

Open Source Software

Jaeger / Metzger

5. Auflage 2020
ISBN 978-3-406-73497-7
C.H.BECK

sprechende Nutzungsrechte vergibt. Erfolgt die Lizenzierung durch den Arbeitgeber oder durch den Inhaber ausschließlicher Nutzungsrechte, so kommt allenfalls eine analoge Anwendung in Betracht. Der Gesetzgeber hat diesen Fall offensichtlich nicht bedacht. Es ist jedoch anzunehmen, dass auch hier die Lizenzierung im Hinblick auf unbekannte Nutzungsarten nicht faktisch ausgeschlossen werden sollte.

Dagegen steht auch den Urhebern von Open Source Software das uneingeschränkte **Widerrufsrecht** gem. § 31a Abs. 1 S. 3 UrhG zu. Die Möglichkeit des „Opt-out“ einzelner Beteiligter an Open Source Entwicklungen im Hinblick auf unbekannte Nutzungsarten schränkt die Nutzbarkeit entsprechender Programme nicht in unzumutbarer Weise ein. Im Übrigen bestehen für den Widerruf erhöhte Anforderungen, wenn die Programmierung des betreffenden Entwicklers mit Programmbestandteilen anderer Programmierer zu einer Gesamtheit im Sinne von § 31a Abs. 3 UrhG zusammengefasst sind. Verhält sich der Urheber treuwidrig, so ist der Widerruf ausgeschlossen.

Eine weitere Linux-Klausel enthalten die Vorschriften zum Vergütungsanspruch gem. § 32c UrhG. Urheber können zwar Rechte an unbekanntem Nutzungsarten vergeben. Ihnen steht zum Ausgleich aber ein Vergütungsanspruch gegen den Vertragspartner zu. Auch hier greift eine **Ausnahmevorschrift für Open Source Lizenzen**. Zwar kann der Urheber auf den Vergütungsanspruch im Grundsatz nicht verzichten. Er kann gem. § 32c Abs. 3 S. 2 UrhG aber „*unentgeltlich ein einfaches Nutzungsrecht für jedermann einräumen*.“ Ausweislich der Begründung des Regierungsentwurfs soll damit im Einklang mit den Regelungen in §§ 32, 32a UrhG eine Nachforderung des Urhebers bei Verwendung entsprechender Lizenzmodelle ausgeschlossen werden.⁷⁴⁰ Dies ist im Interesse der Rechtssicherheit zu begrüßen. Allerdings wird man auch hier die Einschränkung machen müssen, dass der Ausschluss nur dann greift, wenn der Urheber selbst eine Freigabe als Open Source vorgenommen hat. Erfolgt die Freigabe durch den Arbeitgeber oder den Inhaber ausschließlicher Nutzungsrechte, kann es durchaus zu einer Nachforderung durch Urheber kommen.

Unabhängig davon, ob es sich um einen vor oder nach dem 1.1.2008 geschlossenen Open Source Lizenzvertrag handelt, ist von Bedeutung, ob die betreffende Nutzungsart zum Zeitpunkt der Freigabe durch den Urheber bekannt gewesen ist. Für unbekannte Nutzungsarten bleibt es bei Verträgen, die vor dem 1.1.2008 geschlossen wurden, bei der Anwendung des § 31 Abs. 4 UrhG a.F., für „Neuverträge“ ist vor allem das Widerrufsrecht des § 31a Abs. 1 S. 3 UrhG relevant. Hierbei sollte allerdings beachtet werden, dass Open Source Entwickler ihre Programme oftmals weiterentwickeln und in einer neuen Version zu einem späteren Zeitpunkt erneut als Open Source freigeben. Diese spätere Lizenzierung umfasst dann für die gesamte von diesem Urheber entwickelte Software die bis dahin bekannten Nutzungsarten, sofern diese dem Lizenztext nach unter die Lizenz fallen.⁷⁴¹

An den genannten Einschränkungen der Disponibilität hinsichtlich unbekannter Nutzungsarten ändert auch ein Vermerk des Urhebers im Sinne der Ziffer 9 GPL-2.0 nichts, sein Programm der jeweils aktuellen Version der GPL unterstellen zu wollen.⁷⁴² Nach Ziffer 9 GPL-2.0 behält sich die Free Software Foundation das Recht vor, neue Versionen der GPL zu veröffentlichen.

⁷⁴⁰ BT-Drs. 16/1828, S. 25.

⁷⁴¹ Redeker-Jaeger, Handbuch IT-Verträge, Kap. 1.20, Rn. 112 ff.

⁷⁴² Siehe hierzu auch unten Rn. 263 f.

- 193** Vorgesehen ist, dass Urheber, die Software der GPL unterstellen, in einem Vermerk darauf hinweisen können, dass das Programm auch nach den Bestimmungen „jeder späteren Version“ der Lizenz genutzt werden kann. Die jeweils aktuellste Version soll darüber hinaus für all die Software gelten, die keinen Hinweis auf eine bestimmte Version enthält. Enthalten neue Versionen der GPL die Einräumung von Rechten für Nutzungsarten, welche zum Zeitpunkt der ersten Lizenzierung noch nicht bekannt waren,⁷⁴³ so bleiben trotz eines Vermerks im Sinne der Ziffer 9 GPL-2.0 für diese Nutzungsarten die Restriktionen des § 31 Abs. 4 UrhG a.F. bzw. §§ 31a, 32c UrhG n.F. zu beachten. Der Urheber muss auch hier vor Geschäften geschützt werden, deren wirtschaftliche Tragweite zum Zeitpunkt des Vertragschlusses noch nicht richtig eingeschätzt werden kann. Praktischer Relevanz haben diese rechtlichen Aspekte bislang aber noch nicht erlangt.

