

Rechtliche Verantwortungsebenen und dingliche Verfügungen bei der Überlassung von Open Source Software

Benno Heussen¹

Die kostenlose Überlassung von Software hat sich im Laufe der letzten Jahre zu einem eigenen Geschäftsmodell entwickelt, das die unterschiedlichsten Ebenen bis hin zum Endkunden, der ebenso Unternehmer wie Verbraucher sein kann, umfasst. Seine Produkte (z.B. Linux) verkörpern symbolisch die Freiheit des Informationsaustausches als Gegenmodell zu proprietärer Software (z.B. Microsoft). Die aktuelle Streitfrage, ob und inwieweit das AGB-Recht die Rechtsbeziehungen zwischen den Beteiligten regelt, kann gelöst werden, wenn man einzelne Verantwortungsebenen abschichtet und dabei die Beziehungen zwischen Fachleuten, Unternehmern und Verbrauchern voneinander trennt. So wird es möglich, die Rechtsbeziehungen außerhalb der Verbraucherebene und teilweise auch auf Unternehmensebene aus dem vertraglichen Kontext zu lösen und als rein dingliche Verfügungen zu interpretieren. Die dann noch verbleibenden gesetzlichen Haftungsregeln reichen für den Schutz auf der Ebene der Fachleute und (mindestens teilweise) der Unternehmer völlig aus.

I. Ein typisches Szenario

Die Diskussion über die Rechtsprobleme¹ bei Open Source Software (OSS) leidet darunter, dass die Autoren selten klar genug sagen, an welche Fallgestaltungen sie ihre rechtlichen Überlegungen anknüpfen. Bei der schwierigen Mischlage unterschiedlicher Sachverhalte kommt man um ein komplexes Szenario nicht herum:

Computertomografen² sind komplexe technische Systeme, die aus mehreren Einzelkomponenten bestehen. Um für bestimmte Darstellungen (z.B. dreidimensionale Schnitte) handelsübliche PCs verwenden zu können, müssen individuelle Programme geschrieben werden.

Ein Softwareentwickler B, der sowohl Arzt an einem Universitätskrankenhaus wie Ingenieur ist, hat ein solches Programm entwickelt. Dazu hat er seinerseits auf ein Computerprogramm zurückgegriffen, das der Informatikprofessor A im Rahmen eigener wissenschaftlicher Tätigkeit zur Steuerung von komplexen Maschinen entworfen hat. Welche Maschinen das im Einzelfall sein können, ist in dem Programm nicht definiert. Um die Rechengänge in den oberhalb dieses Programms liegenden weiteren Softwareebenen zu steuern, enthält dieses Computerprogramm an bestimmten Stellen eine Rechenregel (Algorithmus), die dafür sorgt, die Länge der Zahlenreihen durch Rundungsdifferenzen zu begrenzen. An welcher Stelle der jeweiligen Zahlenreihe die Rundungsdifferenz eingesetzt wird, ist nicht definiert.

Der Softwareentwickler B übersieht bei Entwicklung seines PC-Schnittstellenprogramms, dass es für die präzise Feineinstellung im Bereich der Anwendungsprogramme erforderlich sein wird, Zahlenreihen von zehn Stellen hinter dem Komma zuzulassen. Stattdessen lässt er nur fünf Stellen hinter dem Komma zu. B berichtet über seine Arbeit in einer ärztlichen Fachzeitschrift und stellt den Aufsatz samt Programm und Entwicklungsdokumentation ins Internet.³ Kollegen aus anderen Universitätsstandorten sind beeindruckt und beginnen damit, unterschiedliche Anwendungsprogramme für einzelne Krankheiten zu entwickeln.⁴ Die Entwickler A und B haben ihre Leistungen jeweils der Öffentlichkeit allgemein auf der Basis der GNU-GPL-Lizenzbedingungen⁵ zugänglich gemacht.

Für den Fachmann ist auf Grund der Offenlegung des Algorithmus und der vorhandenen Dokumentation ohne weiteres ersichtlich, dass die Rundungsdifferenzen fünf Stellen zu früh ansetzen und daher die Anwendungsprogramme nicht präzise sein werden. Softwareentwickler C besitzt diese Kenntnisse leider nicht und entwickelt ein Anwendungsprogramm für Lungenkarzinome auf der Basis des falschen Algorithmus. Er bietet dieses Programm einer Vielzahl von Krankenhäusern an, wobei er keine Lizenzgebühren verlangt, sondern nur für begleitende Tätigkeiten vergütet werden will.

Krankenhauschef K entschließt sich zum Einsatz dieses Programms und lässt es durch C installieren und anpassen. Ferner kommt ein Wartungs- und Pflegevertrag mit C zu Stande, in deren AGB unter Hinweis auf GNU-GPL-Lizenzbestimmungen ausdrücklich vereinbart wird, dass C für Fehler und darauf beruhende Schäden der kostenlos gelieferten Software nicht haften will. Schon bei den ersten

Heussen: Rechtliche Verantwortungsebenen und dingliche Verfügungen bei der Überlassung von Open Source Software(MMR 2004, 445)

446

fünf Testversuchen werden einige Patienten durch falsch dosierte Strahlungen verletzt. Softwareentwickler C korrigiert den Fehler sofort, kann aber die Schäden nicht ungeschehen machen.

