

Johannes Tränkle
johannes@traenkle.org
<http://www.traenkle.org>

Die Haftung der Mitentwickler von Open Source Software

Stand Januar 2004

Inhaltsverzeichnis

1	Einleitung.....	1
2	Einleitung Open Source Software.....	1
3	Rechtliche und tatsächliche Vorfragen.....	2
3.1	Was wird übertragen?.....	2
3.2	Schuldrechtliche Einordnung.....	3
3.2.1	Schenkung.....	3
3.2.1.1	Entreicherung.....	3
3.2.1.2	Unentgeltlichkeit.....	4
3.2.2	Gesellschaft.....	5
3.2.3	Verfügung.....	6
3.2.4	Ergebnis.....	6
3.3	Zusammensetzung der Entwickler.....	6
3.3.1	Beziehungen der Entwickler untereinander.....	7
3.3.1.1	Basarmethode.....	7
3.3.1.2	Urheberrechtliche Sicht.....	7
3.3.1.2.1	Miturheberschaft.....	7
3.3.1.2.2	Verbundene Werke.....	8
3.3.1.2.3	Bearbeitung.....	8
3.3.1.2.4	Unterscheidungsprobleme.....	8
3.3.1.2.5	Urheber im Arbeitsverhältnis.....	9
3.3.1.3	BGB-Gesellschaft.....	9
3.3.1.4	Ergebnis.....	9
4	Haftungstatbestände	9
4.1	Vertragliche Haftung.....	9
4.1.1	Haftungsausschluss.....	10
4.1.1.1	Einleitung.....	10
4.1.1.2	Verbraucher.....	10
4.1.1.3	Unternehmer.....	10
4.1.1.4	Zwischenergebnis.....	10
4.1.2	Ergebnis.....	11
4.1.3	Angestellte Urheber.....	11
4.2	Verschuldensunabhängige Haftung gemäß dem ProdHaftG.....	12
4.2.1	Produkt.....	12
4.2.1.1	Software auf Datenträger.....	12
4.2.1.2	Download.....	13
4.2.1.3	Application Service Providing (ASP).....	13
4.2.1.4	Zwischenergebnis.....	14
4.2.2	Fehler.....	14
4.2.3	Hersteller.....	16
4.2.4	Schaden und Ausschluss.....	17
4.2.5	Ergebnis.....	19
4.3	Deliktische Haftung.....	19
4.3.1	Gesetzliche Beschränkungen der Haftung, § 521 BGB	19
4.3.2	Verkehrspflichten.....	20
4.3.2.1	Allgemeine Kriterien.....	20
4.3.2.2	Nahe liegender Gebrauch.....	20
4.3.2.3	Vorteil des Verursachers.....	21

4.3.2.4 Preis.....	21
4.3.2.5 Erkennbarkeit.....	23
4.3.2.6 Zwischenergebnis.....	23
4.3.2.7 Person des Verkehrspflichtigen.....	23
4.3.3 Verschulden.....	24
4.3.4 Produkthaftung im Rahmen des § 823 I BGB.....	24
4.3.5 Ergebnis.....	25
4.4 ProdSG.....	25
5 Mehrere Verursacher.....	26
6 Zusammenfassung.....	28

Literaturverzeichnis

- | | |
|-----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cahn, Andreas | Produkthaftung für verkörperte geistige Leistungen
NJW 1996, 2899 |
| Deike, Thies
Wuermeling, Ulrich | Open Source Software: Eine juristische Risikoanalyse
CR 2003, 87 |
| Deike, Thies | Open Source Software: IPR-Fragen und Einordnung ins deutsche Rechtssystem
CR 2003, 9 |
| Heussen, Benno | Free Software – Rechtliche Struktur, Qualität und Haftung
noch nicht veröffentlicht |
| Heussen, Benno | Technische unvermeidbare Software-Fehler – Neue Lösungen durch die Schuldrechtsreform –
noch nicht veröffentlicht |
| Honsell, Heinrich | Produkthaftungsgesetz und allgemeine Deliktshaftung
JuS 1995, 211 |
| Kilian, Wolfgang,
Heussen, Benno,
Hrsg. | Computerrechts-Handbuch
Informationstechnologie in der Rechts- und
Wirtschaftspraxis
Loseblatt
C.H.Beck
Stand Jan. 03, 20. Lfg.
Zitiert: Kilian/Heussen-Bearbeiter |
| Koch, Frank A. | Urheber- und kartellrechtliche Aspekte der Nutzung von Open-Source-Software (I)
CR 2000, 273 |

- Koch, Frank A. Urheber- und kartellrechtliche Aspekte der
Nutzung von Open-Source-Software (II)
CR 2000, 333
- Larenz, Karl Larenz,
Canaris,
Claus-Wilhelm Lehrbuch des Schuldrechts
Zweiter Band
Besonderer Teil
2. Halbband
13. Auflage 1994
C.H.Beck
- Münchener
Kommentar Münchener Kommentar
Bürgerliches Gesetzbuch
- Schuldrecht Allgemeiner Teil
§§ 241-433
3. Auflage, 1994
- Schuldrecht Besonderer Teil I
Band 3
§§ 433-606
3. Auflage, 1995
- Schuldrecht Besonderer Teil III
Band 5
§§ 705-853
3. Auflage, 1997
- C.H. Beck
Zitiert: MüKo-Bearbeiter
- Palandt, Otto Bürgerliches Gesetzbuch
62. Auflage, 2003
C.H. Beck
zitiert: Palandt-Bearbeiter
- Rehbinder, Manfred Urheberrecht
12.Auflage, 2002
C.H.Beck

- Schneider, Jochen Handbuch des EDV-Rechts,
3 . Auflage, 2003,
Verlag Dr. Otto Schmid
- Sester, Peter Open-Source-Software: Vertragsrecht,
Haftungsrisiken und IPR-Fragen
CR 2000, 797
- Spindler, Gerald Rechtsfragen der Open Source Software
Gutachten im Auftrag der Softwareindustrie
Deutschlands e.V. (VSI)
[www.vsi.de/inhalte/aktuell/
studie_final_safe.pdf](http://www.vsi.de/inhalte/aktuell/studie_final_safe.pdf)
- Spindler, Gerald Verschuldensunabhängige Produkthaftung
im Internet
MMR 1998, 119
- Staudinger,
Julius von J. von Staudingers Kommentar zum BGB
mit Einführungsgesetzen und Nebenge-
setzen
Buch 2: Recht der Schuldverhältnisse
§§ 826-829, Produkthaftungsgesetz
Neubearbeitung 2003 von Jürgen Oechsler
Sellier – de Gruyter, Berlin
- Taeger, Jürgen Produkt- und Produzentenhaftung bei
Schäden durch fehlerhafte Programme
CR 1996, 257

1 Einleitung[#]

Ziel dieser Arbeit ist es, die Umstände, unter denen Mitentwickler von Open Source zur Haftung herangezogen werden können, darzustellen. Dabei soll allgemein auf das Thema Open Source Software eingegangen werden, und nicht – zumindest stillschweigend – allein ein Programm als Maßstab herangezogen werden¹. Auch soll die Vielschichtigkeit der Entwicklergemeinde abgebildet werden.

Wie schon Heussen² erwähnt, kann der Programmierer, der Open Source Software weiterentwickeln will, regelmäßig die Risiken, die sich hierbei ergeben, abschätzen, mehr noch, er muss dies letztlich sogar, um überhaupt an eine Weiterentwicklung denken zu können³. Anders stellt sich diese Situation bei einem „einfachen“ Endbenutzer dar. Hier dürfte es – so sie den auftreten – zu den wirklich kritischen Schadensfällen kommen. Denkbar erscheint dies z.B. in einem Fall, in dem Open Source Software als Grundlage für medizinische Geräte dient, die dann wegen der fehlerhaften Software zu einer fehlerhaften Diagnose kommen.

2 Einleitung Open Source Software

Open Source Software wurde insbesondere durch Linux bekannt. Merkmal dieser Software ist, dass der Quellcode mit veröffentlicht wird und der Nutzer Änderungen an diesem Quellcode vornehmen und die so veränderte Software weiterverbreiten darf⁴. Neben Linux ist auch andere Software, wie z.B. die Office-Suite OpenOffice oder der Webserver Apache, Open Source und somit frei.

Um eine Abgrenzung zu Modellen wie „Shared Source“ von Mi-

[#] Die Arbeit wurde für ein Seminar im Rahmen des Eulisp-Studienganges – www.eulisp.de – erstellt.

¹ So hat z.B. Spindler, auch wenn er dies nicht ausdrücklich erwähnt, wohl Linux vor Augen, wenn er verschiedene Schlüsse zieht. Gleiches gilt für Sester, CR 2000, 797, der diesen Hintergrund aber offenlegt.

² Free Software – S. 40.

³ Ohne den Quellcode, zumindest im Ansatz, verstanden zu haben, ist eine Bearbeitung kaum denkbar. Damit ist dem Entwickler aber ein Einblick in die Arbeitsweise des Programms möglich, womit er auch die Risiken erkennen kann.

⁴ Hierbei handelt es sich um die sogenannten vier Säulen der Open Source Soft-

Microsoft zu ermöglichen⁵, wird streckenweise auch von „Free Software“ gesprochen, um den insgesamt unentgeltlichen Charakter der Software zu betonen⁶.

Der Großteil dieser Software wird unter der GNU-GPL-Lizenz⁷ vertrieben⁸. Daneben besteht aber eine Vielzahl anderer Lizenzen, wie z.B. die Apache Software License⁹. Auch Software, die „nur“ mit Quellcode veröffentlicht wird, ohne eine bestimmte Lizenz zu wählen, ist anzutreffen¹⁰.

