaßstab ist zu berücksichtigen?* – Ebenso wie der vertragliche Gewährleistungsmaßstab ist auch der Maßstab der vertraglichen Haftung abhängig von dem zwischen den Vertragsparteien geschlossenen Geschäft. Dabei gilt auch hier: Bei unentgeltlichen Geschäften sind die Anforderungen tendenziell deutlich geringer. Und ebenso wie hinsichtlich der Gewährleistung können auch hinsichtlich der Haftung innerhalb bestimmter Grenzen Abreden zwischen den Vertragsparteien getroffen werden.

*In welchem Umfang bestehen außervertragliche Haftungsrisiken?* – Stets ist zu berücksichtigen, dass über die vertragliche Haftung hinaus auch eine außervertragliche Haftung eintreten kann. Außervertragliche Haftung kann gegenüber jedermann bestehen, nicht nur gegenüber solchen Personen, mit denen Verträge geschlossen wurden. Risiken bestehen dabei vor allem in den Bereichen der Produkt- und Produzentenhaftung. Davon sind allerdings weniger die Urheber als vielmehr Distributoren und beschränkt auch Mitglieder von Core-Teams etc. betroffen.

## **Teil 6. Rechtliche Fragen bei der Gestaltung von Open Source Projekten**

### ***I. Entwicklerzusammenschlüsse***

Da es bei der Entwicklung und Verwertung von Open Source Software keineswegs erforderlich ist, dass jedermann als Einzelperson handelt, sind in diesem Zusammenhang zahlreiche Formen engerer Zusammenarbeit beim Vertrieb und bei der Entwicklung Freier Software zu erkennen. Beispielsweise kann die Software von mehreren Personen nach Absprache gemeinsam verändert werden oder innerhalb unternehmerischer Strukturen weiterentwickelt und vertrieben. Zugleich verfügen Open Source Projekte in der Regel über bestimmte Strukturen, die das „Rückgrat“ des jeweiligen Projektes bilden, indem dort die Entwicklung in gewisse Bahnen gelenkt wird, offizielle Versionen

der Software bestimmt werden und Zielsetzungen definiert. Auch hier bestehen regelmäßig bestimmte Formen gemeinschaftlichen Zusammenwirkens. Alle diese Formen der Zusammenarbeit führen zu spezifischen rechtlichen Fragestellungen, sowohl im Hinblick auf die Verbindung der verschiedenen Beteiligten untereinander als auch im Hinblick auf deren (gemeinschaftliche) Stellung im Rechtsverkehr.

## **1. Zusammenschlüsse zur Schaffung und Verwertung Freier Software**

Open Source Software darf von jedermann verändert werden. Freie Software kann daher unzählige und im Vorwege nicht bereits festgelegte öffentliche Bearbeitungszyklen durchlaufen. Daneben sind jedoch auch zielgerichtete Verbindungen denkbar, innerhalb derer die Software weiterentwickelt wird, ohne dass diese in jedem einzelnen Entwicklungsschritt jeweils veröffentlicht wird. Zu nennen sind hier insbesondere die Miturheberschaft, die Urheberschaft an verbundenen Werken, sowie die Entwicklung von Software in Unternehmensstrukturen, insbesondere durch Angestellte des Unternehmens. Daraus resultieren bestimmte Folgen für die rechtlichen Beziehungen der Beteiligten untereinander:

### **a) Miturheber**

Miturheberschaft liegt vor, wenn eine einheitliche Schöpfung gemeinschaftlich durch gewollte Zusammenarbeit der Miturheber entsteht.<sup>158</sup> Die Urheber müssen also das einheitliche Werk nach gemeinsamer Zielsetzung geschaffen haben. Dabei genügt für Miturheberschaft auch, wenn die verschiedenen Beteiligten auf unterschiedlichen Stufen der Softwareentwicklung tätig werden, etwa einzelne die Vorstufen des Programms (Datenflussplan etc.) erstellen und andere die Codierung vornehmen. Wird gemeinschaftlich eine bestehende Software fortentwickelt, was bei Open Source Software sehr häufig passiert, liegt eine Bearbeitung in Miturheberschaft vor.

Für das Verhältnis der Miturheber untereinander gelten besondere Regelungen. Insbesondere müssen alle Miturheber gemeinsam die Entscheidung zur Veröffentlichung und zur Verwertung treffen, wobei der einzelne Miturheber allerdings seine Zustimmung nicht wider Treu und Glauben versagen darf, § 8 Abs. 2 UrhG. Das bedeutet, dass

---

<sup>158</sup> Vgl. dazu im Einzelnen *Dreier/Schulze*, Urheberrechtsgesetz, § 8 Rz. 2 ff.

nicht einzelne Urheber die Entscheidung treffen dürfen, eine Software Open Source zu stellen oder die Bearbeitung einer Open Source Software zu veröffentlichen.

### **b) Urheber verbundener Werke**

Urheber verbundener Werke sind solche, die ihre jeweiligen Werke zur gemeinschaftlichen Veröffentlichung verbinden, § 9 UrhG. Voneinander grundsätzlich unabhängige Programme werden danach bewusst zu einer Verwertungseinheit verbunden.

Nach ganz überwiegender Ansicht besteht zwischen den Urhebern verbundener Werke eine Gesellschaft bürgerlichen Rechts.<sup>159</sup> Dies hat unter anderem zur Folge, dass alle Entscheidungen, welche innerhalb des Zwecks der Werkverbindung liegen, gemeinschaftlich zu treffen sind und der vorherigen Zustimmung aller Urheber der verbundenen Werke bedürfen. Dies gilt sowohl für den Abschluss als auch für die Beendigung von Verwertungsverträgen und damit auch für die Frage der Freigabe unter einer Open Source Lizenz.