Bei diesem typischen Szenario gehen alle Autoren bisher davon aus, die Software werde an denjenigen, der sie sich aus allgemein zugänglichen Quellen beschafft, verschenkt. Die Aussagen in der GNU-GPL-Lizenz, dass „das volle Risiko bezüglich Qualität und Leistungsfähigkeit des Programms“ bei demjenigen liegt, der die Software einsetzt (§ 11), und dass niemand „für irgendwelche Schäden haftbar“ ist (§ 12), werden als Verstoß gegen die gesetzlichen Leitlinien des vertraglichen Schenkungsrechts der §§ 560 ff. BGB empfunden. Man hält nur mehr oder weniger große Einschränkungen der Haftung für Geschenke für akzeptabel.⁶

II. Der technische Ablauf der Überlassung von OSS

Für die weiteren Überlegungen ist es wichtig zu wissen, wie die Verbindungen zwischen den Inhabern der Rechte an OSS und denjenigen hergestellt werden, denen die Software auf den jeweiligen Ebenen überlassen wird.⁷

Unter „Inhabern der Rechte“ ist eine - häufig nicht mehr genau definierbare - Menge von Entwicklern zu verstehen, die entweder linear hintereinander (Reihe) oder in bestimmten Arbeitsgruppen (Pools) die Version der Software erarbeitet haben, die jeweils zur Verfügung gestellt wird. Die Überlassung geschieht auf den unterschiedlichsten Wegen:

Viele Universitäts-Webseiten enthalten die Möglichkeit, OSS über das Internet herunterzuladen.⁸ Anwendergruppen stellen auf ihren eigenen Webseiten die für den jeweiligen Anwendungszweck konzipierte Software zur Verfügung.⁹

Einzelne Anwender tun das Gleiche.¹⁰

In rechtlicher Hinsicht ist dabei von Bedeutung, dass die Software anonym abgeholt werden kann, sodass es keinerlei Lizenznummern oder Registrierungen gibt. Niemand, der OSS ins Netz stellt, kann auch nur eine Vorstellung davon entwickeln, wer auf dieses Angebot zugreift und wer nicht.¹¹ Nirgendwo entsteht ein Kontakt zwischen den Beteiligten, wenn dieser nicht von einer Seite ausdrücklich gewünscht und von der anderen Seite erwidert wird. Das Ganze erinnert auf den ersten Blick an einen (kostenlosen) Essensautomaten, bei dem jemand von vorne etwas aus einem Fach nimmt, das von hinten für andere Interessenten immer wieder neu bestückt wird. Das Bild ist allerdings noch nicht präzise genug, weil keine Gegenstände herausgenommen werden

und keine Substanz ersetzt werden muss.¹² Software besteht am Ende doch, wie man hier deutlich sieht, aus einer Menge von Ideen, die unterschiedliche Aggregatzustände annehmen können: Sind diese Ideen auf einem Datenträger gespeichert, kann man sie in die Tasche stecken, bietet man sie im Netz zur Kopie an, dann werden sie „flüssig“ und können sich von einem zum anderen Ort bewegen, und wenn sie dort auf einen Hardwarefehler treffen, können sie auch verdampfen und sich in Luft auflösen.¹³ Es ist daher besser, sich statt eines Essensautomaten ein Kopiergerät vorzustellen, das in einer großen Bahnhofshalle steht und auf Knopfdruck einen Text ausgibt, den sich jeder mitnehmen darf - jedoch unter der Bedingung, alle erhaltenen und etwa hinzugefügten Informationen ebenfalls ohne Vergütung weiterzugeben. Entscheidend ist in diesem Bild, dass es nur um die Verfügung über ein Bündel von Gegenständen geht, das auf eigenartige Weise zusammengeschnürt worden ist, wobei Geber und Empfänger nicht miteinander in Berührung treten. Die GNU-GPL-Lizenz drückt das folgendermaßen aus: „Sie sind nicht verpflichtet, diese Lizenz anzunehmen, da sie sie nicht unterzeichnet haben“ (§ 5).

III. Die OSS-Philosophie und ihre Veränderung im Markt

Die OSS-Philosophie geht davon aus, dass die einmal allgemein zugänglich gemachte Software sich wie in einem Schneeballsystem verbreiten wird: Jeder Empfänger der Entwicklungsleistung soll nach der Vorstellung des Entwicklers einen Zusatznutzen zu dem Programm schaffen (Added Value) und das Programm dann an weitere Benutzer geben, die ihrerseits mit Intelligenz und Fantasie zur Verbesserung und Verbreitung des Produkts beitragen. Natürlich werden Dritte sich an einem solchen Modell nur beteiligen, wenn sie von der Entwicklungsleistung und ihren Möglichkeiten selbst beeindruckt sind. Wie *Richard M. Stallman* anschaulich sagt, sollte man über das Modell mehr i.S.v. „freier Rede“ und nicht i.S.v. „Freibier“ nachdenken.¹⁴

Außerhalb dieser Entwicklerszene haben sich in den letzten Jahren vier Interessengruppen gebildet, die sich aus technischen oder wirtschaftlichen Gründen von proprietärer Software lösen wollen:

Die Öffentliche Verwaltung sieht darin ein Mittel der Kosteneinsparung und größerer Flexibilität.¹⁵ Entwicklungsländer/Schwellenländer hoffen, auf der Basis von Open Source eigene Anwenderkonzepte entwickeln zu können und sich gleichzeitig von den westlichen Lizenzgebern zu lösen.¹⁶

Große Hersteller von Hardware und Software, die sich keine eigenen Märkte für bestimmte Softwareprogramme aufbauen wollen (z.B. IBM, Siemens, Sun, Oracle), versuchen - wie es früher einmal war - ihre Hardware mit kostenloser Software attraktiver und offener zu machen.¹⁷

Darüber hinaus hat sich ein namenhafter Markt für Installation, Beratung und andere Umgebungsprodukte von Open-Source-Programmen gebildet (SUSE, RED HAT).¹⁸

Heussen: Rechtliche Verantwortungsebenen und dingliche Verfügungen bei der Überlassung von Open Source Software(MMR 2004, 445)

447

Man erwartet, dass die Unternehmensmarktanteile von OSS in Deutschland von 12% im Jahr 2003 auf 24% im Jahr 2007 und damit von € 131 Mio. im Jahr 2003 auf € 307 Mio. im Jahr 2007 wachsen werden.¹⁹ *IBM* soll bereits mehr als USD 1 Mrd. in die Entwicklung des OSS-Markts investiert haben und *ORACLE* beschäftigt tausend Entwickler damit, die eigenen Softwareprodukte OSS-kompatibel zu programmieren.²⁰

Auch auf der Kundenseite wächst überall die Sensibilität dafür, zu welchen Problemen die Abhängigkeit von wenigen großen proprietären Softwareherstellern auf Dauer führen kann. Ein

Teil der Entwicklung ist gewiss auch darauf zurückzuführen, dass der Einstieg in das OSS-Modell ein Teil der Verteilungskämpfe innerhalb der Gesamtindustrie geworden ist.