Paradigmatisch soll im Folgenden allgemein von der GPL ausgegangen werden, sind sich die verschiedenen Lizenzen doch zumindest bei der Frage, wie mit Haftungsproblemen umgegangen werden soll, ähnlich¹¹.

3 Rechtliche und tatsächliche Vorfragen

3.1 Was wird übertragen?

Bei der Übertragung von – proprietärer, also kostenpflichtiger – Software wird regelmäßig zwischen dem schuldrechtlichen Verpflichtungsgeschäft und der Einräumung von (urheberrechtlichen) Nutzungsrechten unterschieden¹². Dieser Ansatz wird auch bei Open

ware: Die Software darf kopiert, weiterverbreitet, verändert und mit diesen Änderungen wiederum verbreitet werden.

5 Vgl. z.B. das Windows CE Shared Source Premium Program (CEP), Meldung unter

<http://www.microsoft.com/germany/ms/presseservice/meldungen.asp?id=530868> bzw. <http://www.microsoft.com/resources/sharedsource/default.mspx>.

6 Betont wird hierbei insbesondere die Bedeutung der „Free Speech“, vgl. <http://www.gnu.org/philosophy/free-sw.html>.

7 <http://www.gnu.org/copyleft/copying-1.0.html>.

8 Im Folgenden nur § X GPL.

9 <http://www.apache.org/LICENSE-1.0> bzw. <http://www.apache.org/LICENSE-1.1>

10 Unter <http://sourceforge.net> finden sich Projekte im Quellcode, die wechselnden Lizenzen, wenn überhaupt, unterworfen sind. Meist ist eine „AS IS“-Klausel vorhanden. Inwieweit dabei jeweils von Open Source iSd. „offiziellen“ Definition gesprochen werden kann, soll hier offen bleiben.

11 In den meisten Lizenzen findet sich eine „AS IS“ Klausel, die eine Haftung ausschließen soll: Die Software wird nur so zur Verfügung gestellt, „wie sie ist“.

Vgl. z.B. die Apache Software License, <http://www.apache.org/LICENSE-1.1>, a.E., The FreeBSD Copyright, <http://www.de.freebsd.org/copyright/freebsd-license.html>, a.E., oder die Mozilla Public License version 1.1, <http://www.mozilla.org/MPL/MPL-1.1.html>, §7. Eine geordnete Übersicht über diverse Lizenzen findet sich auf http://www.ifross.de/ifross_html/lizenzcenter.html.

Vgl. auch die Aufzählung bei Koch CR 2000, 273, 274.

12 Vgl. z.B. Spindler, S. 72.

Source Software verwendet¹³.

3.2 Schuldrechtliche Einordnung

Open Source Software erhält der User kostenlos. Auch wenn u.U. für das Kopieren ein Entgelt verlangt werden kann, vgl. § 1 GPL, ändert dies nicht an dem Umstand, dass die Software an sich bzw. die Nutzungsrechte kostenfrei übertragen werden (müssen). Damit aber scheiden entgeltliche Verträge wie Auftrag oder Kaufvertrag aus.

3.2.1 Schenkung

Allgemein wird auf das Schenkungsrecht abgestellt¹⁴. Unerheblich dabei ist, ob es sich bei Software um eine Sache iSd. § 90 BGB handelt¹⁵ oder nicht, können doch auch Rechte verschenkt werden¹⁶.

3.2.1.1 Entreicherung

Problematisiert wird bei der Einordnung als Schenkung jedoch häufig die *Entreicherung* des Schenkers¹⁷. Software an sich ist praktisch kostenlos kopierbar¹⁸, womit eine Entreicherung bestritten werden könnte¹⁹.

Mit der Software wird jedoch ein (einfaches) Nutzungsrecht übertragen. Jede Übertragung unkörperlicher Rechte geht ohne „tatsächliche“ Übertragung von statten, stellt für das BGB im Kaufrecht aber kein Problem dar, vgl. § 453 I BGB. Ist der Erhalt von Sachsubstanz für den Kauf unerheblich, so muss dies folglich auch für die Entreicherung bei der Schenkung gelten²⁰. Mithin ist eine Entrei-

13 Anders Heussen – Free Software, der nur auf das Verpflichtungsgeschäft abstellt, hierzu später mehr.

14 Vgl. z.B. Jaeger/Metzer S. 136 ff.; Spindler, S. 73; jeweils m.w.N.

15 In der vielzitierten Entscheidung zur Sachqualität von (Standard-)Software, BGH NJW 1993, spricht der BGH Software keine Sachqualität zu, er spricht lediglich davon, dass diese „zumindest entsprechend“ den für Sachen geltenden Regeln behandelt werden kann, vgl. Kilian/Heussen-Monitz § 31 Rn 16 f.

16 Palandt-Weidenkaff, § 516 BGB, Rn. 5.

17 Vgl. z.B. Jaeger/Metzger S. 141 oder Spindler, S. 73.

18 Wird die (eigene) Open Source Software auf einem eigenen Server im Internet zum Download bereitgestellt, ändert sich diese Ausgangslage. Es entstehen Traffic-Kosten. Dieser Umstand wird aber – soweit ersichtlich – nicht in Betracht gezogen.

19 Vgl. Sester, CR 2000, 797, 799 f.; Koch, CR 2000, 333, 334.

20 Vgl. hierzu Spindler, S. 74; Jaeger/Metzger 141 f.

cherung bei einer Übertragung von Open Source Software gegeben.

3.2.1.2 Unentgeltlichkeit

Ein weiteres Problem stellt die *Unentgeltlichkeit* dar. In § 2 GPL werden diverse Anforderungen gestellt, für den Fall, dass veränderte Software weiterverbreitet werden soll. Man könnte in diesen Verpflichtungen eine Gegenleistung für die Übertragung der Software sehen²¹.

Beschreibt § 2 GPL lediglich den Umfang des übertragenen Verwertungsrechts, kann keine Gegenleistung angenommen werden. Letzteres kann jedoch nur dann überzeugen, wenn eine entsprechende Abspaltung verschiedener Nutzungsrechte überhaupt zulässig ist.

Ob Nutzungsrechte einzeln abgespalten werden können, bestimmt sich danach, inwieweit die Verkehrsauffassung in dem abgespaltenen Recht eine klar abgrenzbare, technisch-wirtschaftlich eigenständige Nutzungsart sieht²². Open Source Software unterscheidet sich von „normaler“ proprietärer Software durch die Mitlieferung des Quellcodes, von Shared-Source-Modellen durch die anschließende freie Verbreitbarkeit, ist folglich technisch abgrenzbar. Auch besteht für Open Source Software ein eigener, ständig wachsender Markt²³, der nicht ausschließlich auf Linux beschränkt ist²⁴. Die GPL führt in § 3 mithin zur Abspaltung eines Teilmarktes, für den ein neues Leistungsbündel geschnürt wird, das sich von anderen existierenden Vertriebsmodellen klar abgrenzt²⁵.

Daneben besteht die Möglichkeit des ursprünglichen Entwicklers weiter, die Software auch proprietär zu vertreiben.

Open Source Software stellt daher eine eigenständige Nutzungsart dar²⁶.

21 So. Sester CR 2000, 797, 800; Koch CR 2000, 333, 334 f.

22 Vgl. BGH NJW 2000, 3571 – OEM-Lizenz; Koch CR 2000, 333.

23 Vgl. z.B. den Bericht im SZ Magazin vom 21.11.03, S. 15 ff., Krieg der Systeme, in dem die Umstellung Münchens auf Linux dargestellt wird.

24 Herausragend dürfte das OpenOffice- und das Apache-Projekt sein.

25 Vgl. Heussen – Free Software, S. 26 f.

26 So auch Koch CR 2000, 333, 333 f., 336, der diesen Schluss aber nur für Software mit Linux- und bzw. Apachebezug ziehen will.

Mithin handelt es sich um keine Schenkung unter Auflagen²⁷ sondern um die Schenkung eines klar abgegrenzten Verwertungsrechts. Weiterhin wird vorgebracht, dass in dem Plan, im Anschluss an die Übertragung der Software durch Support, Pflege etc. Geld zu verdienen, die Entgeltlichkeit zu sehen sei²⁸. Hierbei handelt es sich aber um ein Fernziel, dessen Erreichen noch völlig unsicher ist, und das nichts daran ändert, dass die Software erst einmal kostenlos übertragen wird²⁹. Außerdem ist dieses Fernziel in dieser Allgemeinheit nicht auf alle Entwickler zu übertragbar. Insbesondere bei Programmierern, die in ihrer Freizeit mitentwickeln, greift es nicht.

3.2.2 Gesellschaft

Von manchen Autoren wird aus dem soeben geschilderten Plan, weitere Dienstleistungen zu ermöglichen bzw. zu erbringen, und aus dem Ziel, eine kostenlose Ausweichmöglichkeit zu bestehenden, kostenpflichtigen Angeboten zu schaffen, auf eine Art Gesellschaft geschlossen, die Schenkung also insgesamt abgelehnt³⁰.

Inwieweit die Motive der Entwickler, lediglich eine Möglichkeit zum späteren Absatz eigener Programme bzw. Leistungen zu schaffen, reduziert werden können, erscheint sehr fraglich³¹. Dies insbesondere dann, wenn man das breite Spektrum der als Open Source angebotenen Software im Auge behält und nicht lediglich auf Linux abstellt. Verlangt wird des weiteren ein über den gemeinsamen Gegenstand – die Open Source Software – hinausgehender Zweck³². Selbst wenn man aber ein so ausgestaltetes übergeordnetes, allgemeines Interesse annehmen wollte, ist für die Annahme

27 Eine andere Lösungsmöglichkeit wird von Jaeger/Metzger S. 143 vorgeschlagen: Die Verpflichtungen aus § 2 GPL greift erst dann ein, wenn die Software weiterverbreitet werden soll, den Endnutzer betreffen sie also nicht (Vgl. Spindler S. 74 f. mit Verweis auf Jaeger/Metzger S. 143; Deike CR 2003,9, 15.). Der Weg über eine „nachhängende“ Verpflichtung erscheint aber als zu gekünstelt.