In der Open Source Praxis sind Werkverbindungen selten. Insbesondere ist nicht davon auszugehen, dass alle Urheber der verschiedenen Programme eines modular aufgebauten Betriebssystems ihre Werke zur gemeinschaftlichen Verwertung verbinden. Hier fehlt es regelmäßig an entsprechenden Abreden. Eine Werkverbindung wäre vielfach auch nicht sinnvoll, sollen im Open Source Bereich die einzelnen Module doch grundsätzlich eigenständig sein. Dies wäre bei Werkverbindungen nicht gewährleistet: Läge eine Werkverbindung vor, so hätte dies zur Folge, dass z.B. ein Wechsel der Lizenz, insb. ein Vertrieb unter einer neueren Lizenzversion, nur dann zulässig wäre, wenn sämtliche Urheber aller Module dem Wechsel zustimmen oder von vornherein eine „any-later-version“-Klausel verabredet haben. Demgegenüber kann bei Unabhängigkeit der einzelnen Programme jeder für das eigene Programm selbst entscheiden.

---

<sup>159</sup> Ausführlich dazu *Dreier/Schulze*, Urheberrechtsgesetz, § 9 Rz. 17 ff.

### **c) Angestellte Urheber**

§ 69b UrhG trifft die Regelung, dass bei Software, die im Dienst- oder Arbeitsverhältnis geschaffen wurde, der Arbeitgeber zur Ausübung aller vermögensrechtlichen Befugnisse berechtigt ist, sofern nichts anderes vereinbart wurde.

#### **(1) Bedeutung für neu geschaffene Software**

§ 69b UrhG hat für neu geschaffene Software zur Folge, dass grundsätzlich dem Arbeitgeber die Entscheidung darüber obliegt, ob die Software „Open Source“ vertrieben werden darf oder nicht. Der angestellte Urheber darf seine im Dienst- oder Arbeitsverhältnis geschaffene Software also regelmäßig nicht selbst unter eine Open Source Lizenz stellen. Dies hat Konsequenzen in zweierlei Richtung:

Will der angestellte Urheber seine Software Open Source lizenzieren, muss er entsprechende Abreden mit seinem Arbeitgeber treffen: Insoweit sollten die Parteien bereits im Vorfeld eine Vereinbarung treffen, in der festgelegt wird, dass der Arbeitgeber an der zu entwickelnden Software nicht die vermögensrechtlichen Befugnisse nach § 69b UrhG erhalten soll. Kann im Vorwege der Entwicklung noch keine entsprechende Vereinbarung getroffen werden und erhält der Arbeitgeber daher aufgrund der gesetzlichen Regelung zunächst die vermögensrechtlichen Befugnisse nach § 69b UrhG, dann kann anschließend vereinbart werden, dass der Arbeitgeber diese an den Programmierer zurück überträgt; letzterer kann daraufhin die Software Open Source lizenzieren. Möglich ist schließlich auch, dass der Arbeitgeber in Absprache mit dem Programmierer selbst von seinen ihm zustehenden Rechten Gebrauch macht und die Software unter einer Open Source Lizenz lizenziert.

Will der Arbeitgeber die Software „Open Source“ lizenzieren, kann er dies grundsätzlich ohne Weiteres tun. Ausnahmen bestehen im Wesentlichen allein dort, wo abweichende Abreden mit dem Urheber oder Dritten getroffen wurden. Hinzu kommt, dass in absoluten Ausnahmefällen die Urheberpersönlichkeitsrechte des Urhebers eine Grenze bilden können.

#### **(2) Bedeutung für die Weiterentwicklung von Open Source Software**

Bei der Weiterentwicklung von Open Source Software ist zu unterscheiden: Steht die Software unter keinem Copyleft, kann der Bearbeiter frei entscheiden, wie die Software lizenziert werden soll. § 69b UrhG überträgt diese Entscheidungsgewalt vom angestellten Urheber auf dessen Arbeitgeber. Damit steht diesem grundsätzlich die Entscheidung zu. Es gelten dieselben Grundsätze wie bei der Neuerstellung von Software.

Bei den Copyleft-Lizenzen stellt sich das Problem, wie die Bindung durch das Copyleft mit der umfassenden gesetzlichen Einräumung der vermögenswerten Befugnisse an den Arbeitgeber in § 69b UrhG in Einklang zu bringen ist. Hier ist die These aufgestellt worden, zwischen beiden Voraussetzungen bestehe ein unauflöslicher Gegensatz.<sup>160</sup> Doch handelt es sich letztlich allein um ein Scheinproblem: Mit der Bearbeitung erwirbt der angestellte Urheber ein Bearbeiterurheberrecht. Die vermögenswerten Befugnisse gehen auf den Arbeitgeber über. Weder der Arbeitnehmer noch der Arbeitgeber werden allein durch die Bearbeitung zur Veröffentlichung derselben gezwungen, denn eine Pflicht zur Veröffentlichung kennen die Lizenzen nicht; daher verstößt der Arbeitnehmer durch den gesetzlichen Übergang der Rechte auch nicht gegen die Pflichten aus der Open Source Lizenz. Soll das bearbeitete Programm den unternehmensinternen Bereich verlassen – was keineswegs immer der Fall sein muss – dann muss es selbstverständlich von dem Inhaber der entsprechenden Befugnisse, also dem Arbeitgeber, unter die Copyleft-Lizenz gestellt werden. Der Arbeitgeber selbst ist nämlich an das Copyleft gebunden, denn eine Vervielfältigung und ein Vertrieb der Bearbeitung berühren auch die Rechte des Originalurhebers und der Originalurheber hat seine Rechte nur unter der Bedingung der Open Source Lizenzierung beim Vertrieb von Bearbeitungen eingeräumt.

