

Dieser Artikel ist Teil des
Open Source Jahrbuchs 2007

Bernd Lutterbeck
Matthias Bärwolff
Robert A. Gehring (Hrsg.)

Open Source
Jahrbuch 2007

Zwischen freier Software und Gesellschaftsmodell

erhältlich unter www.opensourcejahrbuch.de.

Die komplette Ausgabe enthält viele weitere interessante Artikel. Sie können diesen und andere Artikel im Open-Source-Jahrbuch-Portal kommentieren oder bewerten: www.opensourcejahrbuch.de/portal/. Lob und Kritik sowie weitere Anregungen können Sie uns auch per E-Mail mitteilen.

Produkthaftung für Open-Source-Software?

CHRISTIAN LAUX UND JAN WIDMER



(CC-Lizenz siehe Seite 563)

Wirtschaftlich hat sich Open Source heute etabliert: Viele Unternehmen haben ihr Geschäftsmodell auf Open Source ausgerichtet. Gerichtsentscheide zum Themenkomplex der Haftung und Gewährleistung für Open-Source-Software fehlen aber. Die juristische Fachschaft verbreitet die Warnung, dass das Haftungsrisiko für Open-Source-Software erheblich sei. Nicht selten wird behauptet, es sei für den Entwickler zu gefährlich, sich am Entwicklungsprozess von Open-Source-Software zu beteiligen. In erster Linie wird dabei die zwingende Produkthaftpflicht erwähnt. Der vorliegende Beitrag geht der Frage nach, ob jeder Entwickler, der Open-Source-Software auf eine öffentlich zugängliche Webseite hochlädt, unter dem Titel der Produkthaftung verantwortlich gemacht werden kann. Der Artikel zeigt auf, dass die Haftung nach den Produkthaftpflichtgesetzen von Deutschland und der Schweiz auf wichtige Vertriebsformen von Open-Source-Software nicht passt und dass die Haftungsfrage heute überbewertet werden dürfte.

Schlüsselwörter: Open-Source-Lizenz · aktiver Vertrieb · passiver Vertrieb · Produkthaftung · Haftungsrisiko

1 Open-Source-Software

Open Source war in den letzten Jahren wirtschaftlich sehr erfolgreich. Die Geschäftsmodelle vieler Unternehmen basieren mittlerweile auf Open-Source-Software. Deshalb muss diese auch juristisch erklärt werden können. Eine wachsende Anzahl juristischer Publikationen befasst sich mit Open-Source-Software.¹ Während der große juristische Rahmen im Wesentlichen abgesteckt ist, ist sich die Lehre bezüglich wichtiger Aspekte nicht einig.

¹ Vgl. insbesondere Jaeger und Metzger (2006), Spindler (2004) und Widmer (2003).

Dazu gehört auch das Thema der Haftung für Open-Source-Software. Da Gerichtsentscheide fehlen, bleiben die vorhandenen Stellungnahmen notgedrungen etwas theoretisch. Die juristische Fachschaft verbreitet die Warnung, dass das Haftungsrisiko für Open-Source-Software erheblich sei. In erster Linie wird die zwingende Produkthaftpflicht erwähnt (Jaeger und Metzger 2006, Rz. 225), ebenso eine weitreichende Produzentenhaftpflicht und gelegentlich sogar das Strafrecht.² Solche Stellungnahmen sorgen für große Verunsicherung unter Softwareentwicklern. Ist es für einen Programmierer zu riskant, Open-Source-Software zu entwickeln?

Man muss sich bewusst sein, dass rein rechtsdogmatische Stellungnahmen für gewöhnlich auch außerhalb der fachlichen Diskussion benutzt werden, um je nach Resultat *für* oder *gegen* Open Source Werbung zu machen. Es ist zu vermuten, dass nicht zuletzt aus diesem Grund das Thema rund um Haftung und Gewährleistung heute so stark umstritten ist. Die Reaktionen auf das Spindler-Gutachten (2003) haben dies veranschaulicht.³

Es würde zu viel Raum in Anspruch nehmen, die Grundlagen zu Gewährleistung und Haftung vollständig darzustellen. Der vorliegende Aufsatz beschränkt sich deshalb auf das Thema der Produkthaftung. Diese wurde geschaffen, um den Konsumenten vor den Gefahren von Konsumgütern zu schützen. Die Produkthaftung wurde rechtlich zwingend ausgestaltet und ist damit immun gegenüber Haftungsausschlussklauseln. Bedeutet das nun, dass jeder Entwickler, der Open-Source-Software auf eine öffentlich zugängliche Webseite hochlädt, unter dem Titel der Produkthaftung verantwortlich gemacht werden kann? Der vorliegende Aufsatz soll einen Denkanstoß zu dieser Frage geben. Da die Rechtslage nach deutschem sowie nach schweizerischem Recht weitgehend identisch ist, wird auf die anwendbaren Gesetzesbestimmungen beider Rechtsordnungen verwiesen.⁴

2 Rechtsbeziehungen beim Vertrieb von Open-Source-Software

Wenn Open-Source-Software weitergereicht wird, werden zeitgleich zwei unterschiedliche Rechtsverhältnisse begründet, die sogenannte Softwareüberlassung sowie eine Open-Source-Lizenz:

Softwareüberlassung Dem Empfänger wird die Software zugänglich gemacht. Denkbar

2 Paul (2005) erwähnt z. B. den Straftatbestand der Datenveränderung im Fall von Datenverlusten.

3 Das Gutachten Spindler hat für Aufruhr gesorgt. Der Auftraggeber der Studie, Der *Verband der deutschen Softwareindustrie (VSI)*, interpretierte die Studie mit dem folgenden Kommentar: „Der VSI sieht [...] bestätigt, dass aus rechtswissenschaftlicher Perspektive Risiken beim Einsatz von OSS existieren. Diese haben eine unmittelbare unternehmenskritische Relevanz, da erhebliche wirtschaftliche Nachteile durch die mangelhafte Durchsetzung urheber-, vertrags- und haftungsrechtlicher Ansprüche entstehen können.“ Diese Meldung provozierte viele Reaktionen, vgl. etwa Knop (2003).

4 Zum deutschen Recht: Bürgerliches Gesetzbuch (BGB), Gesetz über die Haftung für fehlerhafte Produkte (ProdHaftG). Zum schweizerischen Recht: Obligationenrecht (OR, SR. 220), Bundesgesetz über die Produkthaftpflicht (PrHG, SR. 221.112.944).