28 Vgl. Sester CR 2000, 797, 799; Spindler S. 75 m.w.N.

29 Im Ergebnis gleich Spindler S. 75.

30 Sester CR 2000, 797, 801, der aber wegen „gravierender Unterschiede“ „bewusst die Aussage [vermeidet], es liege eine BGB-Gesellschaft vor“.

31 So auch Jaeger/Metzger S. 144 f.

32 So Palandt-Sprau § 705 Rn. 3.

einer Gesellschaft ein weiterer Verbindlichkeitsgrad nötig³³. Ein solcher Rechtsbindungswille ist jedoch allein in dem Ziel, weitere Alternativen am Softwaremarkt zu eröffnen, nicht zu sehen³⁴.

Nötig wäre außerdem eine Pflicht, den Gesellschaftszweck zu fördern, § 705 BGB. Aus der GPL ergibt sich jedoch gerade keine Pflicht, an der Software weiterzuentwickeln, vgl. § 2 I GPL³⁵.

Eine Gesellschaft ist nicht gegeben.

3.2.3 Verfügung

Des weiteren wird vertreten, dass es überhaupt an einem schuldrechtlichen Vertrag fehle³⁶ bzw. dieser nicht nötig sei. Durch eine solche Sichtweise wird das Problem der rechtlichen Einordnung von Open Source Software jedoch nur von der vertraglichen Ebene auf die deliktische Ebene verschoben³⁷. Daher wird diesem Lösungsweg hier nicht gefolgt.

3.2.4 Ergebnis

Als Ergebnis kann festgehalten werden, dass die Übertragung von Open Source Software als Schenkung zu klassifizieren ist³⁸.

3.3 Zusammensetzung der Entwickler

Die Entwicklerschar setzt sich prinzipiell aus zwei Gruppen zusammen.

Auf der einen Seite stehen Firmen wie Sun oder IBM, die Open Source Software von ihren Angestellten entwickeln lassen, auf der anderen Seite Programmierer, die die Entwicklung in ihrer Freizeit vorantreiben. Letztere sollen im Rahmen dieses Textes in Ab-

33 Vgl. allgemein Palandt-Sprau § 705 Rn. 3, 9; Spindler S. 75.

34 Dies bedeutet nicht, dass nicht im Rahmen des § 8 UrhG eine Gesellschaft entstehen könnte, hierzu später mehr. Hiervon betroffen ist aber nicht das System der Open Source Software an sich, sondern lediglich eine bestimmte Konstellation bei der Erstellung der Software.

35 Vgl. Jaeger/Metzger S. 39 f.

36 Heussen – Free Software S. 36 ff.

37 Vgl. hierzu die Ergebnisse später unter 4.3.5.

38 Will man den oben genannten Schwierigkeiten aus dem Weg gehen indem man einen Vertrag sui generis annimmt, vgl. z.B. F.A. Koch, CR 2000, 333, 335, muss dennoch geklärt werden, welchem gesetzlich geregelten Vertrag dieser neue Vertragstyp ähnelt. Damit ist man aber wiederum bei obigen Problemen angelangt, es wurde nichts gewonnen.

grenzung zu professionellen Entwicklern als Freizeitprogrammierer bezeichnet werden.

3.3.1 Beziehungen der Entwickler untereinander

3.3.1.1 Basarmethode

Hintergrund der Open Source Entwicklung ist der Gedanke, dass jeder sich an der Entwicklung beteiligen können soll. Die Software wird, hat sie einen entsprechenden, vom ursprünglichen Entwickler frei festlegbaren, Stand erreicht, zur Diskussion gestellt. Die Entwicklung folgt also keinem vorgefertigten Plan („Kathedralen-Methode“), sondern ist dezentral angelegt („Basarmethode“)³⁹. Damit aber ist eine klare Festlegung der Beziehungen unter den Entwicklern zumindest in der Praxis schwierig.

3.3.1.2 Urheberrechtliche Sicht

Grundsätzlich beantwortet sich die Frage, wie verschiedene Entwickler, die an einer Open Source Software beteiligt waren, zueinander stehen, aus dem UrhG.

Urheber ist der Schöpfer des Werkes, § 7 UrhG. Hierbei sind, neben der Alleinurheberschaft,

- die Miturheberschaft, § 8 UrhG,
- ein verbundenes Werk, § 9 UrhG, oder
- eine Bearbeitung, § 3 UrhG,

denkbar.

3.3.1.2.1 Miturheberschaft

Miturheberschaft ist gegeben, wenn mehrere Personen dergestalt an einem Werk iSd. § 2 UrhG mitwirken, so dass sich die einzelnen Teile nicht mehr extrahieren lassen⁴⁰, § 8 I UrhG. In Abgrenzung zum Sammelwerk entsteht also ein neues Werk, nur das Ganze ist Verkehrs- und Rechtsgegenstand⁴¹. Wichtig ist, dass jeder Beitrag jeweils Werkcharakter hat, um eine Abgrenzung zur bloßen Anregung

³⁹ Vgl. Jaeger/Metzger S. 13 mit Verweisen auf den Ursprung dieser Bezeichnung.

⁴⁰ Vgl. Koch CR 2000, 273, 277; allgemein hierzu Reh binder Rn. 167 ff.

⁴¹ Vgl. Reh binder Rn. 168.

oder Gehilfenschaft zu ermöglichen⁴².

Denkbar erscheint die Miturheberschaft z.B. bei der Entwicklung eines OS wie Linux. Hier sind die einzelnen Teile des Systems nicht allein heraustrennbar. Ähnlich könnte sich die Situation bei OpenOffice darstellen, allerdings sind hierbei die verschiedenen Applikationen wie der Writer oder Impress jeweils einzeln verwertbar, so dass Miturheberschaft nur innerhalb eines dieser Programm denkbar ist.

3.3.1.2.2 Verbundene Werke

Dagegen werden bei *verbundenen Werken* mehrere selbstständige Werke miteinander verbunden, § 9 UrhG. Die ursprünglichen Urheberrechte bleiben bestehen. Die Zusammenführung der Werke darf auch kein eigenes Werk darstellen, wäre sonst doch ein Sammelwerk iSd. § 4 UrhG gegeben⁴³. Beispiel ist die OpenOffice-Suite.

3.3.1.2.3 Bearbeitung

Durch eine *Bearbeitung* entsteht ein eigenes Urheberrecht, § 3 UrhG. Hieran ist zu denken, wenn nachträglich ein Programm erweitert wird, wie es bei Open Source Software ja gerade geschehen soll⁴⁴. Der Entwickler steht insoweit „alleine“. Er ist sonst keiner Person verbunden, wenn man den/die Urheber des bearbeiteten Werkes beiseite lässt.

3.3.1.2.4 Unterscheidungsprobleme

Die Unterscheidung zwischen einem verbundenen Werk und der Miturheberschaft fällt häufig schwer. Im hier behandelten Rahmen ist dies jedoch weniger wichtig, ist das Ergebnis doch jeweils eine Gesellschaft iSd. §§ 705 ff⁴⁵. Damit läuft das Haftungsregime gleich. Zwar ist – wegen der eigenständigen Werke – bei einem verbundenen Werk die Trennung möglich, bei einem in Miturheber-

42 Vgl. Rehbinder Rn. 169 f.

43 Vgl. Koch CR 2000, 273, 277 f.

44 Vgl. Koch CR 2000, 273, 278;

45 Vgl. § 8 II UrhG und allgemein Rehbinder Rn. 174.

schaft geschaffenen Werk nicht, doch stellt sich die entsprechende Frage erst, wenn das Kind in den Brunnen gefallen, der Haftungsfall also aufgetreten ist.

3.3.1.2.5 Urheber im Arbeitsverhältnis

Erstellt der Urheber die Software im Arbeitsverhältnis, so bleibt er zwar Urheber, die Verwertungsrechte liegen aber bei dem Arbeitgeber, vgl. § 69b UrhG. An späterer Stelle wird zu klären sein, wie sich dies auf verschiedene Haftungstatbestände auswirkt.

3.3.1.3 BGB-Gesellschaft

Losgelöst von der Frage, wie die Entwickler urheberrechtlich zueinander stehen, wird auch allgemein eine BGB-Gesellschaft zwischen den Entwicklern angenommen. Weshalb diesem, insbesondere von Sester⁴⁶ vertretenen, Ansatz nicht gefolgt werden kann, wurde bereits dargelegt⁴⁷.

3.3.1.4 Ergebnis

Die vorgestellten Urhebermodelle können beliebig vermischt werden. So kann z.B. eine Bearbeitung durch mehrere Miturheber geschehen.

Festgehalten werden kann, dass im Rahmen der Basarmethode keine allgemeine BGB-Gesellschaft entsteht, es muss vielmehr im Einzelfall genau untersucht werden, ob der einzelne Mitentwickler in einer Gesellschaft aufgeht, wer Mitglied dieser Gesellschaft ist und ob diese möglichen Gesellschaft ein Haftungstatbestand erfüllt.

4 Haftungstatbestände

4.1 Vertragliche Haftung

Ist die Übertragung von Open Source Software als Schenkung zu sehen, so bestimmt sich die vertragliche Haftung nach §§ 521, 523 f. BGB.

Gehaftet wird folglich nur für Vorsatz und grobe Fahrlässigkeit.

⁴⁶ Sester CR 2000, 797, 799 f.

⁴⁷ 3.2.2.