## 2. Zusammenschlüsse zur Koordination von Open Source Projekten

Obwohl Open Source von jedermann verändert werden darf, erfolgt Open Source Entwicklung in der Regel nicht vollständig unkoordiniert. Insbesondere die Veröffentlichung der Software unter einer neuen „offiziellen“ Versionsziffer und die verantwortliche Betreuung der Entwicklung ganzer Programme oder einzelner Module obliegen regelmäßig bestimmten zentralen „Einheiten“ oder einzelnen Verantwortlichen.

---

<sup>160</sup> So insbesondere *Deike*, CR 2003, 9, 13, der unter anderem die kaum haltbare These aufstellt, das Arbeitsentgelt eines Programmierers stelle eine Lizenzgebühr dar.

Die Organisation und rechtliche Strukturierung der zentralen Einheiten ist innerhalb der unterschiedlichen Open Source Projekte sehr verschieden ausgestaltet; sie kann beispielsweise in der Hand von einzelnen Unternehmen liegen,<sup>161</sup> auf spezielle Vereinigungen übertragen worden sein,<sup>162</sup> oder aber innerhalb schlichter Entwicklergemeinschaften erfolgen.<sup>163</sup> Während bei den Unternehmen und Vereinigungen die rechtliche Organisationsform in der Regel ausdrücklich gewählt wurde und leicht erfassbar ist, stellen sich bei den schlichten Entwicklergemeinschaften Probleme in der Abgrenzung, ob es sich (noch) um rein faktische Zusammenschlüsse handelt oder aber (bereits) um eine Gesellschaft bürgerlichen Rechts i.S.d. § 705 BGB oder sonstige gesellschaftsähnliche Strukturen. Voraussetzung einer Gesellschaft ist, dass die Gesellschafter vertraglich einen gemeinsamen Zweck vereinbaren sowie eine Pflicht zur Förderung des Vertragszwecks durch die Gesellschafter.<sup>164</sup> Ein entsprechender Vertrag braucht dabei nicht ausdrücklich abgeschlossen zu werden, sondern er kann auch durch schlüssiges Verhalten zustande kommen. Ob entsprechende Vereinbarungen vorliegen ist dabei jeweils vom Einzelfall abhängig: Ein gemeinsamer Zweck liegt mit der Koordinierung des jeweiligen Projekts regelmäßig vor, schwierig ist jedoch die Frage, ob bestimmte Förderungspflichten bestehen. Anhaltspunkte dafür liegen zum Beispiel in einer festen Aufgabenverteilung zwischen den Mitgliedern.

Für die Praxis sehr interessant, bisher aber noch wenig beleuchtet ist die Frage, inwieweit die jeweiligen Organisationseinheiten für ihre Tätigkeit Dritten gegenüber haftbar sein können. Gerade dort, wo „stabile Versionen“ herausgegeben werden, kommen Ansatzpunkte für eine außervertragliche Haftung der jeweiligen Organisationseinheiten und ihrer Mitglieder in Betracht.<sup>165</sup>

---

161 So z.B. im Falle der Open Source Datenbank MySQL, <<http://www.mysql.com>>.

162 So etwa die Verwaltung von Apache durch die Apache Foundation, <<http://www.apache.org/foundation/contributing.html>>.

163 Vgl. z.B. netfilter/iptables, <<http://www.netfilter.org/about.html#coreteam>>.

164 Vgl. dazu einführend *Palandt/Sprau*, § 705, Rz. 1 f. m.w.N.

165 Vgl. dazu bereits oben S. 78.

## **II. Von Unternehmen initiierte Open Source Projekte**

### **1. Rechte**

Sollen Open Source Projekte von Unternehmen initiiert und begleitet werden, ist zunächst innerhalb des Unternehmens exakt zu evaluieren, ob der zur Lizenzierung vorgesehene Code tatsächlich „Open Source“ lizenziert werden darf. Voraussetzung dafür ist, dass das Unternehmen Inhaber der ausschließlichen Nutzungsrechte ist und keine Rechte Dritter entgegenstehen. Für die Prüfung, ob das Unternehmen über die erforderlichen Nutzungsrechte verfügt, ist dabei zu berücksichtigen, dass – wie soeben bereits ausführlicher dargestellt – der Arbeitgeber regelmäßig die vermögensrechtlichen Befugnisse an solcher Software, die Arbeitnehmer im Rahmen ihrer Tätigkeit geschaffen haben aufgrund gesetzlicher Regelung erhält. Eine vergleichbare Regelung existiert allerdings nicht für Software die von Dritten (z.B. „Freelancern“) im Auftrag des Unternehmens erstellt wurde. Hier muss sich das Unternehmen die ausschließlichen Rechte zur Vervielfältigung, zur Verbreitung, zur Umarbeitung und zur öffentlichen Zugänglichmachung ausdrücklich einräumen lassen. Hat eine solche Rechtseinräumung nicht stattgefunden, darf eine Open Source Lizenzierung nicht erfolgen.

Ist geklärt, dass das Unternehmen die notwendigen Nutzungsrechte erworben hat, ist weiter zu prüfen, ob über diese Rechte nicht bereits Rechtsgeschäfte abgeschlossen wurden, die einer Open Source Lizenzierung entgegenstehen. Dies kann etwa der Fall sein, wenn über bestimmte ausschließliche Befugnisse bereits abweichend verfügt wurde oder wenn etwa mit Dritten Geheimhaltungsabreden etc. getroffen wurden.

Neben der Frage, ob die urheberrechtlichen Nutzungsrechte an der Software vollständig vorliegen, ist stets auch zu beachten, ob für die Herstellung, den Vertrieb oder den Gebrauch der Software eine Patentlizenz Dritter erforderlich ist, die einer Open Source Lizenzierung entgegenstehen könnte.

## 2. Auswahl der Lizenz

In einem weiteren Schritt ist die geeignete Lizenz auszuwählen. Hier sind insbesondere die bereits oben ausführlich dargestellten und allgemein gültigen Gesichtspunkte der Lizenzwahl zu beachten.<sup>166</sup>

## 3. Auswahl der Koordinationseinheiten

Wesentlich im Rahmen der Initiierung und Betreuung von Open Source Projekten durch Unternehmen ist schließlich auch die Auswahl der „Koordinationseinheiten“ für die weitere Softwareentwicklung und Vermarktung.