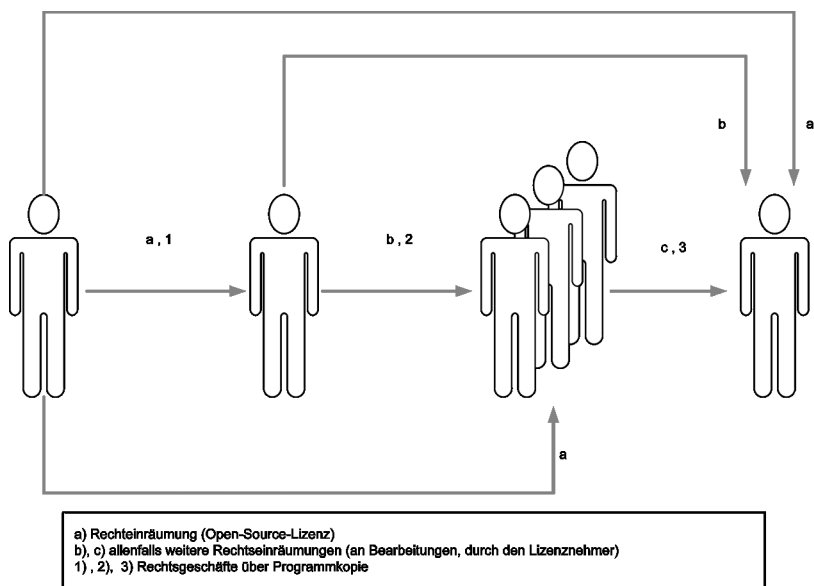


Abbildung 1: Rechtsverhältnisse bei Open-Source-Software

ist, dass ihm die Software auf einem Datenträger zugestellt wird, dass die Software in ein Hardware-Produkt integriert ist (sog. Embedded-Systeme) oder dass der Nutzer die Software per Download von der Webseite des Entwicklers oder eines Dritten⁵ bezieht. Grundlage für die Softwareüberlassung ist ein Vertrag.

Open-Source-Lizenz. Da das derart übermittelte Programm ohne eine Open-Source-Lizenz nur in engem Rahmen genutzt werden kann, bedarf es zur Ausübung der für Open-Source-Software typischen Freiheiten zusätzlich einer Lizenz, einer Open-Source-Software-Lizenz.⁶

Die erwähnte Zweiteilung wird in Abbildung 1 dargestellt. Die Rechtseinräumungen durch eine Open-Source-Lizenz werden mit Kleinbuchstaben bezeichnet, die Verträge zur Softwareüberlassung mit arabischen Ziffern.

⁵ Ein Beispiel wäre der Download von einer öffentlichen Plattform, wie beispielsweise *SourceForge* (<http://sourceforge.net>) oder *Savannah* (<http://savannah.gnu.org>).

⁶ Zur vertragsrechtlichen Einordnung auch des Lizenzverhältnisses vgl. das Urteil des Landgerichts Frankfurt a. M. vom 6. September 2006, Geschäfts-Nr. 2-6 0 224/06, S. 3, abrufbar unter http://www.jbb.de/urteil_lg_frankfurt_gpl.pdf; Entscheidung des Landgerichts München I vom 19. Mai 2004, Az 21 0 6123/04, S. 13, abrufbar unter http://www.jbb.de/urteil_lg_muenchen_gpl.pdf. Zur Diskussion in den USA siehe Madison (2003, 296 ff.) und Wacha (2005).

3 Bedeutung der Anonymität in der Vertragsbeziehung

Über das ihm übermittelte Softwarepaket steht der Empfänger, meist ohne dass er sich dessen bewusst ist, in mehreren Rechtsverhältnissen zu einer Vielzahl von Entwicklern. Für das Open-Source-Lizenzverhältnis sind sowohl die Unentgeltlichkeit als auch die Anonymität der Vertragsbeziehung typisch.⁷ Für die Softwareüberlassung ist zwischen *passivem* und *aktivem* Vertrieb zu differenzieren.

3.1 Passiver Vertrieb

Die Softwareüberlassung ist oft ebenfalls anonym. Dies ist dann der Fall, wenn die Software von einer öffentlichen Plattform heruntergeladen wird, sei es von derjenigen des Entwicklers, sei es von jener eines Dritten. Typischerweise erfolgt der Download von diesen Seiten, ohne dass der Entwickler dies bemerkt. Der Download erfolgt auf Initiative des Empfängers der Software. Diese Form des Vertriebs von Open-Source-Software kann als *passiver Vertrieb*⁸ bezeichnet werden. An der Qualifikation als passiver Vertrieb ändert sich auch dann nichts, wenn ein Komitee über die Einbindung von Programmkomponenten entscheidet;⁹ maßgeblich ist die Form der Verbreitung.

Wer Open-Source-Software entgegennimmt, wird allein durch diese Tatsache nicht verpflichtet, gleichwohl liegt der Softwareüberlassung ein Vertragsverhältnis zu Grunde. Die rechtliche Würdigung dieser Vertragsbeziehung ist in der Lehre heftig umstritten.¹⁰ Darauf muss vorliegend jedoch nicht weiter eingegangen werden, da die rechtliche Qualifikation der Vertragsbeziehung die produkthaftungsrechtliche Diskussion nicht beeinflusst.

7 Oft ergibt sich die Nutzungsberechtigung des Empfängers nicht aus dem Lizenzvertrag, sondern bereits aus dem Gesetz (vgl. Art. 12 Abs. 2 URG; Art. 17 Abs. 1 URV; § 69d Abs. 1 UrhG; Art. 5 Nr. 1 der EG-Richtlinie 91/250/EWG vom 14. Mai 1991 über den Rechtsschutz von Computerprogrammen). Erst wenn der Empfänger Nutzungshandlungen vornimmt, die ohne Lizenz verboten wären, kann angenommen werden, dass zusätzlich zum Softwareüberlassungsvertrag auch ein Lizenzvertrag zu Stande kommt. Hierzu siehe Hoeren (2004) und Jaeger und Metzger (2006, Rz. 204).

8 In der Lehre wird oft der Begriff des *isolierten Vertriebs* verwendet (vgl. etwa Spindler 2004); generell Taeger (1995, S. 211). Da dieser Begriff eher die Trennung des Programms von anderen Programmkomponenten als die Unentgeltlichkeit und Anonymität der Vertragsbeziehung – worauf es ankommt – unterstreicht, wird hier zur Abgrenzung ein anderer Begriff geprägt.

9 Dies ist von Bedeutung im Zusammenhang mit der Haftungsexposition des Komitees (vgl. Spindler 2004). Zu erwähnen ist, dass die meisten Projektleitungen heute in Form einer Stiftung, zum Beispiel <http://www.eclipse.org/org>, oder einer Gesellschaft, vgl. <http://www.mozilla.org/foundation> organisiert sind.