IV. Entwicklungs-, Verteilungs- und Verantwortungsebenen

Beim Einsatz von OSS lassen sich vier Verantwortungsebenen unterscheiden:

Ebene 1 - Austausch unter Entwicklern: Hier tauschen Entwickler Programme aus (wie etwa den Linux Kernel), die die unmittelbare technische Verbindung zur Hardwareumgebung herstellen, wobei die möglichen Störungsrisiken auf höheren Ebenen allenfalls sehr abstrakt, in der Regel aber gar nicht definiert werden können. Die Beschreibung solcher Risiken in unserem Szenario würde etwa lauten: „Der Algorithmus dieses Rechenmoduls ist so definiert, dass Rundungsdifferenzen ab der fünften Stelle hinter dem Komma einsetzen. Es ist daher nur für Anwendungen geeignet, für die damit die jeweils erforderliche Präzision der Berechnung erreicht werden kann.“²¹ Hier befindet sich im Szenario der Informatiker A, der sein Programm aber selbst ohne Rückgriff auf andere Programme entwickelt und allgemein zugänglich gemacht hat.

Auch im Bereich Middleware²² stehen Entwickler im Programmaustausch. Sie verfügen aber nicht über gleichrangiges technisches Wissen, sondern stehen etwa im Verhältnis des Autors zum Übersetzer: Beide behandeln denselben Text, aber in anderen Variationen. Im Szenario kommt hier der Entwickler B (Arzt) ins Spiel, dessen Programm von der Entwicklung des A (Mathematiker) abhängt.

Auch Softwareentwickler, die sich der Anwendungssoftware widmen (im Szenario: Software-Entwickler C) greifen teilweise direkt, teilweise über Vertriebsmittler auf OSS zu. Hier entsteht ein höheres Gefahrenpotenzial als Missverständnisse zwischen Entwicklern von Betriebssystemen, da es nur wenige Entwickler von Anwendungssoftware gibt, die mit den typischen Entwicklungsproblemen von Betriebssystemen hinreichend vertraut sind.

Ebene 2 - Vertriebsebene: Hersteller/Vertriebsmittler: Anders als bei proprietärer Software erhalten die Vertriebsmittler (z.B. SUSE) auch den Source-Code und können deshalb die notwendigen Änderungen und Anpassungen, die der Endkunde verlangt, selbst durchführen. Es mag aber auch Unternehmen geben, die solche Aufträge nicht übernehmen, sondern die OSS, so wie sie ist, einfach weiterreichen.

Ebene 3 - Überlassung an den Endkunden: Endkunde kann sowohl ein Unternehmer wie ein Verbraucher sein. Die Software, die ihm überlassen wird, kann neben der OSS auch proprietäre Programme enthalten. U.a. deshalb sagt die GNU-GPL-Lizenz schon im Vorwort, dass jeder seinem direkten Abnehmer die Bedingungen der Lizenz mitteilen muss, „damit sie ihre Rechte kennen.“

Ebene 4 - Dritte, deren Rechtsgüter berührt sind: Das sind die Vertragspartner des Endkunden und deliktisch Betroffene, also die Patienten im Szenario.

V. Das Modell der dinglichen Verfügung

Die bisherige Diskussion um die Rechtsfragen der OSS geht ganz selbstverständlich davon aus, dass es sich auf allen Verantwortungsebenen und an jeder Schnittstelle in der Kette der Rechtsübertragungen um Vertragsbeziehungen handelt, die durch die Lizenz geknüpft werden. Mein Vorschlag ist es, sich von dieser Vorstellung zu lösen und die Weitergabe von Rechten in der Lizenzkette (mindestens auf den Ebenen 1 und 2) als rein dingliche Verfügungen ohne schuldrechtliches „Grundgeschäft“ zu behandeln.

1. Der Inhalt der Lizenz

Der Begriff „Lizenz“ hat keine Aussagekraft in Bezug auf das rechtliche Modell, das mit einer Vereinbarung zu Stande kommt.²³ Er besagt lediglich, dass der Lizenzgeber keine unbeweglichen oder beweglichen Sachen weitergeben, sondern Rechte einräumen will. Diese Einräumung ist stets ein Verfügungsgeschäft,²⁴ hinter dem üblicherweise ein schuldrechtlicher Austauschvertrag steckt, der die wirtschaftliche Basis der Verfügung bildet: Er regelt alle Ansprüche (insbesondere auf Vergütung), die der Lizenzgeber gegenüber demjenigen geltend machen will, den er mit seiner Verfügung bereichert.

Bei der OSS sollen aber gerade keinerlei Ansprüche zu Gunsten des Verfügenden entstehen, die GNU-GPL-Lizenz beschreibt vielmehr nur die Bedingungen, unter denen die Verfügung wirksam werden soll: Im Kern laufen diese Bedingungen auf drei Elemente hinaus, die das Spiegelbild des typischen Copyrights bilden und daher als „Copyleft“ bezeichnet werden:

Der Lizenznehmer muss offen legen, dass er OSS weitergibt, damit er sich nicht etwa heimlich mit fremden Federn schmückt.

Er muss bei der Weitergabe in jeder Hinsicht genauso großzügig sein, wie er selbst behandelt wurde, darf für die Lizenz also keine Vergütung verlangen (wohl aber für Umgebungsleistungen).

Heussen: Rechtliche Verantwortungsebenen und dingliche Verfügungen bei der Überlassung von Open Source Software(MMR 2004, 445)

448

Er muss Dritte davor warnen, dass die Produkte für jemanden gefährlich sein können, der damit in technischer Hinsicht nicht richtig umgehen kann.