Soll die Software im Wege des Dual Licensing durch das Unternehmen über den gesamten Entwicklungsprozess der Software auch proprietär vermarktet werden, wie dies etwa im Falle von MySQL geschieht, wird die Koordinierung regelmäßig in der Hand des Unternehmens selbst liegen. Demgegenüber kann sich in anderen Fällen eine Ausgliederung auf eine rechtlich selbständige Vereinigung (z.B. Verein oder Gesellschaft) anbieten. Die damit verbundenen Einbußen an Kontrolle über die Software durch das Unternehmen selbst sind abzuwiegen mit der möglicherweise deutlich höheren Akzeptanz des Projektes bei den potentiellen Entwicklern und der aus einer Ausgliederung folgenden Verlagerung von Haftungsrisiken auf den neuen Rechtsträger.<sup>167</sup>

### III. *Nachverfolgbarkeit von Entwicklungsbeiträgen*

Hinsichtlich der Frage, wer für einen bestimmten Teil des Codes verantwortlich ist, kann der Einsatz von Versionskontrollsystemen (z.B. CVS) sinnvoll sein. Damit ist sichergestellt, dass die verschiedenen Versionen der Software rückverfolgbar bleiben. Inwieweit dabei auch ein Rückschluss auf einzelne Personen oder Unternehmen möglich ist, ist davon abhängig, inwieweit die jeweiligen Versionskontrollsysteme die anonyme oder pseudonyme Übertragung zulassen.

Versionskontrollsysteme sind aus rechtlicher Sicht insbesondere für die Frage der Beweisführung interessant. Mit ihnen kann unter Umständen nachgewiesen werden, dass

---

<sup>166</sup> Vgl. oben S. 51 f.

<sup>167</sup> Zu den Haftungsrisiken oben S. 78.

eine bestimmte Person oder ein bestimmtes Unternehmen Codebestandteile bewusst und freiwillig in die Software eingefügt hat. Die Gefahr der späteren Geltendmachung von Ansprüchen durch solche Personen oder Unternehmen unter der Behauptung, der Code sei von Dritten ohne Zustimmung der Rechtsinhaber veröffentlicht worden, kann damit von vornherein deutlich gesenkt werden. Umgekehrt ist allerdings zu berücksichtigen, dass weitreichende Pflichten zur Identitätsoffenlegung zu einer verminderten Akzeptanz des Projektes durch die Nutzer führen können.

#### ***IV. Praktische Hinweise: Merkposten bei der rechtlichen Gestaltung von freien Projekten***

Bei der Initiierung und Gestaltung von Open Source Projekten ist unter anderem<sup>168</sup> – neben der Auswahl einer geeigneten Lizenz<sup>169</sup> – besonderes Augenmerk zu richten, auf die Fragen der Rechtsinhaberschaft an den Codebestandteilen sowie der Ausgestaltung von Koordinierungseinheiten.

Bei der Frage der Rechtsinhaberschaft ist insbesondere zu berücksichtigen:

- Wer ist Inhaber der erforderlichen Rechte? Können u.U. nur mehrere Personen bzw. Institutionen gleichzeitig die Entscheidung zur Open Source Lizenzierung treffen?
- Bestehen vertragliche Verpflichtungen gegenüber Dritten oder wurde über die Rechte bereits verfügt, so dass daher eine Open Source Lizenzierung ausgeschlossen ist?

Für die Frage, in welcher Hand die Koordinierung der laufenden Entwicklung liegen soll, sind unter anderem die folgenden Aspekte zu berücksichtigen:

- Die Herausgabe von offiziellen Versionsziffern kann ein gewisses Haftungsrisiko beinhalten. Wer soll dieses tragen?

---

<sup>168</sup> Ein umfassendes Modell zur Veröffentlichung von Software unter dem Open Source Modell findet sich u.a. bei *Hohensohn/Bretschneider/Hang*, Open Source Software Release: A Guideline for the Publication of Software under the Open Source Status, <[http://oss.fh-coburg.de/events/OSSIE04/hang\\_oss\\_release.pdf](http://oss.fh-coburg.de/events/OSSIE04/hang_oss_release.pdf)>.

<sup>169</sup> Vgl. dazu bereits die praktischen Hinweise oben S. 51 f.

- Soll eine Übertragung der Rechte durch die Nutzer an die jeweilige Koordinierungsstelle erfolgen? Werden die Rechte u.U. für ein Dual Licensing benötigt?
- Welche Regelung erzeugt die höchste Akzeptanz bei den Nutzern?

## **Teil 7. Internationale Aspekte**

### ***I. Bedeutung der Frage nach dem anwendbaren Recht***

Kommt es über Rechte und Pflichten aus einer Open Source Lizenz oder urheberrechtliche Fragen, z.B. nach dem Urheber/Rechtsinhaber oder einer Verletzungshandlung zum Streit, kann sich die schwierige Frage stellen, nach welcher Rechtsordnung der eine oder andere umstrittene Punkt beurteilt werden muss. Besonders bei Auseinandersetzungen über Open Source Software wird eine solche Einschätzung nicht selten erforderlich sein, da die Personen, die in einen Streit verwickelt sind, sich oft in unterschiedlichen Ländern befinden werden. Vertreibt beispielsweise eine amerikanische Gesellschaft ihre Open Source Software über eine Website und lädt ein Nutzer aus Deutschland sich diese herunter, kommt zwischen den beiden ein Lizenzvertrag zustande. Nach welchem Recht aber ist dieser zu beurteilen, falls der Nutzer z.B. gegen die Pflichten aus der freien Softwarelizenz verstößt? Oder: ein Deutscher stellt ein Programm unter eine Open Source Lizenz und vertreibt diese in Deutschland. Ein Franzose behauptet, der Code sei von ihm entwickelt worden und möchte den Deutschen wegen der Urheberrechtsverletzung verklagen. Auch hier stellt sich die Frage nach dem anwendbaren Recht.