4.1.1 Haftungsausschluss

4.1.1.1 Einleitung

Die §§ 521 ff. BGB könnten letztlich unerheblich sein, wenn ein Haftungsausschluss aus den §§ 11 f. GPL⁴⁸ greift.

Bei der GPL handelt es sich um AGB, ist sie doch für eine Vielzahl von Fällen vorformuliert und wird sie der anderen Seite, dem Nutzer, gestellt, vgl. § 305 I 1 BGB.

Aus verschiedenen Gründen ist schon der Einbezug der GPL problematisch⁴⁹.

Dies kann aber zumindest in diesem Rahmen unbeachtet bleiben, wenn die GPL, eine wirksame Einbeziehung vorausgesetzt, keinen wirksamen Haftungsausschluss erwirken kann.

4.1.1.2 Verbraucher

Die GPL schließt in den §§ 11 f. jegliche Haftung aus.

Wird die GPL gegenüber einem Verbraucher gestellt, so ist dies schon aus § 309 Nr. 7 a.)⁵⁰, b.) iVm. § 310 I 1 BGB unzulässig, wird doch keine Ausnahme für die dort genannten Rechtsgüter Leben, Körper und Gesundheit – a.) – bzw. das grobe Verschulden – b.) – gemacht.

4.1.1.3 Unternehmer

Auch wenn § 309 BGB gegenüber Unternehmern nicht unmittelbar gilt, § 310 I 1 BGB, ist über § 307 iVm. § 310 I 2 HS 1 BGB ein völliger Haftungsausschluss, der insbesondere auch die vorsätzliche Schädigung mit umfasst, vgl. hierzu § 276 III BGB, nicht zulässig⁵¹.

4.1.1.4 Zwischenergebnis

Folge einer unzulässigen Klausel ist deren Unwirksamkeit. Daneben

48 Vgl. auch die aufgeführten ähnliche Vorschriften in Fussnote 11.

49 Vgl. allgemein die ShrinkWrap/ClickWrap-Problematik Schneider J Rn. 4 ff.; unproblematisch scheint dies Koch CR 2000, 333, 339 zu sehen.

50 Eine hier nicht näher auszuführende Frage ist die, ob eine Unzulässigkeit einer Klausel, die auf das Leben etc. keine ausdrückliche Rücksicht nimmt, auch dann wegen § 309 BGB gegeben ist, wenn eine entsprechende Verletzung praktisch nicht denkbar ist, wie es z.B. bei einer Textverarbeitungssoftware wie OpenOffice oder dem Browser Mozilla der Fall ist.

51 Palandt-Heinrichs § 309 Rn. 48; BGHZ 20, 164; 70, 364.

ist auch die Abbedingung der gegebenenfalls einschlägigen Produkthaftung unzulässig, § 14 I 2 ProdHaftG.

Damit ist es für die Frage der Haftung unerheblich, ob die GPL wirksam einbezogen wurde, oder nicht, ein – gewollter – völliger Haftungsausschluss ist unwirksam⁵².

4.1.2 Ergebnis

Damit aber bleibt es bei der Haftungsbegründung aus Schenkungsrecht.

Der Entwickler von Open Source Software haftet für Vorsatz und grobe Fahrlässigkeit.

4.1.3 Angestellte Urheber

Ist Vertragspartner ein Einzelner, sind die vertraglichen Haftungsfragen hiermit geklärt. Denkbar ist aber auch die Entwicklung von Open Source Software im Arbeitsverhältnis. Bei proprietärer Software ist Vertragspartner der Arbeitgeber, stehen ihm doch die Verwertungsrechte zu, § 69b UrhG. Die GPL konstruiert die Rechteübertragung aber dergestalt, dass jeweils mit dem Urheber kontrahiert wird, § 6 GPL. Damit scheint der Angestellte in der Haftung zu stehen. Zwar wird ihm im Haftungsfall regelmäßig ein Freistellungsanspruch gegenüber seinem Arbeitgeber zustehen, dies nützt im Insolvenzfall jedoch wenig⁵³.

Es stellt sich jedoch die Frage, ob insoweit der GPL wortgetreu gefolgt werden muss. Trotz des Wunsches, an keiner einzelnen Rechtsordnung festgehalten zu werden (vgl. § 12 GPL), wurde die GPL vor dem Hintergrund des amerikanischen Rechts erstellt. Dort aber erwirbt der Arbeitgeber das Urheberrecht komplett⁵⁴, so dass sich das hier aufgeworfene Problem nicht stellt.

52 Vor dem Hintergrund des alten Rechts kommt Koch CR 2000, 333, 340 zu einem meist anderen Ergebnis. Er sieht die (nunmehr nur) für § 309 Nr. 8 b.) BGB n.F. nötige „neu hergestellte Sache“ regelmäßig als nicht gegeben an. Im neuen Recht ist dies wegen § 309 Nr. 7 BGB i.E. nicht mehr vertretbar.

53 An dieser Stelle zeigt sich ein Vorteil der von Heussen – Free Software, S. 36 vorgeschlagenen Verfügungslösung. Mangels Vertrags kann es zu obigem Problem nicht kommen.

54 „work made for hire“, vgl. Jaeger/Metzger S. 25 f.

Der Gedanke des § 69b I 2 UrhG ist es, dem Arbeitgeber die Verwertungsrechte an der Software zuzuweisen. Dies ändert sich auch dann nicht, wenn Open Source Software erstellt werden soll. Als „Lizenzgeber“ im Sinne der GPL ist daher der Arbeitgeber zu sehen⁵⁵. Mit diesem, nicht mit dem einzelnen Programmierer wird ein Vertrag im haftungsrechtlichen Sinn geschlossen. Auch bei angestellten Programmieren ändert sich die Rechtslage daher nicht. Bei anderen Lizenzen, die eine dem § 6 GPL entsprechende Klausel nicht enthalten, stellt sich obiges Problem nicht⁵⁶.

4.2 Verschuldensunabhängige Haftung gemäß dem ProdHaftG

Des weiteren könnte das ProdHaftG im Haftungsfall einen Anspruch gewähren.

Voraussetzung hierzu ist gemäß § 1 ProdHaftG dass

- ein *Produkt*, § 2 ProdHaftG,
- das mit einem *Fehler*, § 3 ProdHaftG, behaftet ist,
- durch einen *Hersteller*, § 4 ProdHaftG, in den Umlauf gebracht wird und
- einen *Schaden* an bestimmten Rechtsgütern verursacht.
- Außerdem darf die Haftung nicht ausgeschlossen sein.

4.2.1 Produkt

Ein Anspruch könnte schon am ersten Merkmal, dem Produkt, scheitern. § 2 ProdHaftG spricht von einer beweglichen Sache. Diese muss nicht zwingend iSv. § 90 BGB verstanden werden, beruht das ProdHaftG und damit die Formulierung doch auf einer europarechtlichen Grundlage⁵⁷.

4.2.1.1 Software auf Datenträger

Die h.M. geht zumindest bei (Standard-) Software, die auf einem Da-

⁵⁵ So auch Koch CR 2000, 333, 341.

⁵⁶ Vgl. z.B. die Apache-Lizenz.

⁵⁷ Richtlinie des Rates 85/374/EWG vom 25. Juli 1985 über die Angleichung der Recht- und Verwaltungsvorschriften der Mitgliedstaaten über die Haftung für fehlerhafte Produkte – EG-Produkthaftungsrichtlinie (Abl. EG Nr. L 210/29), angepasst durch Art. 23 Buchstabe c in Verbindung mit Anhang III des Abkommens über die EWR (Abl. EG Nr. L 1/1).

tenträger angeboten wird, von einer Sache aus⁵⁸. Hintergrund sind die Entscheidungen des BGH⁵⁹ zur – „zumindest entsprechenden“ – Anwendbarkeit des Sachkaufs auf Software⁶⁰. Doch wird die Sacheigenschaft von Software bestritten⁶¹. Um zu entscheiden muss letztlich gefragt werden, vor welchen Gefahren das ProdHaftG schützen soll. Dies sind Gefahren, die dadurch entstehen, dass der Nutzer ein Produkt „in Händen hält“, den Produzenten mit seinem Produkt also in seinen Privatbereich gelangen lässt, und somit erhöhtes Gefährdungspotential entsteht. Dieses Gefährdungspotential ist bei (Standard-) Software gegeben.

Zumindest bei Standardsoftware, die auf Datenträgern angeboten wird, ist daher von der Sacheigenschaft iSd. § 2 ProdHaftG auszugehen⁶².

4.2.1.2 Download

Anders könnte bei heruntergeladener Software zu entscheiden sein⁶³. Hier erhält der Nutzer keinen Gegenstand in die Hand. Allerdings macht es heute für den Nutzer keinen Unterschied mehr, wie Software zugänglich gemacht wird⁶⁴. Dies ändert aber nichts an dem Gefahrenpotential auf Seiten des Nutzers. Somit ist auch Software, die vom Nutzer heruntergeladen wurde, unter die Produkteigenschaft iSd. § 2 ProdHaftG zu fassen⁶⁵.

4.2.1.3 Application Service Providing (ASP)

Problematisch erscheint die Produkteigenschaft aber bei lediglich

58 Staudinger-Oechsler § 2 Rn. 64 m.w.N.

59 BGH NJW 1993, 2436; NJW 1988, 406.

60 Hieraus folgt aber nicht, wie häufig zu lesen ist, dass der BGH (Standard-) Software als Sache betrachtet, vgl. Kilian/Heussen-Monitz § 31 Rn 16 f.

61 Kilian/Heussen-Monitz § 31 Rn. 16 ff., 234 ff., Honsell Jus 1995, 211, 212.

62 So im Ergebnis auch Kilian/Heussen-Littbarski § 180 Rn. 120.

63 Allgemein zum Problem der Verkörperung geistiger Leistung Cahn NJW 2899 ff.

64 So stellt z.B. Oracle das gesamte Angebot nur noch online zur Verfügung, es erfolgt sodann eine Freischaltung. Auch Borland stellt einen Teil der Produktpalette zum Download mit anschließender Übertragung des Lizenzschlüssels zur Verfügung.