Nur durch diese Bedingungen unterscheidet sie sich von Rechten, die Allgemeingut sind (Public Domain): Wer ein Programm ohne GNU-GPL-Lizenz ins Netz stellt und jedermann anbietet, davon Gebrauch zu machen, muss damit rechnen, dass die oben beschriebenen Bedingungen nicht eingehalten werden und kann sich darüber nicht beschweren.

Die Bedingungen der GNU-GPL-Lizenz knüpfen so betrachtet kein schuldrechtliches Band zwischen den Beteiligten, sondern erschöpfen sich darin, die Grenzen des Rechts, über das verfügt wird, näher zu beschreiben. Diese Beschreibung kann nur mit Worten erfolgen. Wer ein Grundstück zum Zweck der Eigentumsübertragung beschreiben will, braucht Pläne, wer ein Buch verschenkt, muss sich körperlich direkt oder indirekt vom Besitz trennen. Anders bei Rechten: Da man sie nicht berühren kann, ist von außen nicht erkennbar, wem sie zustehen und welchen Inhalt sie haben. Deshalb kann es hier keinen gutgläubigen Erwerb geben, denn der gute Glaube findet keinen Anknüpfungspunkt in der Realität.

Nur drei Voraussetzungen sind erforderlich:

Das Recht muss von anderen denkbaren Rechten so abgegrenzt sein, dass es vom Empfängerhorizont desjenigen aus, der das Recht nutzen will, eindeutig von anderen Produkten unterschieden werden kann.

Die Verfügung muss in dinglicher Hinsicht so eindeutig sein, dass die Rechtszuständigkeit dessen, der nun die Nutzungsrechte haben soll, auch gegenüber Dritten unmittelbar erkennbar bleibt.²⁵

Die Verfügung muss von demjenigen ausgehen, der Inhaber des Rechts ist und er muss die Nutzungsrechte auf jemanden übertragen, der nunmehr Inhaber der Rechte sein kann.

Alle diese Voraussetzungen sind bei der GNU-GPL-Lizenz in vollem Umfang erfüllt, sodass sie auf jeden Fall eine dingliche „Verfügung“ darstellt, also ein Rechtsgeschäft, das unmittelbar darauf gerichtet ist, ein bestehendes Recht zu übertragen: „Der Urheberrechtsvertrag ist ein Rechtsgeschäft, welches - zumindest auch - auf die Einräumung von urheberrechtlichen Nutzungsbefugnissen ausgerichtet ist. Insoweit ist lediglich das Verfügungsgeschäft

angesprochen“.²⁶ Die GNU-GPL-Lizenz begründet darüber hinaus keinerlei schuldrechtliche Vereinbarung zwischen Parteien, sondern beschreibt lediglich das Recht nach Art, Umfang und Zeitdauer. Die §§ 11 und 12 der Lizenz sind so gesehen keine Beschränkung vertraglicher Haftung, sondern stellen nur klar, dass es ohne Vertrag auch keine Gewährleistung für Qualität oder sonstige vertragliche Haftung geben kann: Wo kein Schuldner ist, kann auch kein Gläubiger sein!

2. Die Folgen für Qualität und Haftung

Allerdings stellt sich die Frage, wie sich das Modell der dinglichen Verfügung auf den einzelnen Verantwortungsebenen auswirkt, wenn die überlassene Software Fehler hat.

Ebenen 1 und 2: Hier sind Fachleute tätig, die wissen (oder wissen müssen), welche Haftungsgefahren auftauchen, wenn Sie die ihnen überlassene Software ohne die ausreichende Fachkunde weiterverwenden. Hier beschränkt sich das rechtliche Modell auf die dingliche Verfügung, weil Leistungen weder angeboten noch angenommen werden: Angenommen wird nur die Verfügung über die Rechte!

Ebene 3: Auf dieser Ebene dürfte es sehr selten vorkommen, dass ein Endkunde - sei es ein Unternehmer oder ein Verbraucher - sich ohne Unterstützung von Fachleuten die Software herunterlädt. Unterstellen wir gleichwohl, dass dies hier und da geschehen mag, ergibt sich kein anderes Ergebnis. Die Argumentation, ohne vertragliche Bindung entstünde kein akzeptables Schutzniveau für solche Fälle, wäre nichts anderes als ein Zirkelschluss.²⁷

In den weitaus meisten Fällen und vor allem dort, wo große Industrieunternehmen mit OSS arbeiten, wird sie Teil eines Pakets vertraglicher Leistungen sein, da der Empfänger einer solchen Gesamtleistung auch eine Gesamtverantwortung dessen erwarten darf, der sie ihm anbietet. Entscheidend ist dabei neben objektiven Kriterien (technische Systemverantwortung) der aus der Sicht des Kunden gesehene Einheitlichkeitswille.²⁸ Willkürliche Trennungsklauseln sind daher stets als unwirksam betrachtet worden.²⁹ Haftungsbeschränkungen für den „verschenkten“ Leistungsteil werden daher i.R.d. Maßstäbe, die in der bisherigen Diskussion gesetzt worden sind³⁰ unwirksam sein. Das Gleiche gilt für die Fälle, in denen der Leistungsteil der verschenkten Software gleich- oder höherwertig ist als die anderen, gegen Vergütung angebotenen Leistungen.

Ebene 4: Wer als Dritter von Schäden betroffen ist, die OSS ausgelöst hat, kann sich entweder auf vertragliche Beziehungen zu dem Schädiger (im Szenario: Das Krankenhaus) oder auf deliktische Ansprüche gegen einen oder mehrere Beteiligte stützen. Daneben steht ihm das Produkthaftungsrecht zur Seite.

VI. Haftungsmaßstäbe

Wenn i.R.e. dinglichen Verfügungsmodells jeglicher schuldrechtliche Anknüpfungspunkt für Haftungsregelungen fehlt, drängt sich die Frage auf, wie die Haftungsfolgen geregelt werden sollen.