10 Zur Diskussion, ob die Softwareüberlassung als Schenkung zu behandeln ist vgl. die folgende Literatur: Jaeger und Metzger (2006, Rz. 206, Fn. 567–570) mit Hinweisen auf Judikatur und Lehre; Deike (2003, S. 14 f.); Schiffner (2003, S. 255) – beschränkt auf körperlichen Übertragungsvorgang, für Download: Gefälligkeitsverhältnis; Spindler (2004); Koch (2000, S. 335); Lenhard (2006, S. 376); Weber (2002b, S. 57); Weber (2003b). Gegen Schenkungsrecht: Hoeren (2004, S. 238) und Sester (2000, S. 799 ff.).

3.2 Aktiver Vertrieb

Wird Open-Source-Software demgegenüber von einem spezialisierten Dienstleister (z. B. RedHat oder Novell) vertrieben oder hergestellt, wird die Anonymität überwunden. Der Anbieter der Software wird den Empfänger kennen oder zumindest nach geringerem Aufwand namentlich bestimmen können. Zur Abgrenzung kann man hier von *aktivem Vertrieb* sprechen. Im aktiven Vertrieb sind die Leistungen des Anbieters in der Regel entgeltlich.

4 Allgemeines zur Produkthaftung für Open Source

Aus Sicht des Verbraucherschutzes besteht das berechtigte Anliegen, den modernen Industriegefahren flächendeckend zu begegnen. Diese Gefahren können auch durch Software hervorgerufen werden, was heute keiner besonderen Erläuterung mehr bedarf.

4.1 Grundlagen

Es besteht vermehrt die Tendenz, Software als ein Produkt im Sinne der Produkthaftungspflichtgesetze (in der Schweiz: PrHG; in Deutschland: ProdHaftG) zu begreifen, obwohl Software in den maßgeblichen Gesetzesbestimmungen nicht ausdrücklich als Gegenstand der Produkthaftungspflicht genannt wird.¹¹ Diese Tendenz wird im vorliegenden Beitrag nicht hinterfragt, obwohl die Diskussion noch im Gang ist. Nach dem Produkthaftungsrecht haftet der Hersteller eines Produkts für Körperschäden (Art. 1 Abs. 1 lit. a PrHG; § 1 Abs. 1 ProdHaftG) und auch für Sachschäden, sofern sie nicht im gewerblichen Bereich des Geschädigten eintreten (Art. 1 Abs. 1 lit. b PrHG; § 1 Abs. 1 ProdHaftG). Wenn in der Lehre Gesundheitsschädigungen mit hohen Haftungsobergrenzen erwähnt werden, ist dies nicht falsch. Damit wird aber eher das theoretisch mögliche Risiko und nicht die wahrscheinlichste Risikoexposition beschrieben.¹² Hauptanwendungsfall dürfte vielmehr der Datenverlust bzw. die Daten-

11 Das gilt für proprietäre Programme gleich wie für Open-Source-Software. Zur Rechtslage in der Schweiz: Straub (2002, Rz. 13 ff.). Zur Rechtslage in Deutschland: Taeger (1995, S. 160 ff.); Spindler (2004, S. 195); Jaeger und Metzger (2006, Rz. 227).

12 Wenn ein Hersteller eines medizinischen Steuergeräts vorbestehende Open-Source-Software einsetzt, so haftet jedenfalls er – sowohl aus Vertrag als auch aus Produkthaftungspflicht – für durch Fehlfunktionen entstandenen Schaden. Das Spital wird zusammen mit den Geräten (samt Software) finanzielle Sicherheiten verlangen. Es ist deshalb unvorstellbar, dass das Spital im Schadensfall auch nur entfernt daran denken würde, auf jemand andern als den Vertragspartner (bzw. die von ihm gestellte Sicherheit) zuzugreifen. Wie dem auch sei: Die meisten Open-Source-Lösungen für diesen Bereich beschränken sich heute soweit ersichtlich nach wie vor auf Programme zur Verwaltung von Labor- und Patientendaten (vgl. <http://resmedicinae.sourceforge.net/analysis/littlefish/links/fishlinks.htm>; <http://librehealth.sourceforge.net/inventory.html>), gelegentlich finden sich Anwendungen mit erweiterten Funktionen (z. B. für Telepathologie/-medizin, <http://ipath.sourceforge.net/>).

veränderung auf der Festplatte des privat genutzten Computers sein.¹³ Ob Datenverlust oder Datenänderung jedoch als Sachschaden im Sinne des Produkthaftungsrechts gelten kann, wird kontrovers diskutiert.¹⁴ Es ist also denkbar, dass das Produkthaftungsrecht auf die in der Praxis häufigste Fallkonstellation gar nicht anwendbar ist.

4.2 Herrschende Lehre

In der herrschenden Lehre wird die Produkthaftpflicht auch für Open-Source-Software grundsätzlich bejaht (siehe Spindler (2004, S. 194 ff.), Jaeger und Metzger (2006, Rz. 225 ff.) und Schiffner (2003, S. 258 ff.)). Abgelehnt wird sie nur dort, wo der Entwickler das Programm in der Freizeit geschrieben hat und Hinweise auf eine besondere berufliche Qualifikation des Entwicklers fehlen (Spindler 2004, S. 202; Jaeger und Metzger 2006, Rz. 229). Würde die strikte Produkthaftpflicht aber – mit der herrschenden Lehre – abgesehen von wenigen Ausnahmen immer auch für Open-Source-Software greifen, so würde dies für die vielen Anbieter, die Software im passiven Vertrieb anbieten, ein nicht kontrollierbares Risiko darstellen.¹⁵ Im Falle der Fehlerhaftigkeit von Open-Source-Software könnte ein jeder Entwickler haftbar gemacht werden; denn die Haftungsausschlüsse in den Open-Source-Software-Lizenzen können den Produkthaftungsanspruch nicht einschränken oder beseitigen (Art. 8 PrHG und § 14 ProdHaftG; vgl. z. B. Zirlick (2003, S. 202)). Dieses Resultat weckt Bedenken, auf die der folgende Abschnitt näher eingeht.

4.3 Bedenken

Die Produkthaftpflicht passt nicht zum passiven Vertrieb.¹⁶ Dies wird ersichtlich, wenn die zur Rechtfertigung der Produkthaftpflicht vorgetragenen Argumente daraufhin untersucht werden, ob sie auch für Open-Source-Software im passiven Vertrieb herangezogen werden können.

Risikoallokation Unter diesem Stichwort wird angeführt, der Softwareentwickler als *cheapest cost avoider* (vgl. Morscher (1999, S. 66) und Ann (1991, S. 115)) könne den Schaden am besten vermeiden, also habe er ihn auch zu tragen. Dieses Argument überzeugt im passiven Vertrieb jedoch gerade nicht. Der Entwickler, der seinen Quelltext auf ein öffentliches Repository lädt, kann nicht voraussehen, wozu der Nutzer die Software verwenden wird. Allokationsüberlegungen

13 Dies erklärt, weswegen Gerichtsentscheide zur Produkthaftung bislang fehlen. Schäden an privat genutzten Gegenständen haben meist einen zu geringen Wert, als dass der Geschädigte den Aufwand und das Prozessrisiko eines Gerichtsverfahrens auf sich nehmen würde.