65 So auch MüKo-Cahn § 2 ProdHaftG Rn. 7; Spindler S. 91 f.; im Ergebnis gleich Cahn NJW 2899 ff. Ablehnend Honsell Jus 1995, 211 ff. und Staudinger-Oechsler Rn. 69a;

vgl. Verweise der h.M. bei Staudinger-Oechsler § 2 Rn. 65, dort ist auch eine nähere Darstellung des Streitstands zu finden.

Online übertragener und genutzter Software⁶⁶. Auf diese hat der Nutzer, wurde die Verbindung getrennt und der Computer ausgeschaltet, keinen Zugriff mehr. Zwar bedarf jede Nutzung von Software einer irgendwie gearteten Verkörperung⁶⁷. Dies muss aber nicht heißen, dass dies für die Annahme einer Produkteigenschaft genügt⁶⁸.

Rein Online übertragene Software ist mit Information vergleichbar, die immer irgendwie verkörpert sein muss, damit sie dem Menschen überhaupt verständlich wird⁶⁹, zur Sache wird sie dadurch nicht⁷⁰.

Ist die reine Information im und mit ihr vergleichbare ASP-Angebote folglich keine Sache iSd. § 90 BGB, so muss noch geklärt werden, ob § 2 ProdHaftG ein gegenüber § 90 BGB erweiterter Sachbegriff zugrundeliegt. Vorgetragen wird, dass Software unter die in § 2 ProdHaftG genannte Elektrizität zu fassen sei, bedeute die Übertragung mittels elektromagnetische Ströme doch eine Änderung der die Software übertragenden elektrischen Struktur⁷¹. Elektrizität wird in § 2 ProdHaftG jedoch als (einzige) Ausnahme aufgeführt, der Gang über die Analogie ist folglich verwehrt⁷². Weiterhin fehlt es bei der so übertragenen Software an einem gesteigerten Vertrauen des Nutzers auf die Verfügbarkeit⁷³, ein Unterschied zu dem Fall, in dem er die Software als CD „in den Händen hält“.

4.2.1.4 Zwischenergebnis

Ob Software Produkt iSd. ProdHaftG ist, hängt somit vom Einzelfall ab, ist aber meist zu bejahen.

4.2.2 Fehler

Weiteres Tatbestandsmerkmal ist der Fehler, dessen Definition sich

66 Offen bleiben soll an dieser Stelle, inwieweit ASP von der GPL gedeckt ist.

Hierzu im Hinblick auf eine neue Nutzungsart u.a. Jaeger/Metzger S. 34.

67 So Taeger CR 1996, 257, 261; vgl. auch Cahn NJW 1996, 2899, 2904.

68 Vgl. Cahn NJW 2899, 2904.

69 So Staudinger-Oechsler § 2 Rn. 66 m.w.N.

70 So auch MüKo-Cahn § 2 ProdHaftG Rn. 7 m.w.N.

71 Vgl. die Verweise bei Spindler S. 92 Fn. 573.

72 So auch Cahn NJW 1996, 2899, 2900; ebenfalls ablehnend mit Verweis auf die h.H. Staudinger-Oechsler § 2 Rn. 67.

73 MüKo-Cahn § 2 ProdHaftG Rn.6, ders. NJW 1996, 2899, 2903 f.

aus § 3 ProdHaftG ergibt.

Allgemein ist ein Produkt fehlerfrei, wenn es den berechtigten Sicherheitserwartungen der Allgemeinheit unter Beachtung des jeweiligen Standes von Wissenschaft und Technik entspricht⁷⁴. Anders als das Sachmängelrecht schützt das ProdHaftG das Integritätsinteresse jedes Benutzers und Dritten, also nicht nur des Käufers⁷⁵.

Folgende Fehlerkategorien werden aufgeführt⁷⁶:

- Konstruktionsfehler
- Fabrikationsfehler
- Instruktionsfehler
- Verletzung der Produktbeobachtungspflicht.

Konstruktionsfehler können bei der fehlerhaften Implementierung/Programmierung auftreten. Da das Kopieren von Software kein Problem und damit praktisch fehlerlos möglich ist, dürften Haftungsfälle in Folge von Fabrikationsfehler unwahrscheinlich sein, muss doch der Schaden an einer anderen Sache entstehen, vgl. § 1 I 2 ProdHaftG. Daneben sind aber sowohl Fehler auf der Instruktionsebene als auch bei der Produktbeobachtung denkbar.

Software als komplexes System kann ab einer bestimmten Stufe nicht mehr fehlerfrei hergestellt werden⁷⁷. Folge hieraus könnte sein, dass fehlerfreie Software überhaupt nicht erwartet werden kann⁷⁸. Dieser Schluss erscheint aber angesichts eines erheblichen Gefährdungspotentials von Software als zu weit gehend⁷⁹. Auf der anderen Seite lässt aber auch die Subsumtion jedes Fehlers unter § 3 ProdHaftG die jeweiligen Umstände – hier die unmögliche Fehlerfreiheit – außer acht. Es erscheint daher angebracht, von Software zumindest eine Grundfunktionalität zu fordern, während periphere Fehler nicht unter § 3 ProdHaftG zu fassen sind⁸⁰.

74 Vgl. Kilian/Heussen-Littbarski § 180 Rn. 24.

75 Palandt-Thomas, § 3 ProdHaftG Rn. 1.

76 Vgl. z.B. Palandt-Thomas, § 3 ProdHaftG Rn. 2 ff.

77 Vgl. Heussen ??, Staudinger-Oechsler § 3 ProdHaftG Rn. 90, jeweils m.w.N.

78 Honsell JuS 1995, 211, 212.

79 So auch Taeger CR 1996, 257.

80 So auch Staudinger-Oechsler § 3 ProdHaftG Rn. 90.

Daneben kommt es bei neuer Software häufig zu Problemen, was dem Verkehr auch bekannt ist, womit auch hierauf Rücksicht zu nehmen ist⁸¹ und die Sicherheitserwartungen entsprechend anzupassen sind.

4.2.3 Hersteller

Nunmehr stellt sich die Frage, wer Hersteller der Software ist. Nötig ist hierzu, dass eine Tätigkeit für eigene Rechnung erfolgt⁸². Hersteller iSd. § 4 I ProdHaftG ist folglich der Unternehmer, nicht der einzelne Mitarbeiter⁸³. Unerheblich sind daher die Probleme, die sich durch § 69b UrhG iVm. § 2 b GPL ergeben können⁸⁴, Hersteller bleibt der Unternehmer.

Ein zusätzliches Problem könnte sich aber dadurch ergeben, dass ein Dritter, ein so genannter Distributor, ein Open Source Programm auf einer CD oder einen anderen Datenträger verbreitet.

Weiter oben wurde jedoch ausgeführt, dass es keinen Unterschied macht, in welcher Art und Weise das Programm dargeboten wird⁸⁵. Ist dem aber so, dann kann es auch nicht entscheidend sein, wie das Programm letztlich zum Geschädigten kommt, zumindest dann, wenn der Autor mit der Art der Verbreitung einverstanden war, was im Open Source Bereich bzgl. CDs immer der Fall ist. Somit bleibt der Autor – die restlichen Voraussetzungen als gegeben vorausgesetzt – auch dann Hersteller, wenn ein Dritter das Programm auf CD etc. veröffentlicht⁸⁶. Dies gilt jedoch nur im Hinblick auf die hergestellte Software⁸⁷.

81 Vgl. Kilian/Heussen-Monitz § 31 Rn. 237.

82 Zu den weiteren, hier nicht problematischen Voraussetzungen – Schaffung einer neuen, beweglichen Sache und einem Handeln, das über das im Rahmen des Produktvertriebs nötige Einwirken auf die Sache hinausgeht – vgl. Staudinger-Oechsler § 4 ProdHaftG Rn. 8 ff.

83 Staudinger-Oechsler § 4 ProdHaftG Rn. 9.

84 Vgl. zur Problematik Spindler S. 53 f. bzw. die Ausführungen unter 4.1.3.

85 Auf einem Datenträger oder als Downloadmöglichkeit.

86 So im Zusammenhang von Autor und Verlag auch MüKo-Cahn § 2 ProdHaftG Rn. 6 f. bzw. ders. NJW 2899, 2904.

87 Daneben ergibt sich aus § 4 III ProdHaftG ein weiteres Problem: Wegen der Basarmethode und der Nutzung von Pseudonymen ist es gegebenenfalls nicht mehr möglich, den Hersteller eines bestimmten Softwareprodukts ausfindig zu machen. Dann aber ist u.U. auch der Lieferant Hersteller, also gegebenenfalls der Distributor oder sogar der den Download Ermöglichende.

4.2.4 Schaden und Ausschluss

Ist es sodann zu einem unter das ProdHaftG fallenden Schaden gekommen⁸⁸, erklärt § 1 II ProdHaftG dennoch die Ersatzpflicht in einigen Fällen für ausgeschlossen.

Als problematisch erscheint im Rahmen der Open Source Software die Nr. 3⁸⁹.

Open Source Software ist per Definition nicht für den Verkauf bestimmt.

Daneben darf sie aber auch nicht im Rahmen einer beruflichen Tätigkeit erstellt worden sein. Beruf ist jede nicht nur private, auf Dauer und Gewinnerzielung hin angelegte Tätigkeit⁹⁰. Die Grenzen werden hierbei allgemein sehr eng gezogen⁹¹.

Somit fällt Software, die ein Arbeitnehmer im Rahmen seiner Tätigkeit erstellt hat⁹², nicht unter Nr. 3⁹³, das ProdHaftG findet also Anwendung.