Der erste Ansatz ist die Produkthaftung: Der/die Entwickler sind zweifellos „Hersteller“ (§ 4 ProdHaftG, ihre Haftung bemisst sich nach § 3 ProdHaftG) und es scheint, dass die dort genannten Voraussetzungen ein fairer Ausgleich der Interessen zwischen dem Entwickler und jedem weiteren in der Verteilerkette befindlichen Lizenznehmer sind. Das Produkthaftungsgesetz unterscheidet sinnvoll zwischen Personenschäden und Sachschäden, es knüpft an die Sicherheit eines Produkts an, es berücksichtigt den Stand der Technik (der Zeitpunkt, in dem das Produkt in den Verkehr gebracht wird) und den Empfängerhorizont (den Gebrauch, mit dem billigerweise gerechnet werden kann).

Daneben gelten die deliktsrechtlichen Vorschriften (§ 823 ff. BGB), deren Haftungsmaßstab zwar zusätzlich das Vorliegen

Heussen: Rechtliche Verantwortungsebenen und dingliche Verfügungen bei der Überlassung von Open Source Software(MMR 2004, 445)

449

von Fahrlässigkeit voraussetzt, deren Haftungsfolgen aber einschneidender sind, wenn gegen diesen Maßstab gehandelt wurde.

Es kann Personen geben, die vorsätzlich gefährliche Software verschenken, um anderen damit zu schaden. Ihre Haftung steht außer Zweifel und dürfte sich meist schon aus § 823 Abs. 2 BGB i.V.m. einem Strafgesetz ergeben. Die Fahrlässigkeit hängt davon ab, ob eine Pflichtverletzung durch einen Softwarefehler objektiv erkennbar und vermeidbar ist.³¹ Soweit Restrisiken verbleiben, muss darüber aufgeklärt werden.³² Das Preis-Leistungs-Verhältnis und das Kosten-Nutzen-Verhältnis können bei Ermessensentscheidungen eine Rolle spielen.³³

Die Anwendung dieser Grundsätze zeigt für die GNU-GPL-Lizenz folgendes Bild: Da anders als bei proprietärer Software der Source-Code offen gelegt wird, kann der Lizenznehmer darin steckende Entwicklungsrisiken selbst beurteilen. Er wird es auch als Erster merken, wenn er den Source-Code nicht versteht (falls er ehrlich zu sich selbst ist), und wird dann hoffentlich „die Finger davon lassen“. Er weiß, dass die Überlassung der Software ihm zwar eigene Entwicklungsarbeit erspart, aber nicht die Mühe, Schritt für Schritt nachzuvollziehen, was der Lizenzgeber sich gedacht hat, und die mögliche Gefährlichkeit selbst einzuschätzen. Aus der Empfängerperspektive ist dieser Verantwortungslevel ohne weiteres erkennbar. Auf der Seite des Lizenzgebers ist der Maßstab für die eigene Verantwortung bereits dadurch gesenkt, dass er nicht weiß, was der Lizenznehmer mit der überlassenen Software tatsächlich anfangen will. Es gibt kein schuldrechtliches Band zwischen beiden. Man kann das Szenario in allen Varianten durchspielen und wird mit diesen beiden Ansätzen stets zu fairen³⁴ Ergebnissen kommen.

Darüber hinaus wird es in vielen Fällen, sobald die Verantwortungsebene 4 zwischen Herstellern und Vertriebsmittlern erreicht wird, häufig Verträge über Begleitleistungen geben, die neben der OSS-Lizenz erbracht und vergütet werden. Innerhalb dieser Leistungsbeziehung gelten Gewährleistung und Haftung einschließlich der Regeln des AGB-Rechts (§ 305 ff. BGB) uneingeschränkt. Die Abgrenzung zwischen beiden Bereichen wird im Einzelfall immer wieder Schwierigkeiten machen. Sie zu lösen gehört jedoch zu den typischen Aufgaben des juristischen Handwerks.

Überall dort, wo man von einer rein dinglichen Verfügung ausgeht, müssen allerdings (mindestens) zwei Kernfragen beantwortet werden: Wie wirkt das Modell sich auf den Zweckübertragungsgrundsatz und den Erschöpfungsgrundsatz aus?

VII. Bedeutung des Zweckübertragungsgrundsatzes

Der Zweckübertragungsgrundsatz besagt, dass „sich der Umfang des Nutzungsrechts nach dem mit seiner Einräumung verfolgten Zweck bestimmt“ (§ 31 Abs. 5 UrhG). Er betrifft sowohl einfache schuldrechtliche Lizenzen wie auch die sachenrechtlichen Befugnisse.³⁵ Die GNU-GPL-Lizenz gibt bis auf die unveräußerlichen Persönlichkeitsrechte des/der Entwickler alle Entwicklungsergebnisse aus der Hand, knüpft dies aber an bestimmte Bedingungen, die eingehalten werden müssen. So kann der mit der GNU-GPL-Lizenz verfolgte Zweck des „Free Speech“ nur erreicht werden, wenn jeder, der die Software weitergibt, die Pflicht zur umfassenden Weitergabe auf jeder folgenden Ebene offen anspricht und sich nicht unerlaubt wie ein „Eigentümer“ der Software verhält. § 5 der GNU-GPL-Lizenz sagt: „Indem Sie das Programm (oder ein darauf basierendes Datenwerk) verändern oder verbreiten, erklären Sie Ihr

Einverständnis mit dieser Lizenz und mit allen ihren Bedingungen bzgl. der Vervielfältigung, Verbreitung und Veränderung des Programms oder eines darauf basierenden Datenwerks.“

Während im Bereich des Zweckübertragungsgrundsatzes üblicherweise darum gestritten wird, ob der Lizenznehmer bestimmte Rechte nutzen darf oder nicht, stellt sich die Frage hier anders: Er darf alle Rechte nutzen, diese aber gegenüber Dritten nicht proprietär verengen. Eine Verletzung des Grundsatzes ist daher nicht erkennbar.