14 Vgl. zum Stand der Lehre, ob Datenverlust als Sachschaden qualifiziert: Bartsch (1998), Günther (2001), Schiffner (2003), Straub (2004), Weber (2003a).

15 Spindler (2004, S. 203) weist zu Recht darauf hin, dass für solches Tätigwerden heute kein Versicherungsschutz erhältlich scheint.

16 Vgl. zur hier verwendeten Begriffsbestimmung Abschnitt 3.1

müssen bei Open-Source-Software vielmehr dazu führen, dass der Entwickler (bei passivem Vertrieb) grundsätzlich nicht haftet.

Voraussehbarkeit des Schadenseintritts Hierzu wird Folgendes angeführt:

„Der Schaden sollte nicht demjenigen verbleiben, der mehr oder weniger zufällig von ihm betroffen wird, sondern von demjenigen getragen werden, der das Schadensrisiko wenigstens im allgemeinen beherrscht.“ (Taeger 1996, S. 270)

Sofern der Geschädigte die schädigende Software nicht selber heruntergeladen hat, mag zutreffend sein, dass er den Schadenseintritt nicht voraussehen konnte. Dieses Argument führt jedoch nicht dazu, dass der Entwickler im passiven Vertrieb zur Rechenschaft zu ziehen ist. Der Entwickler gibt die weitere Verwendung der Software aus der Hand und kann sie nicht kontrollieren (Lutterbeck et al. 2000, S. 110). Ein allfälliger Schadenseintritt ist für ihn in keiner Weise voraussehbar. Eine Software kann mit Blick auf die eine Verwendungsart fehlerfrei sein, während sie für eine andere untauglich ist. Das Beispiel des Ariane-5-Desasters verdeutlicht dies: Das Programm, welches schließlich zur Zerstörung der *Ariane 5* führte, funktionierte jahrelang reibungslos – in der Vorgängerrakete *Ariane 4*. Das Programm war aber für die *Ariane 5* wiederverwendet worden und verursachte dort falsche Resultate (vgl. Lions 1996). Das Testen eines Programms auf seine Funktionsfähigkeit hin setzt demnach voraus, dass die Spezifikation für den konkreten Einsatzbereich bekannt ist. Das ist im passiven Vertrieb von Open-Source-Software regelmäßig nicht der Fall. Der Entwickler von Open-Source-Software sollte deshalb nicht haften, wenn er die Software nur passiv anbietet.

Nutzen Unter diesem Aspekt wird angeführt, der Schaden solle von demjenigen getragen werden, der den wirtschaftlichen Nutzen aus dem Gebrauch eines Wirtschaftsguts hat. Die Tatsache, dass der Empfänger die Software gebraucht, nützt nun aber gerade beim passiven Vertrieb nur dem Nutzer, nicht aber dem Entwickler; denn der Entwickler erhält keine Gegenleistung für die Softwareüberlassung.¹⁷ Auch diese Überlegungen führen dazu, die Produkthaftung des Entwicklers von Open-Source-Software zu verneinen, wenn der Entwickler die Software nur passiv anbietet.

Versicherungsmöglichkeit Die meisten privaten Haftpflichtversicherungen decken das aus dem Upload von Software auf das Internet entstehende Schadensersatzrisiko in der Regel nicht ab.¹⁸ Entsprechend geht das zur Begründung der Produkt-

17 Vgl. Abschnitt 3.1.

18 Siehe Fußnote 15.

haftpflicht jeweils angeführte Argument, wonach der Entwickler das Risiko auf eine Versicherung abwälzen könne, ins Leere.

Transaktionskosten Unter diesem Argument wird im Allgemeinen versucht, die Entscheidung eines kommerziellen Softwareanbieters mit traditioneller wirtschaftlicher Logik nachzuzeichnen. Übersteigen die Kosten der Produktion (samt Versicherung) die erwarteten Einnahmen, so werde der Entwickler von der – wirtschaftlich demzufolge ineffizienten – Softwareentwicklung absehen. Auch diese Überlegung ist für Open-Source-Software im passiven Vertrieb nicht angebracht, denn sie lässt die Motivationen zur Herstellung von Open-Source-Software außer Acht. Dieser Begründungsansatz würde bei Open-Source-Software in allen denkbaren Konstellationen zum Schluss führen, dass es Open-Source-Software nicht geben dürfe, denn die Bejahung der Produkthaftung könnte viele Entwickler davon abhalten, sich weiter an Open-Source-Entwicklungsprojekten zu beteiligen. Das genannte Argument führt zu nicht haltbaren Resultaten.

Die Analyse der traditionellen Begründungen für das Produkthaftpflichtrecht lässt nur den Schluss zu, dass das Produkthaftpflichtrecht für Open-Source-Software (im passiven Vertrieb) nicht das richtige Instrument ist.¹⁹ Nach dem Produkthaftungsrecht gilt der Umstand als haftungsbegründend, dass ein Produkt in Verkehr gebracht wurde, obwohl es für den vorgesehenen Zweck nicht tauglich war:

„Produkthaftung setzt ein im Inverkehrbringen eines fehlerhaften Produkts liegendes rechtswidriges Verhalten [...] voraus.“ (Taeger 1995, S. 102)

Für den passiven Vertrieb ist nun entscheidend, dass der Entwickler ganz augenscheinlich nicht in der Lage ist, das Programm für den schließlich verwendeten Einsatzzweck tauglich zu machen – und zwar mangels Vorausssehbarkeit. Der Nutzer des Programms hätte umgekehrt aber alle notwendigen Instrumente hierzu in der Hand (insbesondere Einsicht in den Quelltext). Deshalb darf allein die Tatsache, dass Open-Source-Software auf einer öffentlich zugänglichen Plattform zum Download angeboten wird, keine Produkthaftung auslösen.

Es ist hier nicht der Ort, die Grundlagen der Produkthaftpflicht dogmatisch lückenlos aufzubereiten. Die nachstehende Darstellung beschränkt sich deshalb darauf, die wichtigsten Argumente darzustellen, die aus Entwicklersicht gegen die Anwendbarkeit des Produkthaftpflichtrechts angeführt werden können.

¹⁹ Spindler (2004, S. 203) räumt zwar ein, dass die Produkthaftpflicht im Einzelfall durchaus unpassend sein könne, ist aber der Auffassung, dass Verallgemeinerungen nicht möglich seien. Vielmehr müsse *im Zweifel* davon ausgegangen werden, dass ein wirtschaftlicher Zweck vorliege und damit das Produkthaftpflichtrecht zur Anwendung komme.