Die Bestimmung des anwendbaren Rechts bei internationalen Sachverhalten wird durch das Internationale Privatrecht (IPR) geregelt, das - entgegen des Eindrucks, den die Bezeichnung erweckt - kein international gültiges, sondern nationales Recht ist. Jeder Staat bestimmt selbst darüber, welches Recht auf welchen Sachverhalt mit Auslandsbezug anwendbar ist, die nationalen Gerichte wenden bei der Beurteilung der Frage nach dem anwendbaren Recht jeweils das eigene IPR an.

## **II. Wann ist deutsches Recht anwendbar?**

### **1. Unterscheidung zwischen Urheber- und Vertragsstatut**

Wann deutsches Recht auf einen internationalen Sachverhalt Anwendung findet, hängt zunächst davon ab, ob der jeweils zu beurteilende Aspekt nach dem sog. „Vertragsstatut“ (nach dem das anwendbare Recht für vertragsrechtliche Fragen bestimmt wird) oder dem „Urheberrechtsstatut“ (nach dem das anwendbare Recht für urheberrechtliche Fragen bestimmt wird) beurteilt werden muss. Beide Statuten haben unterschiedliche Anknüpfungspunkte, so dass die Beurteilung nach dem einen oder anderen Grundsatz zu abweichenden Ergebnissen bei der Frage nach dem anwendbaren Recht führen kann.

Unter das Vertragsstatut fällt das Zustandekommen des Lizenzvertrages, die Wirksamkeit des Vertragsschlusses, die Auslegung der Bestimmungen eines Vertrages (einer Lizenz) sowie Haftung und Gewährleistung, vgl. Art. 32 des Einführungsgesetzes zum Bürgerlichen Gesetzbuch (EGBGB). Dagegen ist das anwendbare Recht in Bezug auf die Schutzfähigkeit eines Programms, die Frage nach dem ersten Inhaber des Urheberrechts, die Miturheberschaft, die Übertragbarkeit urheberrechtlicher Nutzungsbefugnisse, die Verletzung des Urheberrechts, die Schutzdauer und die Rechtsgeschäfte über das Urheberpersönlichkeitsrecht nach dem Urheberrechtsstatut zu ermitteln<sup>170</sup>.

#### **a) Die Bestimmung des anwendbaren Rechts über das Vertragsstatut**

Das Vertragsstatut ist in den Art. 27 ff. des Einführungsgesetzes zum Bürgerlichen Gesetzbuch (EGBGB) geregelt. Die Normen des EGBGB zum Vertragsstatut basieren auf dem in der ganzen Europäischen Union geltenden „Übereinkommen von Rom über das auf vertragliche Schuldverhältnisse anzuwendende Recht von 1980 (EVÜ)“. Soweit an einem Vertrag also zwei Parteien aus unterschiedlichen Staaten der EU beteiligt sind, wird das Vertragsstatut gleich zu beurteilen sein.

Nach Art. 27 Absatz 1 EGBGB gilt vorrangig das Prinzip der freien Rechtswahl. Hiernach können die Parteien eines Vertrages selbst das anwendbare Recht verbindlich vereinba-

---

<sup>170</sup> Vgl. Jaeger/Metzger, Open Source Software, S. 90.

ren. Schließen z.B. ein Deutscher und ein Franzose einen Lizenzvertrag und vereinbaren die Geltung deutschen Rechts, findet dies auf vertragsrechtliche Fragen, wie die Sachmängelgewährleistung, die Erbringung der Leistungen oder die Vertragsauslegung, Anwendung. Grenzen der freien Rechtswahl ergeben sich unter anderem aus Art. 29 EGBGB. Danach darf bei Verbraucherverträgen die Rechtswahl nicht dazu führen, dass dem Verbraucher die zwingenden verbraucherschutzrechtlichen Rechte entzogen werden, die in demjenigen Staat gelten, in dem er seinen gewöhnlichen Aufenthalt hat.

Die meisten Open Source Lizenzen enthalten keine Rechtswahlklausel<sup>171</sup>. Kommt es über schuldrechtliche Aspekte solcher Lizenzen zwischen Parteien unterschiedlicher Nationalität zum Streit, muss das Vertragsstatut anhand der gesetzlichen Regeln zum IPR ermittelt werden. In Deutschland gilt in solchen Fällen Art. 28 EGBGB. Hiernach findet auf vertragsrechtliche Fragen das Recht des Ortes Anwendung, „mit dem der Vertrag die engsten Verbindungen aufweist“. Nach Art. 28 Absatz 2 EGBGB ist dies im Zweifel der Ort, an dem die für den Vertrag „charakteristische Leistung“ erbracht wird.

Bei der Ermittlung des anwendbaren Rechts über das Vertragsstatut muss man also zunächst untersuchen, um welche Art Vertrag es sich handelt und welche charakteristischen Elemente dieser enthält. Im Zusammenhang mit der Verbreitung von Open Source Software ist zu differenzieren zwischen dem Lizenzvertrag und anderen Leistungen, die in manchen Fällen von einer der Parteien (regelmäßig einem Distributor) erbracht werden. Der Lizenzvertrag ist die freie Softwarelizenz<sup>172</sup>. Diese regelt lediglich die Befugnisse bei der Verwertung des jeweiligen Programms. Kauft der Nutzer z.B. eine Programmkopie im Softwarehandel, wird zudem ein Kaufvertrag zwischen dem Händler und dem Nutzer geschlossen, der in jeder Hinsicht – auch bei der Einschätzung des anwendbaren Rechts – von dem Lizenzvertrag zu unterscheiden ist.