VIII. Bedeutung des Erschöpfungsgrundsatzes

Ebenso bedeutsam ist der Erschöpfungsgrundsatz (§ 17 Abs. 2, 69c Nr. 3 Satz 1 UrhG), der besagt, dass der Inhaber eines Rechts auf das Schicksal einzelner Kopien oder dinglicher abgespaltener Rechte keinen Einfluss mehr nehmen kann. Bei üblichen Lizenzmodellen ist der Hersteller normalerweise daran interessiert, die Rechte in dinglicher Form so kleinteilig abzuspalten, dass ein möglichst großer Umfang bei ihm verbleibt.³⁶ Ihm ist bewusst, dass er alles, was er aus der Hand gegeben hat, nicht mehr kontrollieren kann. Die Rechtsprechung lässt eine solche Abspaltung jedoch nur zu, wenn das jeweilige Produkt technisch und/oder wirtschaftlich von verwandten Produkten hinreichend klar unterschieden werden kann. Es muss eine „nach der Verkehrsauffassung als solche hinreichend klar abgrenzbare, wirtschaftlich-technische Verwendungsform“ durch „eine entweder qualitative Verbesserung oder eine quantitative Erweiterung der Nutzungsmöglichkeiten“ entstehen.³⁷ Bei der GNU-GPL-Lizenz erhält der Lizenznehmer seine Rechte unmittelbar vom Hersteller, also nicht über eine Vertriebskette. Schon deshalb ist der Erschöpfungsgrundsatz nicht anwendbar.³⁸

Man kann die Sache aber auch anders sehen: Wenn vom ersten Entwickler angefangen die Software durch eine Vielzahl von Bearbeitungen geht und schließlich beim nächsten Lizenznehmer ankommt, dann könnte man sie auch als ein Bündel von Rechten betrachten, das durch viele Hände gegangen und (ggf.) um jeweils weitere Nutzungsrechte vermehrt worden ist. Aber auch in diesem Fall ist der Erschöpfungsgrundsatz nicht verletzt: Jeder Lizenzgeber will nicht nur, dass sein erster Lizenznehmer alle

Heussen: Rechtliche Verantwortungsebenen und dingliche Verfügungen bei der Überlassung von Open Source Software(MMR 2004, 445)

450

Rechte in vollem Umfang erhält, er sorgt vielmehr mit seinen Bedingungen dafür, dass auch alle weiteren Lizenznehmer in der Kette kein Recht zur Abspaltung von Teilrechten haben. Sieht man das so, dann muss OSS entweder in technischer oder in wirtschaftlicher Hinsicht ein Produkt sein, das sich von anderen Produkten hinreichend klar abgrenzen lässt, denn der Erschöpfungsgrundsatz fordert Klarheit und unmittelbare Erkennbarkeit für den Markt.³⁹ Das Ziel der Abspaltung des Teilmarkts „Open Source Software“ gelingt in der GNU-GPL-Lizenz dadurch, dass in § 3 ein für dieses Teilprodukt besonderes Leistungsbündel vollkommen eigener Art geschnürt wird. Hierdurch entsteht ein in dieser Form bisher unbekannter virtueller Entwicklerpool jedenfalls auf den Ebenen, auf denen Entwickler beteiligt sind. Soweit der Adressat der Lizenz (ich bezeichne ihn hier absichtlich nicht als Vertragspartner) die überlassenen Leistungen lediglich selbst nutzen will, hat der erste Entwickler (oder die Reihen und/oder Pools anderer Entwickler) kein erkennbares Motiv, sich für das Schicksal der von ihm erbrachten Leistungen zu interessieren. Dieses Interesse erwacht aber in aller Schärfe, wenn der Empfänger bearbeitete oder vervielfältigte Stücke an Dritte weitergibt und dabei versucht, das definierte Bündel aufzuschnüren: Versucht er das, verliert er die ihm übertragenen Rechte! Das Bündel ist, wenn man den Maßstab der bisherigen Aufspaltungsrechtsprechung anlegt, hinreichend klar von reinen Vertriebsmodellen abgrenzbar, und zwar schon dadurch, dass bei diesen Vertriebsmodellen nur

der Objectcode weitergegeben wird, nicht aber der Source-Code. Wie *Koch* sehr richtig sagt:⁴⁰ Das eine ist ein Softwareentwicklungsmodell, dessen Ziel eine möglichst schnelle Durchsetzung bestimmter Ideen im Markt ist, das andere ein Vertriebsmodell, das nicht möglichst viele Ideen, sondern möglichst viele Kopien in den Markt schicken will. Das GPL-Leistungsbündel wird von der Idee der „Freie Rede“ zusammengehalten, die Leistungsbündel der Free- oder Shareware hingegen enthalten Werbegeschenke, denen solche Ideen fremd sind.

Wie sieht es aber auf Ebene 3 aus, auf der der Endkunde ein bestimmtes Softwareprodukt erhält, das er nach dem Vertragszweck offensichtlich selbst lediglich benutzen, aber nicht bearbeiten und vervielfältigen will? Die GNU-GPL-Lizenz sagt klar, dass auf jeder Ebene der Weitergabe der Source-Code weitergegeben oder wenigstens angeboten werden muss. Auch hinsichtlich des Endkunden ist die Kontinuität des „Free-Speech-Modells“ mithin gewahrt, auch wenn in der Praxis selten davon Gebrauch gemacht wird.

IX. AGB - Rechtliche Grenzen

Allerdings ist zu bedenken, dass auch für sachenrechtliche Verfügungen das AGBG gilt.⁴¹ Zwar ist die dingliche Verfügung über Rechte keine sachenrechtliche, weil über Ideen und nicht über körperliche Gegenstände verfügt wird, aber auch sie hat ein gesetzliches Leitbild, das oben beschrieben wurde: Fände man in der GNU-GPL-Lizenz also z.B. eine Klausel, die die Klarheit der Verfügung gefährdete, so würde man sie wohl gem. § 307 BGB als unwirksam erachten.