5 Keine Produkthaftungspflicht im passiven Vertrieb

Nachstehend wird dargelegt, welche Abwehrargumente dem Entwickler zur Verfügung stehen, um die unerwünschten Nachteile abzuwenden, die sich bei Anwendbarkeit des Produkthaftungspflichtenrechts auf Open-Source-Software ergeben würden. Die Abwehrargumente beruhen im Wesentlichen auf dem Umstand, dass für den Entwickler von Open-Source-Software in der Regel nicht voraussehbar ist, wie sein Entwicklungsbeitrag schließlich verwendet wird.²⁰ Es kann differenziert werden danach, ob der Geschädigte die Software direkt von der Webseite des Entwicklers bezogen hat (dazu Abschnitt 5.2) oder nicht (dazu Abschnitt 5.1).

5.1 Schädigung eines Dritten ohne Vertragsverhältnis

Steht der Geschädigte in keinem Vertragsverhältnis zum Entwickler, kann der Entwickler gegen die Produkthaftung insbesondere die folgenden Argumente anführen:

Information objektiv ungefährlich Zumindest, wenn die Software nur im Quelltext vorliegt, kann mit guten Gründen gegen die Anwendung des Produkthaftungsrechts argumentiert werden. Quelltext ist noch viel stärker als kompilierte Software bloße Information (Weber 2002a, S. 1016). Bloße Informationen sollten keine Produkthaftung auslösen, solange sie nicht in ein Produkt mit Schadenspotenzial eingebunden worden sind. Oft kann der Quelltext ohne vorgängige Kompilation gar nicht dazu eingesetzt werden, eine Datenverarbeitungsanlage anzutreiben.²¹ Trifft dies zu, ist die im Quelltext verkörperte Information objektiv ungefährlich.²² Wer bloße Information zur Verfügung stellt, sollte deshalb noch nicht als Hersteller eines Teilprodukts behandelt werden (vgl. Günther 2001, S. 241 mit Verweisen). Einschränkend ist Folgendes zu erwähnen: Da Computerprogramme heute immer häufiger direkt nur im Quellcode installiert werden und die Maschine den Code erst im Runtime-Modus kompiliert, dürften die Anwendungsfälle für dieses Argument bei realistischer Betrachtung immer stärker zurückgehen.

Fehlende Produzenteneigenschaft Open-Source-Software befindet sich in einem laufenden Entwicklungsprozess, dessen Abschluss nicht vorausgesagt werden kann. Dies ist ein wesentlicher Unterschied zur proprietären Software, bei der es im Belieben des Softwareunternehmens liegt, den Abschluss des Entwicklungsprozesses und damit die Marktreife zu bestimmen. Hat ein Dritter die auf der Entwicklerplattform verfügbare Software heruntergeladen und anschließend bis zur Marktreife gebracht, muss dieser als Produzent gelten, nicht aber der Entwickler der *Robversion*. Dies muss jedenfalls dann gelten, wenn ein anderer als der

²⁰ Siehe Abschnitt 4.3.

²¹ Siehe Straub (2004, S. 261) und Straub (2002, Rz. 28).

²² Zur Haftung für reine Information vgl. Straub (2002, Rz. 17 ff.), Hilty (2001, S. 813 ff.).

ursprüngliche Entwickler die Software zur Marktreife gebracht hat. Ein Dritter, meist ein Mehrwertdienstleister, hat die Software übernommen und für einen bestimmten Zweck tauglich gemacht. Wenn der ursprüngliche Entwickler den Einsatzbereich der Software nicht kannte, hat der spätere Entwickler den ursprünglichen faktisch vollständig substituiert. Mit anderen Worten ist der nachfolgende Mehrwertdienstleister an die Stelle des ursprünglichen Entwicklers getreten und hat diesen ersetzt. Der Entwickler solcher isoliert vertriebener Open-Source-Software kann unter diesen oft vorliegenden Umständen nicht als Produzent gelten. Mit Blick auf den heutigen Stand in der Literatur ist dieses Argument allerdings problematisch. Auch ein Entwickler von Bestandteilen des schließlich fehlerhaften Produkts gilt nach dem Produkthaftpflichtrecht üblicherweise als Produzent (vgl. Fellmann (1996, N 6 f.) zu Art. 2 PrHG; Straub (2002, Rz. 71 ff.)), eine Position, die in Bezug auf Open-Source-Software jedenfalls überdacht werden sollte.

*Nicht in Verkehr gebracht*²³ In den Verkehr wird die Software dann gegeben, wenn sie mit dem Ziel der Vermarktung aus dem eigenen Herrschaftsbereich herausgegeben wird (siehe Taeger (1996, S. 265); Straub (2002, Rz. 132 ff.); Fellmann (1996, N 2 f.) zu Art. 5 PrHG). Mit dem Abrufbarmachen auf einer öffentlich zugänglichen Webseite hat der Entwickler die Software zweifelsohne aus seinem Herrschaftsbereich entlassen. Ob dies zur Vermarktung gegenüber einem Konsumenten erfolgte, ist abhängig davon, ob die Software *tel quel* vom Konsumenten verwendet werden kann oder ob das Programm erst noch in ein weiteres Programm eingebunden werden muss, um eingesetzt werden zu können.²⁴ Sind weitere Entwicklungsarbeiten oder auch nur die Verknüpfung mit einem Programm erforderlich, kann wiederum mit guten Gründen gegen die Anwendbarkeit der Produkthaftung argumentiert werden.²⁵

Fehlende Voraussehbarkeit Wenn Open-Source-Software passiv vertrieben wird, kann jedermann die Software herunterladen. Der Entwickler kann somit in der Regel nicht voraussehen, auf welchen Plattformen und in welchem Installationsumfeld seine Software eingesetzt werden wird; die im Zusammenhang mit der Software möglicherweise entstehenden Risiken sind für den Entwickler deshalb nicht im Voraus erkennbar. Umgekehrt hat es der Empfänger der Software in der Hand, die Risiken des Softwareeinsatzes zu kontrollieren, denn er hat im Gegensatz zu proprietärer Software die Möglichkeit, den Quelltext selber nach Sicherheitsdefiziten zu überprüfen.²⁶ Da der Entwickler ganz anders als

23 Nach Art. 5 Abs. 1 lit. a PrHG; § 1 Abs. 2 Ziff. 1 ProdHaftG.

24 Was zum Beispiel bei Programmbibliotheken der Fall ist.

25 Siehe zur fehlenden Voraussehbarkeit bereits Abschnitt 4.3.

26 Vgl. – allerdings zur Produzentenhaftung/Delikts haftung – Spindler (2004, S. 193); vgl. auch BGH vom 5. Mai 1992: VI ZR 188/91, NJW 1992, 2016, 2018; BGH vom 14. Mai 1996: VI ZR NJW 1996, 2224, 2226.