In Betracht kommt aber auch der Freizeitprogrammierer, der in seiner Freizeit Software erstellt oder weiterentwickelt. Inwieweit dieser

Kriterium könnte hierbei sein, inwieweit der Distributor eine eigene Zusammenstellung liefert.

Bei Distributoren wie Suse, die das Produkt mit einem eigenen Namen versehen, ist dies wegen § 4 I 2 ProdHaftG sowieso der Fall.

Geklärt werden muss daher, ob dies auch für den Bereich der Open Source Software gelten kann. Anders als bei bloßen Entwicklern, werden Distributoren zumeist gewerblich aktiv. Dennoch stellt § 1 GPL vor dem Hintergrund des Umlageprinzips ein Problem dar: Der Distributor darf für die Software selbst kein Entgelt verlangen, lediglich für das Kopieren oder sonstige Leistungen. Dann aber ist ihm über den Preis gerade keine wirksame Umlagerung des Produktrisikos möglich, liegt das Hauptrisiko doch in der Software, nicht in dem von ihm genutzten Datenträger.

Eine nähere Klärung dieser Problematik fällt aber nicht in den Fokus dieser Arbeit.

88 Körper, Gesundheit, andere Sache beschädigt, vgl. § 1 I ProdHaftG.

89 Für Nr. 2 könnte an einen Virus gedacht werden, Nr. 5 vermischt sich mit den zur Fehlerhaftigkeit der Software gemachten Aussagen. Nr. 1 und Nr. 4 erscheinen weniger relevant.

90 Staudinger-Oechsler § 1 ProdHaftG Rn. 92.

91 Vgl. die Beispiele bei Staudinger-Oechsler § 1 ProdHaftG Rn. 93.

92 Zur Abgrenzung Jaeger/Metzger S. 154; vgl. allgemein z.B. das Engagement von Sun bei OpenOffice oder die Mitwirkung von IBM bei Linux.

93 Die Befreiungsgründe der Nr. 3 müssen *kumulativ* vorliegen (so auch MüKo-Cahn § 1 ProdHaftG Rn. 36 u. Staudinger-Oechsler § 1 ProdHaftG Rn. 89). Es ist daher unerheblich, ob im Anschluss an die Verbreitung der Software Support o.ä. angeboten werden soll. Entscheidend ist die Herstellung im Rahmen der beruflichen Tätigkeit. Dies scheint Spindler S. 93 zu übersehen.

Kenntnisse verwendet, die er während seiner Berufstätigkeit erhalten hat, ist unerheblich⁹⁴. Nr. 3 ist in diesem Fall folglich erfüllt.

Spindler⁹⁵ will hier dennoch im Zweifel einen beruflich-wirtschaftlichen Zweck annehmen. Er stellt zum einen darauf ab, dass Programmierer von Open Source Software häufig ihre Reputation auf Arbeitsmärkten durch ihre Mitarbeit erhöhen wollen, zum anderen auf das Produkt Open Source Software⁹⁶ selbst. Dieses sei auf eine möglichst breite Verbreitung ausgelegt – komme einer industriellen Produktion⁹⁷ damit nahe – und unterscheide sich häufig bis auf die Unentgeltlichkeit kaum von proprietärer Software.

Diese Argumente vermögen nicht zu überzeugen. Die Reputation des Programmierers ist ein derart weit gefasster Begriff, dass hierunter praktisch alles gefasst werden kann. Die meisten Handlungen eines Menschen sind in irgendeiner Form auf Dritte ausgerichtet, so dass jede auf die gesellschaftliche Stellung abstellende Handlung demgemäß unter Nr. 3 fallen könnte.

Hintergrund der Nr. 3 ist u.a. das Umlageprinzip⁹⁸, also die Möglichkeit des Herstellers, über den Preis die Versicherungsprämie und damit das Risiko auf alle Kunden verteilen zu können. Diese Versicherbarkeit fehlt bei einem Freizeitprogrammierer völlig⁹⁹. Zwar ist auch dem hobbymäßigen Hersteller eine Versicherung nicht möglich¹⁰⁰, doch zielt dieser wenigstens auf die wirtschaftliche Verwertung ab. Anders beim hobbymäßigen Programmierer von

94 Vgl. MüKo-Cahn § 1 ProdHaftG Rn. 40.

95 S. 94.

96 Vor Augen scheint er, wie auch im Rest des Werks, Linux zu haben.

97 Vgl. hierzu die Motive, die zum Erlass des ProdHaftG bzw. der zugrundeliegenden Richtlinie führten, die inkl. der geschichtlichen Entwicklung bei Staudinger-Oechsler Einl zum ProdHaftG Rn. 5 ff. aufgezeigt werden.

98 Vgl. Staudinger-Oechsler, Einl zum ProdHaftG Rn. 16.

99 Eine Anfrage bei der Allianz zu Möglichkeiten der Versicherbarkeit wurden leider nicht beantwortet. Jedoch findet sich bei der Allianz kein Standardprodukt, das eine Produkthaftpflicht umfasst. Dies dürfte mit den unterschiedlichen Produkten und den damit verbundenen Risiken zusammenhängen. Damit aber muss jede dieser Versicherungen individuell ausgehandelt werden, was sich auch auf die Prämien auswirken dürfte (vgl. die Motivationen zur Einführung einer Höchstsumme, MüKo-Cahn § 10 ProdHaftG Rn 1 m.w.N.).

Ein Hinweis auf die nur sehr eingeschränkt angebotenen Versicherungen findet sich auch bei Taeger CR 1996, 257, 271.

100 vgl. das Beispiel bei Staudinger-Oechsler, § 1 ProdHaftG Rn. 93.

Open Source Software.

Dennoch regelmäßig einen beruflich-wirtschaftlichen Zweck anzunehmen, erscheint mithin verfehlt¹⁰¹.

4.2.5 Ergebnis

Als Ergebnis kann daher festgehalten werden, dass das ProdHaftG bei Open Source Software zwar grundsätzlich Anwendung finden kann, bei einem Programmierer, der nicht im Rahmen seiner beruflichen Tätigkeit mitwirkte, scheidet eine Ersatzpflicht jedoch aus. Anwendung findet das ProdHaftG somit nur auf Hersteller wie z.B. Sun¹⁰² oder IBM¹⁰³.

4.3 Deliktische Haftung

Durch Software herbeigeführte Schäden sind zumeist mittelbare Schäden. Ausgeklammert werden sollen im Folgenden vorsätzliche Schädigungen, die ohne weiteres zu einer Haftungsbegründung führen. Außerdem soll die Kausalität zwischen Software und Schaden als gegeben betrachtet werden.

Problematisch ist die Bestimmung einer fahrlässigen Verursachung.

4.3.1 Gesetzliche Beschränkungen der Haftung, § 521 BGB

Weiter oben wurde das Grundverhältnis einer Open Source Software-Übertragung als Schenkung erkannt. Bei einer solchen schränkt § 521 BGB den Haftungsmaßstab ein. Diese Einschränkung wäre praktisch wirkungslos, würde sie nicht auch im Deliktsrecht gelten, könnte § 521 BGB doch ohne Probleme umgangen werden¹⁰⁴. Damit aber gilt § 521 BGB auch im Rahmen des § 823 BGB¹⁰⁵.

Die Schenkung als schuldrechtlicher Vertrag kann jedoch nur Wirkungen zwischen den beteiligten Parteien entfalten. Wird, verursacht durch Software, ein Dritter verletzt, kann § 521 BGB nicht

¹⁰¹ So im Ergebnis auch Jaeger/Metzger S. 153.

¹⁰² OpenOffice.

¹⁰³ Linux.

¹⁰⁴ Paradigma ist die Entwicklung der weiterfressenden Mängel durch die Rechtsprechung.

¹⁰⁵ So auch Palandt-Thomas Einf. v. § 823 BGB Rn. 8; a.A. und zum Streitstand MüKo-Kollhosser § 521 Rn. 2 ff.

zur Bestimmung des Haftungsmaßstabes herangezogen werden.

4.3.2 Verkehrspflichten

4.3.2.1 Allgemeine Kriterien

Damit muss im Rahmen des § 823 BGB auf die Verkehrspflichten abgestellt werden. Funktion der Verkehrspflichten ist es, den Tatbestand der widerrechtlichen *fahrlässigen* Verletzung näher zu klären¹⁰⁶. Die Bestimmung dieser Pflichten stellt ein Hauptproblem im Bereich der (Open Source) Software dar.

Allgemein besagen Verkehrspflichten, dass Rücksicht auf die Gefährdung Dritter zu nehmen ist. Dabei muss nicht für alle denkbaren Schadensmöglichkeiten Vorsorge getroffen werden. Entscheidend sind vielmehr die Sicherheitserwartungen des betroffenen Verkehrs, die im Rahmen der wirtschaftlichen Zumutbarkeit zur Schadensvermeidung geeignet sind¹⁰⁷. Beachtet werden müssen aber auch der nahe liegende, bestimmungswidrige Gebrauch.¹⁰⁸

Larenz teilt Verkehrspflichten in drei Kategorien ein¹⁰⁹:

- Sicherheit bezogen auf den eigenen Bereich
- Übernahme von Aufgaben
- vorangegangenes, besonders gefährliches Tun

Hauptansatzpunkt im Rahmen der Open Source Software dürfte die Haftung für die Sicherung des eigenen Bereichs sein. Eine übernommene Aufgabe ist bei der Entwicklung kaum einmal gegeben¹¹⁰ und auch vorangegangenes besonders gefährliches Tun dürfte lediglich in Randbereichen denkbar sein¹¹¹. Hauptaugenmerk wird daher die Bereichshaftung sein.