X. Internationale Perspektive

Die saubere Schnittstelle zwischen Verpflichtung und Verfügung (Abstraktionsprinzip), wie sie für das deutsche Recht typisch ist, findet sich in anderen Rechtsordnungen - vor allem in den angloamerikanischen Rechtssystemen - nicht. Dort wird für einen Vertragschluss allerdings verlangt, dass der Empfänger einer Leistung irgendeine Gegenleistung versprechen muss, ohne die es keine vertragliche Bindung geben kann: „In order for a contract to be valid and binding, each party must be bound to give some legal consideration to the other by conferring a benefit upon him or suffering a legal detriment at this request“ (17A American Jurisprudence 2d, Contracts § 117 [May 2002]).⁴²

Demnach fehlt es dort aus ganz anderen Gründen an einer vertraglichen Bindung.

Auch in anderen Rechtsordnungen wird es erhebliche Unterschiede im Schuldrecht geben. Wo immer man aber ein Urheberrecht hat, gibt es Regeln, wie die Rechte übertragen werden können. Es fehlt also niemals an einer Verfügung, auch wenn sie nicht wie im deutschen Recht als „dingliche“ verstanden werden kann.

[1] Rechtsanwalt Dr. Benno Heussen ist Partner der HEUSSEN Rechtsanwaltsgesellschaft mbH (Berlin) und Honorarprofessor an der Universität Hannover. Für technische Hinweise danke ich Herrn Dipl.-Ing. cand. iur. Falk Ewers, für redaktionelle Mithilfe Herrn RRef. Gunnar Thomas, LL.M.

¹ *Jaeger/Metzger, Open Source Software - Rechtliche Rahmenbedingungen der freien Software, 2002. Beide versuchen, über das gesetzliche Leitbild des Schenkungsvertrags die nötige Bewegungsfreiheit zu gewinnen, während Spindler (Hrsg.), Rechtsfragen bei Open Source, 2004, einen erheblich strengeren Maßstab anlegt. Zu den IPR-Fragen: M. Junker, Anwendbares Recht und internationale Zuständigkeit bei Urheberrechtsverletzungen im Internet, Diss. Saarbrücken 2001 (www.cgi.uni-kassel.de/~dbupress/download_frei.pdf.cgi).*

² Hier findet sich eine einfache Beschreibung eines CT: www.klinikum-sn.de/bereiche/kliniken/strahlenklinik/patinfo/ablauf/ct.htm. Allgemeineres zum CT auch unter:

www.surfmed.de/13-diagnose_und_behandlungsverfahren/13-01-diagnosemethoden/13-01-02-bv_u_gk/13-01-02-06-ct.html.

3 Ein Beispiel für ein LINUX-basierendes Programm, welches die Bildverarbeitung für Computertomografen unterstützt, ist im Internet von der Fa. *Digital Medics* zu finden. Es fällt unter die GNU-Lizenz und soll eine Grundlage bilden für die Entwicklung spezieller Anwendungssoftware: www.digitalmedics.de/html/company.html.

4 Dass die Entwicklung von dreidimensionalen Visualisierungsanwendungen für Medizingeräte, basierend auf einem LINUX-Betriebssystem, nicht ganz fern liegend ist, zeigt auch der Artikel von Zerbst, www.linux-magazin.de/Artikel/ausgabe/2001/12/federlesen/federlesen.html.

5 OSS wird sehr häufig unter diesen Lizenzbedingungen weitergegeben (<http://opensource.org/licences>). Daneben existieren aber auch andere Lizenzen, die sich inhaltlich an unterschiedlichen Stellen davon unterscheiden; Übersicht: *Jaeger/Metzger* (o. Fußn. 1), S. 54 ff.

6 Unter der Vielzahl von Aufsätzen zum Thema, die neben den in Fußn. 1 genannten Monografien entstanden sind, ist besonders der Beitrag von *Frank A. Koch*, CR 2000, 273 ff. u. 333 ff. hervorzuheben; s. jüngst auch nochmals *Frank A. Koch*, Rechtsrisiko Open Source Software, Informatik-Spektrum 2004, S. 55.

7 *Jaeger/Metzger* (o. Fußn. 1), Kap. 4.

8 ftp.zrz.tu-berlin.de/pub/packages/linux/.

9 www.parzelle.de; www.tuxmobil.org.

10 Übersicht: www.pro-linux.de.

11 Dies dürfte jedenfalls dann gelten, wenn die selbst entwickelte Software über die Internetseiten einer Vereinigung oder Institution vertrieben wird. Hat der Entwickler eine eigene Seite oder Administratorrechte für die Seite, können mit sog. Trace-Tools wie z.B. Visualroute (www.visualroute.com) zumindest die IP-Nummern der auf die Seite zugreifenden Nutzer ermittelt werden.

12 Wie schnell man beim Nachdenken über Rechte irritiert wird, wenn man sich gedanklich auf die Ebene der Sachen begibt, zeigt ein Dialog aus den Anfängen urheberrechtlichen Denkens: *James Boswell*: „Ich wandte ein, von einem Werk eine gekürzte Fassung abzdrukken, sei gestattet, was dasselbe sei, wie wenn der Kuh Hörner und Schwanz abgetrennt würden.“ *Samuel Johnson*: „Nein, man bewirkt dadurch, dass die Kuh ein Kalb wirft“ (*Boswell/Johnson*, Leben und Meinungen, 1981, S. 666, Dialog v. 20.8.1773).

13 Bereits *Heussen*, GRUR 1987, S. 779.

14 www.gnu.org/philosophie.

15 www.bundestux.de.

16 www.heise.de/newsticker.

17 www.eclipse.org; www.metagroup.de/studien/2002/linux/profil.

18 www.suse.de; www.redhat.com.

19 www.heise.de/newsticker/meldung/38206.