der Empfänger der Software die Art der Verwendung seiner Software nicht voraussehen kann, darf der Fehlerbegriff für passiv vertriebene Open-Source-Software nur in Ausnahmefällen bejaht werden. Ein solcher Ausnahmefall kann angenommen werden, wenn die Software für einen ganz bestimmten Einsatz tauglich gemacht wurde und sich der Nutzer auf die Tauglichkeit zu diesem Zweck verlassen durfte, z. B. wenn die Open-Source-Software als selbständig lauffähiges *executable* verfügbar gemacht wird und es abgesehen von der Installation durch den Nutzer keiner weiteren Schritte bedarf, damit die Software auf einem bestimmten System einsetzbar ist (§ 3 Abs. 1 lit. b ProdHaftG).²⁷ Produkthaftpflicht kann sich auch daraus ergeben, dass der Entwickler der Open-Source-Software seine Software besonders anpreist und so besondere Sicherheitserwartungen weckt, z. B. mit Mitteilungen im Umfeld der Download-Möglichkeit (vgl. Art. 4 Abs. 1 lit. a PrHG; § 3 Abs. 1 lit. a ProdHaftG). Fehlt es an solchen qualifizierenden Umständen, muss angenommen werden, dass eine Fehlfunktion und somit ein Fehler nicht so sehr mit der Open-Source-Software selber verknüpft ist, sondern vielmehr deshalb entstanden ist, weil die Software in einer für den Entwickler nicht voraussehbaren Art und Weise eingesetzt wurde. Sollte ein Fehler gleichwohl bejaht werden, sind jedoch die Haftungsausschlussgründe gemäß § 1 Abs. 2 Ziff. 2 ProdHaftG bzw. Art. 5 Abs. 1 lit. b PrHG heranzuziehen. Nach diesen Bestimmungen ist die „Ersatzpflicht des Herstellers [...] ausgeschlossen, wenn nach den Umständen davon auszugehen ist, daß das Produkt den Fehler, der den Schaden verursacht hat, noch nicht hatte, als der Hersteller es in den Verkehr brachte“ (§ 1 Abs. 2 Ziff. 2 ProdHaftG). Der Entwickler ist beweispflichtig, dass dieser Haftungsausschlussgrund zur Anwendung kommt (Fellmann 1996, N 1 zu Art. 5 PrHG). Nach Art. 5 Abs. 1 PrHG muss der Entwickler zwar den Beweis der Fehlerlosigkeit antreten. Unseres Erachtens muss dieser Beweis aber bereits dann gelingen, wenn der Entwickler nachweist, dass der Fehler teilweise auf mangelnde Kompatibilität mit dem System zurückzuführen ist, auf welchem die Software zum Einsatz kam. Ist der Fehler auf mangelnde Kompatibilität der Software mit anderen Programmen oder der Systemumgebung zurückzuführen, liegen Umstände vor, die zeigen, dass der Fehler noch nicht vorlag, als das Programm in Verkehr gebracht wurde.

27 Eine bloße Programmbibliothek kann demnach nicht als installationsfähiges Produkt angesehen werden. Als Anwendungsbeispiel für ein vom Nutzer direkt installierbares Programm kann aber die OpenOffice-Suite angeführt werden, die in lauffähiger Version zum Download unter der Website <http://www.openoffice.org> angeboten wird. Das Risiko ist in diesen Fällen allerdings beschränkt, da die durch das Produkthaftungsrecht geschützten Schäden (Körperschaden) durch diese Software wohl kaum verursacht werden können. Theoretisch besteht ein Restrisiko nur hinsichtlich Datenverlusten von der Festplatte des privaten Nutzers. Ob Datenverlust als Sachschaden im Sinne des Produkthaftungsrechts qualifiziert werden kann, ist jedoch noch nicht restlos geklärt.

Ungenügend enger Kausalszusammenhang Wenn ein anderer als der ursprüngliche Entwickler die auf der öffentlichen Plattform angebotene Open-Source-Software zur Marktreife gebracht hat, ist die Kausalkette vom Fehler bis zur Haftung in aller Regel so lang, dass der ursprüngliche Beitrag des Entwicklers hinsichtlich der Fehlerverursachung in den Hintergrund tritt.

Software nicht im Rahmen der beruflichen Tätigkeit hergestellt Je nach Fallkonstellation kann der Entwickler mit dem Argument durchdringen, er habe nicht „für den Verkauf oder eine andere Form des Vertriebs mit wirtschaftlichem Zweck hergestellt“ (Art. 5 Abs. 1 lit. c PrHG; vgl. auch § 1 Abs. 2 Ziff. 3 ProdHaftG; siehe Spindler (2004, S. 202) sowie Jaeger und Metzger (2006, Rz. 229)). Dabei sollte bejaht werden, dass beim Angebot zum kostenlosen Download in der Regel kein Vertrieb mit wirtschaftlichem Zweck vorliegt (vgl. Jaeger und Metzger (2006, Rz. 228) und Spindler (2004, S. 203)).

5.2 Bei Schädigung des Vertragspartners

Ist der direkte Vertragspartner geschädigt, so ist ergänzend das Argument entlastend, dass der Vertragspartner in die Software keine berechtigten Sicherheitserwartungen haben durfte, denn ein Fehler kann von vornherein nur dann vorliegen, wenn berechnete Sicherheitserwartungen enttäuscht wurden (Taeger 1995, S. 102). Maßgeblich ist, ob das Produkt zum Gebrauch tauglich ist, mit dem vernünftigerweise gerechnet werden darf (Art. 4 Abs. 1 PrHG; § 3 Abs. 1 ProdHaftG).

Fehlerhaftigkeit im Sinne des Produkthaftpflichtrechts ist ein unbestimmter Rechtsbegriff. Erst im Konfliktfall stellt ein Richter fest, ob die Software von einem wie auch immer zu erwartenden Sicherheitsstandard abweicht oder nicht. Der Begriff der Fehlerhaftigkeit ist produkthaftungsrechtlich autonom auszulegen (siehe Straub (2002, Rz. 32) und Spindler (2004, S. 203 f.)). Dabei wird auf die Erwartungshaltung der Allgemeinheit abgestellt (siehe Straub (2002, Rz. 33 f., 39 ff.) und Spindler (2004, S. 203 f.)). Es ist ein objektiver Maßstab anzulegen (vgl. Fellmann 1996, N 3 zu Art. 4 PrHG). Entscheidend ist, welches Maß an Sicherheit eine verständige Person in Kenntnis und bei Abwägung aller relevanten Umstände für erforderlich hält. Von der Produkthaftung ausgeschlossen sind demnach Fehler, die nach dem „Stand der Wissenschaft und Technik“ zum Zeitpunkt des Inverkehrbringens nicht erkannt werden konnten (Taeger 1996, S. 267). Bei Open-Source-Software kann dem Entwickler deshalb die Tatsache zugute kommen, dass die Entwicklung oft in einer großen Community diskutiert wird. *Community building* (Stürmer und Myrach 2006, S. 219 ff.) erhält hier eine zusätzliche, haftungsbeschränkende Komponente (Lemyre 2002, Rz. 105): Wenn eine ganze Vielzahl von Entwicklern den Fehler nicht bemerkt hat, so kann der Fehler nicht im Widerspruch zum Stand der Wissenschaft und Technik angesehen werden. Der Entwickler wird gleichwohl Vorsorge treffen müssen. Wenn er im Konfliktfall nachweisen kann, dass er die Software auf die kritischsten Fehler hin geprüft hat,

steht er deutlich besser da. Außerdem empfiehlt es sich zu dokumentieren, wie groß die Entwicklercommunity zum Zeitpunkt des Uploads der Open-Source-Software war und wie lebhaft die Diskussion fremder Codebestandteile zu diesem Zeitpunkt geführt wurde.