4.3.2.2 Nahe liegender Gebrauch

Als nahe liegender Gebrauch ist im Rahmen der Open Source Soft-

¹⁰⁶ Vgl. Larenz S. 403.

¹⁰⁷ BGH NJW 85, 1076.

¹⁰⁸ Vgl. zum gesamten Komplex Palandt-Thomas § 823 BGB Rn. 58 bzw. allgemeiner Larenz S. 399 ff.

¹⁰⁹ Larenz S. 411.

¹¹⁰ Auch wenn die Entwicklung von Open Source Software als Werkvertrag denkbar erscheint.

¹¹¹ Denkbar erscheint z.B. die Programmierung eines mit erheblichen Mängeln behafteten Programms im medizinischen Bereich.

ware auch die Bearbeitung des Sourcecodes anzusehen. Der ursprüngliche Entwickler müsste also auch für mögliche Veränderungen Vorsorge treffen. Dies ist praktisch aber nicht möglich. Wer den Quellcode in Händen hält, kann damit tun, was immer er will. Vorkehrungen des ursprünglichen Autors können ohne große Probleme umgangen werden, sie sind wirkungslos. Einer der Hintergründe der Verkehrspflichten ist aber die Verantwortung für einen – je nach Fall anders ausgestalteten – Herrschaftsbereich¹¹². Ein solcher Herrschaftsbereich besteht mit Weggabe des Quellcodes aber wie ausgeführt nicht (mehr). Es kann im Rahmen der Open Source Software mithin nicht gefordert werden, dass für Veränderungen Dritter am Quellcode im Rahmen einer Verkehrspflicht gehaftet wird. Denkbar erscheint aber die Verpflichtung, einen Hinweis in den Quellcode aufzunehmen, wenn unbedachte Veränderungen zu erheblichen Fehlfunktionen führen könnten. Daneben ist aber auch zu beachten, dass der Entwickler damit rechnen kann, dass ein Bearbeiter weiß, wie er mit dem Quellcode umzugehen hat. Dieser Aspekt der Selbstgefährdung kann aber nur gegenüber dem Bearbeiter greifen, einem Dritten kann er nicht vorgehalten werden.

4.3.2.3 Vorteil des Verursachers

Weiterer Ansatzpunkt bei der Begründung von Verkehrspflichten ist der finanzielle Vorteil, der sich aus der Schaffung einer Gefahrenquelle für den Verursacher ergibt. Dieser ist aber – vom Prestige abgesehen – im Bereich der Open Source Software gering bzw. nicht vorhanden, womit die Anforderungen, die an den Entwickler gestellt werden, gesenkt werden müssen.

4.3.2.4 Preis

Beachtet man die Sicherheitserwartungen der betroffenen Verkehrskreise, so ist auf die kostenlose Übertragung der Software hinzuweisen. Anders als bei kostenpflichtiger Software kann der Nutzer bei kostenloser Software keinen entsprechenden Sicherheitsstan-

¹¹² Vgl. Larenz S. 400.

dard erwarten. Gleichzeitig ist aber darauf hinzuweisen, dass Open Source Software häufig als Ersatzprodukt zu proprietärer Software angepriesen wird¹¹³, was einem völlige Herabsetzen der Sicherheits-erwartungen entgegensteht.

Letztlich muss daher gesagt werden, dass der Nutzer zwar kein fehlerfreies Produkt erwartet, wie dies meist bei kostenpflichtiger Software der Fall ist, jedoch auch bei Open Source Software gewisse Sicherheitserwartungen vorhanden sind.

Als Abgrenzung bietet sich der eigene Anspruch des Produkts an: Je höher dieser ist, je eher also ein kostenpflichtiges Produkt ersetzt werden soll, desto höher sind auch die Sicherheitserwartungen¹¹⁴. Diese sind wegen der Kostenfreiheit aber dennoch niedriger als bei proprietärer Software.

Verkehrspflichten können auch durch Selbstschutzargumente eingeschränkt werden¹¹⁵. Ist für den Nutzer eine bestimmte Gefährdung erkennbar, so ist er vor dieser nicht mehr explizit zu schützen. Im Bereich der Software ist diesbezüglich an Alpha-, Beta- oder Unstable-Releases zu denken. Diese werden gerade zu Testzwecken und mit dem Hinweis, dass noch nicht alle Bereiche fehlerfrei funktionieren, veröffentlicht, so dass kein Nutzer überrascht sein sollte, kommt es tatsächlich zu einem Ausfall. Ein Missbrauch dergestalt, dass nur noch Unstable-Releases veröffentlicht werden, ist kaum vorstellbar¹¹⁶, dem könnte aber gegebenenfalls unter Umgehungsgesichtspunkten begegnet werden.

Daneben könnte zur Bestimmung der Verkehrspflichten auch die Gedanken der ökonomischen Analyse des Rechts herangezogen werden. Diese fragen danach, wer die auftretenden Kosten eher tragen kann. Abzugrenzen sind die Kosten der Schadensver-

113 Vgl. z.B. Linux, OpenOffice, MySQL oder Apache.

114 So sind die Erwartungen, die in Linux oder OpenOffice gesetzt werden, höher, als die, die in die diversen unter SourceForge.Net angebotenen Spiele gesetzt werden.

115 Vgl. hierzu Larenz S. 414 f; BGH NJW 1985, 1076 f.

116 Der Endnutzer, um den es hier primär geht, wird selten als „Unstable“ bezeichnete Versionen installieren, womit bei einem entsprechenden Vorgehen dieses, immer wichtiger werdende Marktsegment abgeschnitten würde.

hinderung von der Höhe des eventuell auftretenden Schadens¹¹⁷.

Beachtet werden muss hierbei wiederum, dass Open Source Software kostenfrei übertragen wird, dem Entwickler ein überhöhtes finanzielles Engagement folglich nicht zugemutet werden kann. Hieraus kann geschlossen werden, dass die Anforderungen an die Annahme einer Verkehrspflicht gesenkt werden müssen.

4.3.2.5 Erkennbarkeit

Daneben muss zur Annahme einer Verkehrspflicht die Schadensmöglichkeit erkennbar sein. Erst dann wirkt die Gefahrschaffung haftungsbegründend¹¹⁸.

Inwieweit mit dem Einsatz der eigenen Software in haftungstechnisch schwierigem Umfeld gerechnet werden muss, ist Frage des Einzelfalls. Am Beispiel eines mit Linux ausgestatteten medizinischen Geräts könnte dies jedoch z.B. dann problematisch sein, wenn lediglich eine abgespeckte Version des Betriebssystems verwendet wird. Für den Entwickler kann insbesondere nicht erkennbar sein, in welcher Version seine Software später verwendet wird. Insoweit könnte die Haftungsbegründung scheitern.

4.3.2.6 Zwischenergebnis

Letztlich muss festgestellt werden, dass die Entscheidung, welche Verkehrspflichten bestehen, eine Frage des Einzelfalls ist. Dabei können die soeben angeführten Kriterien als Ansatzpunkte herangezogen werden. Im Vergleich zu proprietärer Software steht der Entwickler von Open Source Software wegen deren Eigenheiten jedoch haftungstechnisch günstiger.

4.3.2.7 Person des Verkehrspflichtigen

Verkehrspflichtig ist derjenige, der die Bestimmungsgewalt über die Gefahrenquelle innehat¹¹⁹.

Also auch derjenige, der eine Gefahrenquellen schafft oder andau-

117 Vgl. Larenz S. 417.

118 Palandt-Thomas § 823 Rn. 58.

119 Larenz S. 418; Palandt-Thomas § 823 Rn. 59.

ern lässt¹²⁰. Daraus folgt aber auch, dass derjenige die Gefahr wieder aus der Welt schaffen können muss. Dies ist bei Open Source Software u.U. schwierig, ist der Verbleib des Quellcodes doch unkontrollierbar. Um aus der Verkehrspflicht entlassen zu werden, erscheint es daher denkbar, eine korrigierte Version und einen Hinweis an der Stelle, an der der Code ursprünglich veröffentlicht wurde, zu verlangen. Inwieweit dies aber angesichts der Verbreitung der Software genügen kann, ist Frage des Einzelfalls.

4.3.3 Verschulden

Auch wenn die Verkehrspflichten gerade bei fahrlässigen Verletzungen gebraucht werden, ist neben einer reinen Verletzung auch noch das Verschulden zu beachten. Dieses muss sich auf die in § 823 BGB genannten Rechtsgüter beziehen, nicht nur auf die Verletzung der Verkehrspflicht¹²¹. Es wird jedoch bei der Verletzung einer Verkehrspflicht regelmäßig indiziert.

Ab wann von Fahrlässigkeit auszugehen ist, bestimmt sich danach, inwieweit ein Fehler objektiv erkennbar und vermeidbar war¹²², ist folglich eine Frage des Einzelfalls.

4.3.4 Produkthaftung im Rahmen des § 823 I BGB

Neben den soeben dargestellten Voraussetzungen des § 823 BGB sind gegebenenfalls die Grundsätze der deliktischen Produkthaftung zu beachten¹²³. Diese orientiert sich an den Voraussetzungen des ProdHaftG. Es gelten also die dort gemachten Aussagen zum Herstellerbegriff¹²⁴. Betroffen sind wiederum Organisations-, Instruktions- und Produktbeobachtungspflichten¹²⁵.

Für die Frage der Haftungsbegründung ist insbesondere auf die Un-

¹²⁰ So Larenz S. 407.

¹²¹ Larenz S. 426.

¹²² Vgl. hierzu BGH NJW 1994, 517, 520; BGH NJW 1991, 1553; Palandt-Heinrichs § 276 Rn. 15 ff.

¹²³ Vgl. allgemein Palandt-Thomas § 823 BGB Rn. 201 ff.