## **(1) Lizenzverträge**

---

<sup>171</sup> Ausnahmen machen diesbezüglich die MPL und die CPL. In Ziff. 11 ist die Anwendbarkeit des Rechts des Staates Kalifornien bestimmt; Ziff. 7 CPL erklärt das Recht in New York für anwendbar. Auch in den beiden bisher existierenden deutschen freien Softwarelizenzen finden sich Rechtswahlklauseln. Sowohl § 15 der „Bremer Lizenz für freie Softwarebibliotheken“ (zu finden unter <<http://www.ostc.de/Bremer-Lizenz.pdf>>) als auch § 10 der „Lizenz für freie Software“, <<http://www.d-fsl.de>>, erklären deutsches Recht für anwendbar.

<sup>172</sup> Näheres hierzu bereits oben S. 55.

Bei einem Lizenzvertrag liegt die charakteristische Leistung in der Einräumung der Nutzungsrechte. Die vertragsrechtlichen Fragen einer Open Source Lizenz sind daher nach dem Recht zu beurteilen, das an dem Ort gilt, an dem der Lizenzgeber seinen gewöhnlichen Aufenthaltsort hat. Ist der Lizenzgeber eine juristische Person, ein Verein oder eine Gesellschaft, gilt das Recht am Ort von deren Hauptverwaltung (vgl. Art. 28 Absatz 2 Satz 1 EGBGB). Lizenzgeber ist bei Open Source Lizenzen stets der Rechtsinhaber (vgl. z.B. Ziff. 6 GPL). Es kommt daher zur Bestimmung des anwendbaren Rechts nach dem Vertragsstatut in Bezug auf die lizenzrechtlichen Fragen nicht auf den Sitz des Distributors oder desjenigen an, von dem man die Programmkopie erhalten hat. Lädt sich ein Nutzer beispielsweise eine GPL-Software von einem Server in den USA herunter, die von Entwicklern in Deutschland, Frankreich und der Schweiz programmiert wurde, ist in Bezug auf die Frage nach den sich aus der Lizenz ergebenden Befugnissen nicht etwa das Recht der USA anwendbar, sondern die deutsches, französisches und Schweizer Recht. Wird ein Programm von in Deutschland lebenden Entwicklern unter eine Open Source Lizenz gestellt, findet also deutsches Recht auf die schuldrechtlichen Aspekte der auf dieser Basis mit Nutzern geschlossenen Lizenzverträge Anwendung.

Dies klingt zunächst einfach, wird aber dann kompliziert, wenn – wie bei Freier Software häufig der Fall – mehrere Lizenzgeber (z.B. Entwicklergemeinschaften und Bearbeiter der Software) existieren, die in unterschiedlichen Ländern ansässig sind. Hier kommt es zu einer Aufspaltung der anwendbaren Rechtsordnungen. Handelt es sich nicht um ein gemeinsames Werk, also eine Software, die in Miturheberschaft entwickelt wurde, findet für jeden Bestandteil gesondert das Recht am Aufenthaltsort bzw. Sitz des jeweiligen Rechtsinhabers Anwendung. Gleiches gilt, wenn ein Programm in Miturheberschaft durch eine rein virtuell verbundene Entwicklergemeinschaft erstellt wurde, wenn deren Mitglieder in verschiedenen Ländern sitzen und sie keine Gesellschaft (z.B. eine Gesellschaft bürgerlichen Rechts, GbR) mit einem fest bestimmten Sitz gegründet haben. Haben sich Entwickler aus verschiedenen Nationen dagegen zu einer Entwicklergemeinschaft zusammengetan und zwecks Verwertung der Open Source Software an einem bestimmten Ort eine Rechtsperson (z.B. die Apache Foundation) gegründet, kann diese als Lizenzgeberin fungieren. Gleiches gilt, wenn die Rechte an der Freien Software zur

Wahrnehmung durch ein "Copyright Assignment"<sup>173</sup> auf die FSF oder durch ein "Fiduciary Licence Agreement" (FLA)<sup>174</sup> auf die FSF Europe übertragen wurden. Diese Möglichkeit ist nach Ziff. 10 der GPL vorgesehen. In einem solchen Fall fungiert die FSF (oder in oben genannten Fällen eine Gesellschaft) als Lizenzgeber, so dass - jedenfalls nach deutschem IPR (Art. 28 Absatz 1 EGBGB) - deren Sitz als Anknüpfungspunkt für die lizenzvertragsrechtlichen Fragen gilt. Handelt es sich jedoch um eine von einem Dritten, der nicht der Entwickler-Gesellschaft zugehört oder der seine Rechte nicht der FSF übertragen hat, bearbeitete Version der Software, sind die lizenzvertraglichen Aspekte für dessen Bestandteile wiederum nach dem Recht seines Aufenthaltsortes zu beurteilen.

## **(2) Andere Vertragstypen**

Auch bei den anderen im Umfeld von Open Source Software abgeschlossenen Verträgen kommt es für die Bestimmung des anzuwendenden Vertragsrechts regelmäßig auf den Ort an, an dem die charakteristische Leistung erbracht wird. Geht es um die Pflichten des Verkäufers/Distributors, liegt die charakteristische Leistung in der Verschaffung des Eigentums an der Programmkopie. Diese wird regelmäßig am Sitz des Verkäufers erbracht.