Dass vor der Marktreife berechnete Sicherheitserwartungen in passiv vertriebene Open-Source-Software gesetzt werden könnten, ist auch deshalb ausgeschlossen, weil heute dem breiten Publikum bekannt sein dürfte, dass Open-Source-Software ständig weiterentwickelt wird und die Entwicklergemeinschaft deshalb keine Haftung und Gewährleistung für Funktionalität und Fehlerlosigkeit übernehmen will. Zusammenfassend kann basarartig entwickelte Open-Source-Software nicht gleichermaßen unter den Produktbegriff des Produkthaftpflichtrechts fallen wie proprietäre Software.

6 Produkthaftpflicht im aktiven Vertrieb

Im aktiven Vertrieb bestehen diese Argumente grundsätzlich nicht, auch wenn Open-Source-Software schadensverursachend war. Hier dürfte die Produkthaftpflicht grundsätzlich gleich zum Tragen kommen wie beim herkömmlichen proprietären Softwarevertrieb (Straub 2004). Wesentlich ist der Unterschied, dass der Anbieter der Open-Source-Software die Verwendungsart der Software kannte oder voraussehen konnte.

Bei genauer Betrachtung kann die Haftung aus PrHG/ProdHaftG allerdings auch nach aktivem Vertrieb nicht immer greifen. Open-Source-Software zeichnet sich dadurch aus, dass sie vom Empfänger weiterentwickelt werden darf. Das heißt, dass der Anbieter von Open-Source-Software im aktiven Vertrieb nicht gleichermaßen Kontrolle über das weitere Schicksal der Software hat wie ein Hersteller von proprietärer Software. Dieser Unterschied wirkt sich auf die produkthaftungsrechtliche Würdigung aus. Macht der Anwender von den aus einer Open-Source-Software-Lizenz abgeleiteten Freiheiten Gebrauch, müssen dem Anbieter wiederum die bereits erwähnten Einreden gegen eine Haftpflicht stehen offen (vergleiche Abschnitt 5). Mit anderen Worten lässt sich die Produkthaftung für Open-Source-Software auch nach aktivem Vertrieb nur dann bejahen, wenn der Anwender die Software gerade nicht wie Open-Source-Software benutzt. Wird aktiv vertriebene Open-Source-Software vom Empfänger weiterentwickelt, darf der Anbieter der ursprünglichen Version nicht verantwortlich gemacht werden für das Schädigungspotenzial, das von der weiterentwickelten Software ausgeht (a. M. aber Taeger (1996, S. 264 f.); kritisch hierzu Günther (2001, S. 210)). Im Haftungsprozess wird also relevant, ob der Anwender die Software unverändert oder erst nach einer dem Vertrieb folgenden Modifikation eingesetzt hat.

Somit kann ein Anbieter von Open-Source-Software unter Umständen der Produktheftpflicht unterstehen. Dieses Risiko lässt sich vertraglich nicht wegbedingen, da die Produktheftung immun ist gegen die Haftungsausschlussklauseln in Open-Source-Lizenzen. Der Geschädigte muss Ansprüche aus Produkthaftungsrecht innerhalb von drei Jahren nach Schadenseintritt (§ 12 ProdHaftG; Art. 9 PrHG), spätestens

jedoch innerhalb von zehn Jahren seit Inverkehrbringen der Software geltend machen (§ 13 ProdHaftG; Art. 10 PrHG).

Die praktischen Konsequenzen aus der Anwendbarkeit des Produkthaftungsrechts werden sich jedoch in Grenzen halten, da Software meistens „nur“ Datenverluste verursacht, nicht aber Körperschäden. Datenverluste sind – wenn überhaupt – von vornherein nur dann ein nach dem Produkthaftungsrecht ersetzungsfähiger Schaden, wenn die Daten auf einem privat genutzten Computer gespeichert waren. Sollte Open-Source-Software aber ein erhöhtes Schädigungspotenzial aufweisen (insbesondere, wenn die Möglichkeit von Körperschädigungen nicht ausgeschlossen werden kann), wird der Anbieter sie nur dann freigeben, nachdem er jede einzelne Komponente auf Herz und Nieren geprüft hat – selbstverständlich mit der Folge, dass solche Anwendungen nur gegen einen erhöhten Preis zu haben sind.

7 Zusammenfassung

Das Risiko, dass sich für einen Entwickler von Open-Source-Software aus dem Produkthaftrecht ergibt, dürfte heute überbewertet werden. Die Rechtsordnung hält für Open-Source-Software grundsätzlich befriedigende Lösungen bereit. Der Rechtsanwender muss jedoch bereit sein, die Besonderheiten von Open-Source-Software zu berücksichtigen und auf Wertungsunterschiede einzugehen. Werden diese Unterschiede berücksichtigt, gelangt man zum Resultat, dass das Konzept der Produkthaftung auf Open-Source-Software nicht recht passt.

Im passiven Vertrieb kann es nur dann zur Produkthaftung kommen, wenn Open-Source-Software entweder als selbstständig laufende Anwendung (*executable*) verfügbar gemacht wird oder wenn eine Sondersituation vorliegt, insbesondere wenn der Anbieter die Software besonders anpreist und so erhöhte Sicherheitserwartungen weckt.

Auch bei aktiv vertriebener Open-Source-Software kann das Konzept der Produkthaftung nicht unbesehen angewendet werden. Produkthaftung scheidet insbesondere dann aus, wenn Open-Source-Software nach dem Vertrieb modifiziert wurde.

Literatur

- Ann, C. (1991), *Die Produkthaftung des Lizenzgebers: Zur Haftungssituation von Innovationsverträgen im deutschen und im amerikanischen Recht*, Heymann, Köln.
- Bartsch, M. (1998), *Software und das Jahr 2000 – Haftung und Versicherung für ein technisches Großproblem*, Nomos Verlagsgesellschaft, Baden-Baden.
- Deike, T. (2003), 'Open Source Software – IPR-Fragen und Einordnung ins Deutsche Rechtssystem', *Computer und Recht* **19**(1), S. 9–18.