20 Trotz dieser Kommerzialisierung ist anzunehmen, dass die Abneigung der freien Softwareentwickler gegen Reglements jeder Art immer lebendig bleiben und sich sogar verstärken wird: Erstmals haben wir die technischen Mittel, sowohl einsam zu bleiben, als auch unter vielen Einsamen miteinander Kontakt zu halten. *Friedrich Nietzsche* hat das offenbar geahnt als er schrieb „Hundert tiefe Einsamkeiten bilden zusammen die Stadt Venedig - dies ihr Zauber. Ein Bild für die Menschen der Zukunft.“ *Martin Heidegger* zitiert das in einem Brief an *Karl Jaspers* v. 12.8.1949 mit der Bemerkung: „Was hier gedacht wird, liegt außerhalb der Alternative von Kommunikation und Nichtkommunikation.“ (Briefwechsel 1920-1963, 1990, S. 181).

21 Eine solche Risikobeschreibung wird in der Praxis aber nicht vorgenommen, weil sie für jeden Fachmann völlig selbstverständlich ist: Wer mit zehn Stellen hinter dem Komma rechnen muss,

erkennt auf den ersten Blick, dass dies nicht möglich ist, wenn der Rechengvorgang nach fünf Stellen hinter dem Komma abgebrochen wird. Ob und unter welchen Bedingungen man aber mit zehn Stellen rechnen muss, kann der Entwickler auf der untersten Stufe nicht einmal ahnen.

22 Unter Middleware wird Software verstanden, die nicht unmittelbar an der Maschine ansetzt, sondern Betriebssysteme (oder Plattformen) mit Anwendersoftware unterschiedlicher Zielrichtungen verbindet (vgl. avs-www.informatik.uni-hamburg.de/teaching/ss-96/vortrag1/def.htm). Hier können Risiken der Anwenderebene erkennbar werden.

23 *Dreier/Schulze*, Komm. zum Urheberrechtsgesetz, 2004, Rdnr. 6 vor § 31.

24 *Dreier/Schulze* (o.Fußn. 23), Rdnr. 15 ff. zu § 31.

25 *Palandt/Heinrichs*, BGB, 62. Aufl. 2003, Rdnr. 16 vor § 104 BGB.

26 *Nordemann/Hertin*, Urheberrechtsgesetz, 9. Aufl. 1998 vor § 31 Rdnr. 8; s.a. BGHZ 1, 304; *MüKo-BGB/Schramm*, 4. Aufl., Rdnr. 6 zu § 185 BGB.

27 Wenn es bestimmte Fallkonstellationen gibt, die die Schutzbedürftigkeit bestimmter Verkehrskreise nahe legt, kommt es leicht zu solchen Zirkelschlüssen, wie *Doepner* sehr anschaulich für den Bereich des § 1 UWG nachgewiesen hat, GRUR 2003, 825, 826 mit zahlr. Nw. u.a. zu *BGH GRUR* 2001, 1166, 1168 f. - Fernflugpreise.

28 *BGH NJW* 1988, 2880 - Pachtvertrag.

29 *BGH NJW* 1987, 2004 - Architekturkomplettpaket.

30 *S. Jaeger/Metzger* (o. Fußn. 1); *Spindler* (o. Fußn. 1); *Koch* (o. Fußn. 6); u.a.

31 *BGH NJW* 1994, 517, 520 - Gewindeschneidemittel; *BGH NJW* 1991, 1543 - Neue Medikamente.

32 *BGH NJW* 1989, 1538 - Entbindungskonzept.

33 *BGH NJW* 1990, 906 - Pferdeboxen. Es sei hier nur angedeutet, dass für die Frage der Vermeidbarkeit die Grundregeln der ökonomischen Analyse des Zivilrechts einen erheblichen Erkenntnisgewinn bringen (sog. Learned-Hand-Formel s. *Schäfer/Ott*, Lehrbuch der ökonomischen Analyse des Zivilrechts, 3. Aufl. 2000, S. 177 ff.; *MüKo-BGB/Grundmann*, Rdnr. 63 zu § 276 BGB).

34 Der Begriff „Fairness“ bezieht sich auf *John Rawls*, Eine Theorie der Gerechtigkeit, 1979, insb. Kap. 1: Gerechtigkeit als Fairness, S. 19 ff.

35 *BGH NJW* 2002, 896 - Spiegel-CD; *Hertin*, in: *Fromm/Nordemann*, Urheberrecht, 9. überarb. u. erg. Aufl. 1998, Rdnr. 28 zu § 31/32.

36 So hat etwa *Microsoft* erfolglos versucht, Windows-Lizenzen für „Fabrikneue Computer“ und für „Gebrauchte Computer“ voneinander zu trennen (*BGH NJW* 2000, 3571 = MMR 2000, 749 m. Anm. *Leistner* - OEM-Lizenz).

37 *Hertin* (o. Fußn. 35), Rdnr. 5 ff. zu §§ 31/32; möglich z.B. zwischen Hardcover und Taschenbuch (*BGH GRUR* 1992, 310).

38 *Spindler/Wiebe*, CR 2003, 873. *Jaeger/Metzger* (o. Fußn. 1), S. 32 Fußn. 126; *Koch*, CR 2000, 333, 336; *Omsels*, Open Source und das deutsche Vertrags- und Urheberrecht, in: *FS Hertin*, 2002, S. 146.

39 *BGH NJW* 2000, 3571 = MMR 2000, 749 m. Anm. *Leistner* - OEM-Lizenz.

40 *Koch*, CR 2000, 333, 335, bejaht daher folgerichtig ein abspaltbares Recht, lässt aber die Frage, ob daneben auch schuldrechtliche Beziehungen entstehen, bewusst offen (S. 334).

41 *Palandt/Heinrichs*, BGB, 62. Aufl. 2003, Rdnr. 1 Überbl v § 305, Rdnr. 4 zu § 305.

42 Für diesen Hinweis danke ich den Kollegen *Clemens Kochinke* und *Jens Nebel* in Washington.