Probleme bereitet allerdings die Ermittlung des anwendbaren Rechts, wenn eine Software von einem Server im Ausland heruntergeladen wurde. Wenn sich z.B. ein deutscher Nutzer eine Open Source Software von einem Server in den USA herunterlädt, kann unklar sein, wer der Anbieter des Download-Dienstes ist und wo dieser seinen Sitz hat. Dies gilt v.a. für Community-Plattformen wie Sourceforge. Als Anbieter kommen insoweit der Plattformbetreiber oder das die Softwareentwicklung betreuende Team in Betracht. Ist dieses Team nicht an einem bestimmten Ort gesellschaftlich organisiert, sondern sitzen dessen Mitglieder in verschiedenen Ländern, kann kein einheitlicher Sitz des Anbieters der Software bestimmt werden. Welches Recht in solchen Fällen anwendbar ist, ist eine zurzeit noch ungeklärte Rechtsfrage. Es wäre denkbar, hier an den

---

<sup>173</sup> Siehe hierzu <<http://www.fsf.org/licenses/why-assign.html>>.

<sup>174</sup> Vgl. <<http://www.germany.fsf.europa.org/projects/fla/fla.de.html>>.

gewöhnlichen Aufenthaltsort des Serverbetreibers anzuknüpfen<sup>175</sup> oder an den Ort, von wo das Werk heruntergeladen wurde.

## b) Die Bestimmung des anwendbaren Rechts über das Urheberstatut

Dem Urheberrecht liegt das sog. Territorialitätsprinzip zugrunde, das gesetzlich nicht geregelt ist. Hierbei handelt es sich um einen international anerkannten Rechtsgrundsatz, der besagt, dass an einem Werk nicht „ein einzelnes Urheberrecht“ besteht, sondern viele, jeweils nur innerhalb eines Staates geltende Urheberrechte. Jedes Urheberrecht endet also an der Grenze des Staates, das es gewährt. „Das“ Urheberrecht, beispielsweise eines Programmierers an einer Software, besteht also aus internationaler Sicht aus einer Vielzahl einzelner, territorial begrenzter Urheberrechte.

Dieser Grundsatz hat Auswirkungen auf die Frage nach dem anwendbaren Recht. Basierend auf dem Territorialitätsprinzip gilt in Bezug auf urheberrechtliche Fragen eine spezielle Anknüpfungsregel, das sog. Schutzlandprinzip<sup>176</sup>. Dieses besagt, dass auf urheberrechtliche Fragen stets das Recht des Landes Anwendung findet, für dessen Gebiet der Rechtsinhaber Schutz sucht<sup>177</sup>. Grundsätzlich findet hiernach das deutsche Urheberrecht Anwendung, wenn die Verletzungshandlung, z.B. eine ungenehmigte Vervielfältigung oder Verbreitung, in Deutschland stattgefunden hat. Schwieriger wird die Beurteilung, wenn sich eine Urheberrechtsverletzung grenzüberschreitend auswirkt. Dies gilt besonders für Handlungen, die über das Internet vorgenommen werden. Lädt jemand eine Software auf einen im Ausland befindlichen Server (hierin liegt eine Vervielfältigung), ist das Recht des Landes anwendbar, in dem sich die Person befindet, die den Upload vornimmt<sup>178</sup>. Nimmt jemand von einem ausländischen Server einen Download vor (auch dies ist eine Vervielfältigungshandlung), ist an den Ort anzuknüpfen, von dem

---

<sup>175</sup> Ähnlich *Spindler*, Rechtsfragen bei Open Source, S. 143. Allerdings kann diese Konstruktion zu unbefriedigenden Ergebnissen führen, wenn dieser Ort zu den Rechtsordnungen der an einem Erwerbsgeschäft über eine Open Source Software Beteiligten keine Bezüge aufweist. Sinnvoll kann dies dementsprechend nur sein, wenn das Entwicklerteam oder zumindest ein Mitglied desselben, selbst den Server betreibt, da dann wenigstens ein Teil einer Vertragspartei (ein Mitglied des Anbieters des Downloaddienstes) zu der anwendbaren Rechtsordnung eine Bindung hat.

<sup>176</sup> Vgl. etwa BGH MMR 1998, S. 35 ff. - Spielbankaffaire.

<sup>177</sup> Vgl. *Ulmer*, Immaterialgüterrechte im internationalen Privatrecht, Nr. 18 (S. 12).

<sup>178</sup> *Jaeger/Metzger*, Open Source Software, S. 91, auch zum Folgenden. Manche Autoren wollen jedoch auf den Standort des Servers abstellen, vgl. etwa *Hoeren/Thum* in *Dittrich* (Hrsg.), Beiträge zum Urheberrecht, Band 5, Wien 1997, S. 84.

aus der Nutzer den Download vornimmt<sup>179</sup>. Kompliziert ist die Bestimmung des auf Internet-Angebote anwendbaren Rechts. Da mit dem Angebot in aller Regel ein Werk (z.B. ein Programm) auf der ganzen Welt zugänglich gemacht wird, sind auch weltweit sämtliche Rechtsordnungen von dieser Handlung betroffen. Folglich wäre das Recht aller Staaten gleichzeitig auf ein und dieselbe Handlung anwendbar. Dieser Umstand kann für den Anbieter der Software sehr problematisch sein. Theoretisch müsste er hiernach sein Verhalten an allen Urheberrechtsordnungen der Erde ausrichten. Ob dies im Streitfall von den Gerichten auch so gesehen wird, ist bisher ungewiss, da hierüber ersichtlich noch nicht entschieden wurde. Immerhin existieren Ansätze, die mit dem Schutzlandprinzip bei Nutzungshandlungen im Internet einhergehende Rechtsunsicherheit abzumildern. So wurde z.B. vorgeschlagen, nur auf das Recht der Länder abzustellen, auf die ein Angebot bewusst und gezielt ausgerichtet ist. Ob sich solche Modelle indes durchsetzen können, ist gegenwärtig nicht abzusehen.

## 2. Fazit

Die Fragen des anwendbaren Rechts sind außerordentlich komplex. Das aus den Zeiten vor dem Internet-Verkehr stammende IPR ist den globalen Herausforderungen bislang nicht gewachsen. Neue internationalprivatrechtliche Regelungsmodelle, die mehr Rechtssicherheit versprechen, begegnen meist anderen Bedenken und sind auch nur schwer – zumal im internationalen Raum – durchsetzbar.