¹²⁴ Allerdings ist der Kreis der Ersatzpflichtigen kleiner: Der Importeur, Vertriebs- händler oder Lieferant gehört nicht dazu, vgl. Palandt-Thomas § 823 BGB Rn. 216.

¹²⁵ Palandt-Thomas § 823 BGB Rn. 206 ff.

terschiede zum ProdHaftG in Bezug auf die Beweislast hinzuweisen¹²⁶. Zwar hat auch hier der Geschädigte den Fehler, den Schaden und die Ursächlichkeit zu beweisen, u.U. wird ihm aber eine Beweislastumkehr im Hinblick auf die ursprüngliche Fehlerfreiheit gewährt¹²⁷. Doch muss der Hersteller beweisen, dass ihn kein Verschulden bzgl. des Fehlers trifft, es kommt also zu einer Beweislastumkehr¹²⁸. Dieser Beweis ist schwer zu führen.

Für den Bereich der Open Source Software ist wiederum darauf hinzuweisen, dass einige Entwickler nicht unter den Begriff des Herstellers iSd. § 4 ProdHaftG und damit des § 823 I BGB fallen dürften.

4.3.5 Ergebnis

Für Entwickler von Open Source Software besteht die Gefahr einer Haftung aus § 823 BGB. Wegen den gegenüber Entwicklern proprietärer Software geringeren Anforderungen ist eine solche Gefahr aber gering. Dies gilt auch gegenüber Dritten, da auch bei diesen die angepassten Verkehrspflichten beachtet werden müssen.

4.4 ProdSG

Auch das ProdSG¹²⁹ bestimmt, welche Anforderungen im Hinblick auf die Fehlerfreiheit von Produkten an einen Hersteller zu stellen sind. Dabei ist der Aufbau mit dem ProdHaftG vergleichbar.

Einer der Hintergründe des Gesetzes ist es, nur sichere Produkte in den Verkehr zu lassen, vgl. § 1 Nr. 1 ProdSG.

Auch wenn der Begriff des Herstellers etwas weiter gefasst ist, als im ProdHaftG, greift das ProdSG nur bei Personen, die gewerbs- oder geschäftsmäßig tätig werden, § 1 S. 1 ProdSG.

Mithin kann allgemein auf die Ausführungen zum ProdHaftG verwiesen werden.

Das ProdSG gewährt aber, anders als das ProdHaftG, keinen

¹²⁶ Daneben besteht keine Höchstsumme, keine Selbstbeteiligung und kein Ausschluss des Schmerzensgeldes.

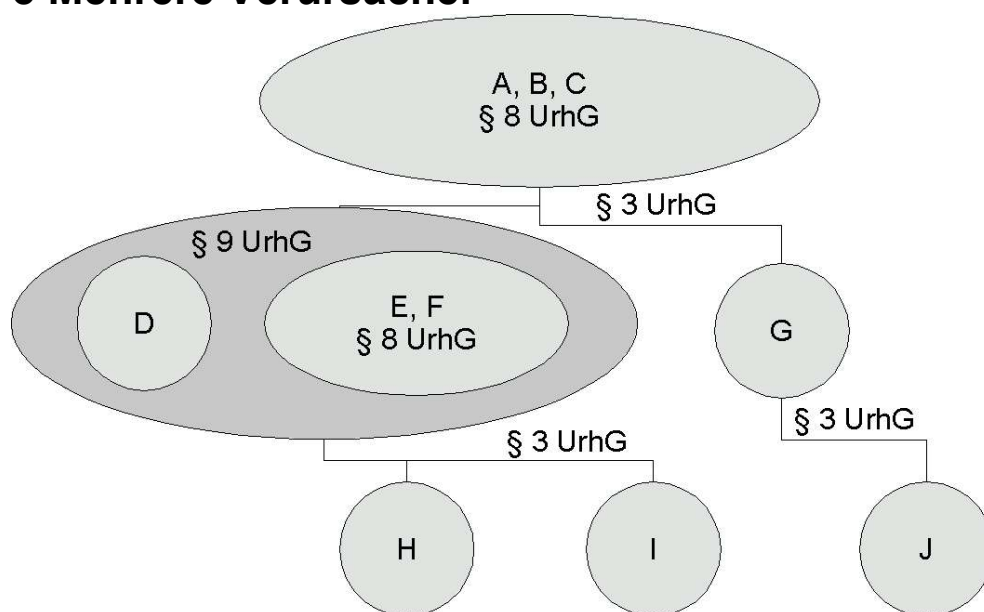
¹²⁷ BGH 104, 323; NJW 93, 528.

¹²⁸ Vgl. Palandt-Thomas § 823 BGB Rn. 220 m.w.N.

¹²⁹ Allgemein zum ProdSG iVm. Computerleistungen Littbarski in Kilian/Heussen § 180 Rn. 168 ff.

eigenen Schadensersatzanspruch, statuiert vielmehr nur Eingriffsbefugnisse der zuständigen Behörden, vgl. §§ 7 ff., 15 ProdSG. Auch stellt es kein Schutzgesetz iSd. § 823 II BGB dar¹³⁰. Aus diesen Gründen ist es für die Haftung nur dergestalt von Bedeutung, als dass sich durch den im ProdSG geforderten Sicherheitsstatus die allgemeinen Anforderungen im Rahmen des § 823 I BGB ändern könnten (Verkehrspflichten). Hierbei muss aber beachtet werden, dass das ProdSG nicht für alle möglichen Beteiligten gilt, so dass auch Folgerungen für die Verkehrspflichten nicht auf allen Ebenen gezogen werden können.

5 Mehrere Verursacher



Bis zu dieser Stelle wurde bei einem Schaden stillschweigen immer von einem Verursacher ausgegangen. Gerade bei Open Source Software erscheint es aber denkbar, dass Mehrere beteiligt waren. Besteht wegen der Urheberschaft eine Gesellschaft, so haftet diese nach außen hin im vollen Umfang, nach innen erfolgt ein Ausgleich¹³¹. Haftbar gemacht werden kann letztlich also jeder Entwickler.

Stehen mehrere Entwickler(gemeinschaften) nebeneinander, muss differenziert werden.

¹³⁰ Vgl. Kilian/Heussen-Littbarski § 180 Rn. 175.

¹³¹ Vgl. Palandt-Spau § 705 Rn. 24.

Im vertraglichen Bereich stehen diese Entwickler selbstständig nebeneinander. Jeder haftet für den Schaden, für den er verantwortlich ist.

Im deliktischen Bereich gilt jedoch § 830 I BGB. Trotz selbstständigem Beitrag haften die Beteiligten gesamtschuldnerisch, § 840 I BGB.

Gleiches gilt bei einer Haftung aus dem ProdHaftG, vgl. § 5 ProdHaftG.

Dennoch muss auch im Bereich der Gesamtschuldnerschaft differenziert werden:

Weiter oben¹³² wurde dargelegt, dass für Fehler, die andere zu einem späteren Zeitpunkt verursachen, der ursprüngliche Autor nicht haftbar ist. Wie aber ist zu entscheiden, wenn z.B. G einen Fehler unterlaufen ist, J die Software verändert, den Fehler aber nicht bemerkt?¹³³

Es muss zwischen Freizeitprogrammierern und Unternehmen unterschieden werden. Während der Verkehr von einem Unternehmen wie z.B. IBM erwartet, dass dieses Software, die veröffentlicht wird, kontrolliert, ist dies bei einem einzelnen Programmierer nicht der Fall. Eine entsprechende Verkehrspflicht, alles zu kontrollieren, besteht nicht. Sie würde im Rahmen größerer Projekte wie z.B. Linux Unmögliches verlangen. Besteht aber eine entsprechende Verkehrspflicht nicht, haftet der Entwickler nicht selbstständig, weshalb eine Gesamtschuldnerschaft ausscheidet.

Allgemein kann daher gesagt werden, dass nur derjenige haftet, der den Schaden verursacht hat.

132 4.3.2.2.

133 Vgl. Grafik.

6 Zusammenfassung

Allgemeines

- ✓ Open Source Software wird durch eine Schenkung übertragen.
- ✓ Bei der gemeinsamen Entwicklung von Open Source Software entsteht im allgemeinen *keine* Gesellschaft sondern ein – praktisch schwer zu durchschauendes – Geflecht von Beziehungen.

Haftungstatbestände

- ✓ Im Bereich der *vertraglichen Haftung* ergeben sich wegen § 6 GPL. Probleme, die sich jedoch mit einer entsprechenden Auslegung lösen lassen.
- ✓ Ungeachtet der Frage, inwieweit Software eine Sache iSd. § 90 BGB darstellt, fällt Open Source Software unter das *ProdHaftG*. Allerdings ist jeweils genau zu prüfen, inwieweit ein Ausnahmetatbestand greift.
- ✓ Die Haftungsprivilegierung aus dem Schenkungsrecht greift auch im *Deliktsrecht*, nicht aber gegenüber Dritten. Problematisch und Frage des Einzelfalls ist die Bestimmung der Verkehrspflichten.

Der Haftungsfall

- ✓ Für die Klärung der Frage, welches Haftungsregime jeweils anzulegen ist, ist zwischen Unternehmern und Freizeitprogrammierern zu trennen.
- ✓ Die verschiedenen Entwickler von Open Source Software haften, wenn überhaupt, nur für die von ihnen zu verantwortenden Schäden. Eine Gesamtschuldnerschaft besteht regelmäßig nicht, mit Ausnahme des § 8 UrhG. Daneben ist aber eine deliktische Gesamtschuldnerschaft, § 840 BGB, möglich.
- ✓ Unbeachtet der sowieso bestehenden beweisrechtlichen Probleme bei einem Schadensfall ist das Haftungsrisiko für den einzelnen Freizeitprogrammierer gering.
Für Unternehmen erhöht sich das Risiko u.a. wegen dem ProdHaftG.