Produkthaftung für Open-Source-Software?

- Fellmann, W. (1996), Kommentar zum Produkthaftungspflichtgesetz, in H. Honsell, N. P. Vogt und W. Wiegand (Hrsg.), 'Obligationenrecht I, Basler Kommentar zum schweizerischen Privatrecht', 2. Aufl., Helbing & Lichtenhahn, Basel.
- Günther, A. (2001), *Produkthaftung für Informationsgüter. Verlagszeugnisse, Software und Multimedia im deutschen und US-amerikanischen Produkthaftungsrecht*, Verlag Otto Schmidt, Köln.
- Hilty, R. M. (2001), *Lizenzvertragsrecht: Systematisierung und Typisierung aus schutz- und schuldrechtlicher Sicht*, Stämpfli, Bern.
- Hoeren, T. (2004), Open Source und das Schenkungsrecht – eine durchdachte Liaison?, in R. Bork, T. Hoeren und P. Pohlmann (Hrsg.), 'Recht und Risiko', Bd. 2, Verlag Versicherungswirtschaft, Karlsruhe, S. 229–240.
- Jaeger, T. und Metzger, A. (2006), *Open Source Software*, 2. Aufl., C. H. Beck, München.
- Knop, C. (2003), 'Ein Juraprofessor schreckt die Linux-Welt auf', *FAZ* **13. Aug 2003**.
- Koch, F. (2000), 'Urheber- und kartellrechtliche Aspekte der Nutzung von Open-Source-Software', *Computer und Recht* **16**(6), S. 333–344.
- Lemyre, P.-P. (2002), Les logiciels libres sous l'angle de la responsabilité civile, Diplomarbeit, Université de Montréal. <http://www.juriscom.net/uni/visu.php?ID=106> [19. Dez 2006].
- Lenhard, F. (2006), *Vertragstypologie von Softwareüberlassungsverträgen*, Herbert Utz Verlag, München.
- Lions, J. L. (1996), 'Ariane 5 – Flight 501 Failure', Bericht des Untersuchungsausschusses vom 19. Juli 1996. <http://homepages.inf.ed.ac.uk/perdita/Book/ariane5rep.html> [19. Dez 2006].
- Lutterbeck, B., Horns, A. H. und Gehring, R. A. (2000), Sicherheit in der Informationstechnologie und Patentschutz für Softwareprodukt – ein Widerspruch?, Kurzgutachten für den Bundeswirtschaftsminister, TU Berlin.
- Madison, M. (2003), 'Reconstructing the Software License', *Loyola University Chicago Law Journal* **35**(1), S. 275–340. http://www.law.pitt.edu/madison/papers/reconstructing_the_software_license.pdf [18. Dez 2006].
- Morscher, L. (1999), 'Gewährleistung und Haftung bei der Software-Überlassung', *Schweizerische Zeitschrift für Wirtschaftsrecht* **1999**(2), S. 61–77.
- Paul, A. G. (2005), Außervertragliche Haftung für Open Source Software, in 'GI Jahrestagung (1)', S. 131–136. <http://trustsoft.uni-oldenburg.de:19080/30/01/ISOS-2005-Beitrag.pdf> [18. Dez 2006].
- Schiffner, T. (2003), *Open Source Software: freie Software im deutschen Urheber- und Vertragsrecht*, VVF-Verlag, München.
- Sester, P. (2000), 'Open-Source-Software: Vertragsrecht, Haftungsrisiken und IPR-Fragen', *Computer und Recht* **16**(12), S. 797–807.

- Spindler, G. (2003), Rechtsfragen der Open Source Software, Gutachten, Verband der Softwareindustrie Deutschlands e. V., München. http://lehrstuhl-spindler.uni-goettingen.de/extern/ross/downloads/studie_final.pdf [19. Dez 2006].
- Spindler, G. (Hrsg.) (2004), *Rechtsfragen bei Open Source*, Verlag Otto Schmidt, Köln.
- Straub, W. (2002), *Produktehaftung für Informationstechnologiefehler: EU-Produktehaftungsrichtlinie und schweizerisches Produktehaftungsgesetz*, Schulthess Verlag, Zürich.
- Straub, W. (2004), *Informatikrecht*, Stämpfli Verlag, Bern.
- Stürmer, M. und Myrach (2006), Open Source Community Building, in B. Lutterbeck, M. Bärwolff und R. A. Gehring (Hrsg.), 'Open Source Jahrbuch 2006 – Zwischen Softwareentwicklung und Gesellschaftsmodell', Lehmanns Media, Berlin, S. 219–234. http://www.opensourcejahrbuch.de/download/jb2006/chapter_05/osjb2006-05-02-stuermer-myrach [26. Dez 2006].
- Taeger, J. (1995), *Außervertragliche Haftung für fehlerhafte Computerprogramme*, Mohr, Tübingen.
- Taeger, J. (1996), 'Produkt- und Produzentenhaftung bei Schäden durch fehlerhafte Computerprogramme', *Computer und Recht* **12**(5), S. 257–271.
- Wacha, J. (2005), 'Taking the Case: is the GPL enforceable?', *Santa Clara Computer & High Tech Law Journal* **21**, S. 451–492. [19. Dez 2006].
- Weber, R. H. (2002a), Ali Baba oder das Risiko exklusiver Informationsinhaltsrechte, in R. J. Schweizer, H. Burkert und U. Gasser (Hrsg.), 'Festschrift für Jean Nicolas Druey', Schulthess, Zürich, S. 1009–1028.
- Weber, R. H. (2002b), Freie Software – Befreiung vom Vertragstypenkonzept, in F. Harrer, W. Portmann und R. Zäch (Hrsg.), 'Besonderes Vertragsrecht – aktuelle Probleme: Festschrift für Heinrich Honsell zum 60. Geburtstag', Schulthess, Zürich, S. 41–59. <http://www.rwi.unizh.ch/zik/news/Freie%20Software1.pdf> [19. Dez 2006].
- Weber, R. H. (2003a), Haftungsgrundlagen beim IT-Outsourcing, in R. H. Weber, M. Berger und R. Auf der Maur (Hrsg.), 'IT-Outsourcing', Schulthess, Zürich, S. 117–138.
- Weber, R. H. (2003b), Open Source Software – Vertragsgestaltung, in R. H. Weber, M. Berger und R. Auf der Maur (Hrsg.), 'Geschäftsplattform Internet IV', Schulthess, Zürich, S. 73–92.
- Widmer, M. J. (2003), *Open Source Software – Urheberrechtliche Aspekte freier Software*, Stämpfli, Bern.
- Zirlick, B. (2003), *Freizeichnung von der Deliktshaftung*, Stämpfli, Bern.