Daher versucht man die Schwierigkeiten der Anwendbarkeit unterschiedlicher Rechtsordnungen zu umgehen, indem das Sachrecht (also z.B. das Urheber- oder Verbraucherschutzrecht) zunehmend harmonisiert wird. Der Sinn einer solchen Vorgehensweise liegt auf der Hand: ähneln sich die zur Anwendung infrage kommenden Rechtsordnungen oder sind sie gar identisch, schwindet die Bedeutung des internationalen Privatrechts. Allerdings ist man von einem international einheitlichen Recht selbst auf dem Gebiet des Urheberrechts, in dem schon seit langem an einer Vereinheitlichung der Rechtsordnungen gearbeitet wird, noch weit entfernt.

---

<sup>179</sup> Möhring/Nicolini-Hartmann, vor §§ 120 ff., Rz. 32.

Immerhin existieren innerhalb der Europäischen Union in weiten Teilen schon heute einheitliche Gesetze. Dies gilt vor allem für den Urheberrechtsschutz von Computerprogrammen, der auf eine EU-Richtlinie von 1991 zurückgeht.

### **III. Praktische Hinweise: Sollten Rechtswahlklauseln verwendet werden?**

Zur Vermeidung von Problemen, die sich aufgrund des Urheberstatuts ergeben können, können Rechtswahlklauseln nichts beitragen. Das Schutzlandprinzip ist eine zwingende Anknüpfungsregel, von der durch Verträge nicht abgewichen werden kann.

Etwas anderes gilt jedoch in Bezug auf die außerordentlichen Schwierigkeiten, die sich bei der Bestimmung des anwendbaren Rechts aufgrund des Vertragsstatuts ergeben können. Ohne Rechtswahlklausel herrscht für alle Vertragspartner in Bezug auf die vertraglichen Pflichten Rechtsunsicherheit. So ist es für einen Lizenzgeber einer Open Source Software ohne eine solche Bestimmung schon kaum möglich, sich über das Maß seiner Gewährleistungsverpflichtung im Klaren zu sein, da ohne eine Rechtswahlklausel auf diese Frage u.U. eine Vielzahl von Rechtsordnungen zur Anwendung kommen können<sup>180</sup>. Diesbezüglich können Rechtswahlklauseln in freien Softwarelizenzen einen wichtigen Nutzen erfüllen. So wäre es für deutsche Open Source-Projekte beispielsweise im Regelfall von erheblichem Vorteil, wenn die verwendete Lizenz eine Anwendbarkeit deutschen Rechts bestimmen würde<sup>181</sup>. Aus Sicht jedenfalls des deutschen und europäischen IPR würde dies dazu führen, dass Verträge über Programme, die von deutschen Entwicklern unter Open Source Lizenzen gestellt werden, im Bereich der Anwendbarkeit des Vertragsstatuts auch nach deutschem Recht beurteilt würden. Damit ist zwar nicht ausgeschlossen, dass ein ausländisches Gericht nach dem dort geltenden IPR zu einem anderen Ergebnis, also zur Anwendbarkeit einer anderen Rechtsordnung gelangt. Viele per Open Source Lizenz zwischen deutschen Entwicklern und Ausländern geschlossene Lizenzverträge könnten jedoch von der Rechtssicherheit einer solchen Rechtswahlklausel profitieren.

---

<sup>180</sup> Die vorstehenden Ausführungen zur Bestimmung des anwendbaren Rechts auf Gewährleistungspflichten in Lizenz- und Kaufverträgen gelten nur für das deutsche IPR. Nach dem IPR anderer Staaten kann sich jedoch durchaus ergeben, dass die Rechte der Länder zur Anwendung kommen, in denen ein Programm verbreitet wird (Recht am Sitz des Lizenznehmers/Käufers).

<sup>181</sup> So wie dies in den unter Fn. 28 genannten deutschen Lizenzen vorgesehen ist.

Bei international verbreiteter Software können Rechtswahlklauseln jedoch auch zu Problemen führen. So ist denkbar, dass manche Nutzer sich von der Nutzung abschrecken lassen, wenn sie sich hierbei nach einer fremden Rechtsordnung richten müssen.

Auch wenn freie Softwarelizenzen verwendet werden, die keine Rechtswahlklausel vorsehen (wie die GPL), können von Seiten der Entwickler, die ihre Programme hierunter vertreiben wollen, Vorkehrungen getroffen werden, die die Probleme bei der Bestimmung des anwendbaren Rechts zumindest verringern. Soweit möglich, wäre es beispielsweise v.a. in kleinen Entwicklergemeinschaften (mit Mitgliedern aus verschiedenen Ländern) sinnvoll, eine Gesellschaft mit festem Sitz zu gründen, die als Lizenzgeber auftritt. Bei großen, v.a. unorganisierten Entwicklergemeinschaften wäre eine solche Vorgehensweise zwar umso vorteilhafter, dürfte aber im Zweifel an den praktischen Schwierigkeiten einer Entscheidungsfindung (alle Mitglieder der Entwicklergemeinde müssten dem zustimmen) scheitern.

Eine Alternative kann darin liegen, die Rechte einer bereits bestehenden Gesellschaft oder Institution zur treuhänderischen Wahrnehmung zu übertragen, wie dies auch bei einer Übertragung auf die FSF durch ein „Copyright Assignment“ oder das „Fiduciary License Agreement“ geschieht. Auch eine solche Maßnahme führt zur einer Vereinfachung der Frage nach dem anwendbaren Recht bei internationalen Sachverhalten und im Übrigen auch zu einer erleichterten Durchsetzbarkeit der Rechte an einer Open Source Software, da die Rechte der u.U. vielen Einzelpersonen auch hier gebündelt werden.