V. Nutzung im Wege des Software as a Service (SaaS)

- 194** Softwarenutzung im Wege des Netzwerkzugriffs, zumeist über das Internet, hat sich zu einer wirtschaftlich relevanten Nutzungsform entwickelt. Damit erhält der Nutzer keine Kopie der Software, die er bei sich installiert, sondern verwendet die Funktionalität, die auf einem Rechner eines Dritten abläuft. Dieser wird dann als *Application Service Provider*, *Cloud-Dienstleister* oder *SaaS-Anbieter* bezeichnet.⁷⁴⁴ In diesem Zusammenhang stellen sich einige spezielle Fragen der Open Source Lizenzierung, die nachfolgend behandelt werden.

1. Berücksichtigung von SaaS in freien Lizenzen

- 195** In einigen moderneren Lizenzversionen, wie der GPL-3.0, AGPL-3.0, LGPL-3.0 und der MPL-2.0 wird SaaS speziell berücksichtigt und in allen genannten Lizenzen auch entsprechende Nutzungsrechte eingeräumt. Insoweit kann auf die Ausführungen zu den entsprechenden Lizenzen verwiesen werden.⁷⁴⁵ Ältere Lizenzen, mit Ausnahme der speziell dafür entwickelten Affero General Public License, Version 1,⁷⁴⁶ erwähnen diese Nutzungsart nicht, da diese bis etwa Mitte/Ende der 1990er Jahre nicht bekannt war und danach zunächst keine relevante wirtschaftliche Rolle spielte.⁷⁴⁷

2. Einräumung der erforderlichen Nutzungsrechte

- 196** Die Frage, ob Open Source Software auch im Wege des SaaS genutzt werden darf, ist zu Recht stärker in den Fokus gerückt.⁷⁴⁸ Zusätzlich komplex wird das Thema durch den Umstand, dass in den USA offenbar kein besonderes urheberrechtliches Nutzungsrecht – d.h. über die Berechtigung zur Benutzung der auf dem Server installierten Programmkopie hinaus – für erforderlich erachtet wird. Insbesondere

⁷⁴³ Hierbei ist auf die Urhebersicht abzustellen, dies allerdings anhand eines generalisierenden Maßstabs, vgl. *Schricker/Loewenheim-Spindler*, § 31a, Rn. 27.

⁷⁴⁴ Dazu auch Rn. 37, 188 und 355. Zu technischen Unterschieden vgl. *Poble/Ammann*, K&R 2009, 625, 626 und *Söbbing*, ITRB 2015, 147f. Urheberrechtlich dürften sich die jedoch nicht auswirken, vgl. *Marly*, Rn. 114 ff.

⁷⁴⁵ Vgl. Rn. 76, 85, 102.

⁷⁴⁶ <http://www.affero.org/oagpl.html>.

⁷⁴⁷ Hier werden unterschiedliche Zeitpunkte vorgeschlagen, vgl. *Hilber-Paul/Niemann*, Handbuch Cloud Computing, Teil 3, Rn. 81, 204.

⁷⁴⁸ Vgl. *Redeker-Jaeger*, Handbuch IT-Verträge, Kap. 1.20, Rn. 112 ff.; *Ballhausen*, IFOSSLR 6 (1), 2014, 61; *Hilber/Reintzsch*, CR 2014, 697; *Hilber-Paul/Niemann*, Handbuch Cloud Computing, Teil 3, Rn. 202 ff.

fällt SaaS nicht unter das „*right to distribute*“, da Sec. 106 (3) US Copyright Act dafür die Überlassung einer Kopie verlangt.⁷⁴⁹ Damit wird aus Sicht des US-Rechts die Befugnis zum SaaS durch das Vervielfältigungsrecht bzw. das Recht zum einfachen Programmablauf (z. B. Ziffer 0 GPL-2.0: „*The act of running the Program is not restricted*“) gewährt und Lizenzpflichten fallen entsprechend nicht an.⁷⁵⁰

Unter der Anwendung deutschen Urheberrechts ist die Rechtslage schwieriger. Nach überwiegender Auffassung benötigt der SaaS-Anbieter das **Recht der öffentlichen Zugänglichmachung** gem. § 69c Nr. 4 UrhG, um Dritte die Funktionalität eines Computerprogramms im Wege des Netzwerkzugriffs nutzen zu lassen.⁷⁵¹ Davon ist wohl der Fall abzugrenzen, dass über den Server keine **Anwendungen** zugänglich gemacht werden, so dass keine zusätzliche Nutzungen durch Dritte erfolgen, sondern die Nutzung im Wesentlichen nur beim Serverbetreiber stattfindet (z. B. Software zum Betrieb des Webserver und andere „**Infrastruktursoftware**“) und der auf den Server zugreifende Nutzer nur mittelbar davon profitiert. Die Abgrenzungen sind hier im Einzelnen schwierig und noch wenig untersucht. Das auch in Ziffer 13 AGPL-3.0 verwendete Kriterium, wonach darauf abzustellen ist, dass Nutzer auf die Software interaktiv über ein Computernetzwerk zugreifen („*users interacting with it remotely through a computer network*“), erscheint interessensgerecht.

Wenn der Lizenznehmer das Nutzungsrecht zur öffentlichen Zugänglichmachung für ein SaaS benötigt, fragt sich, ob dieses durch die Open Source Lizenzen, die dazu keine besondere Aussage machen, wirksam eingeräumt wird. Zum einen stellt sich das Problem, dass die Nutzung im Wege des SaaS als neue Nutzungsart anzusehen sein dürfte, die vor Mitte/Ende der 1990er Jahre unbekannt war.⁷⁵² Zum anderen fragt sich, ob Open Source Lizenzen dahingehend ausgelegt werden können, dass auch ohne explizite Benennung des Rechts zur öffentlichen Zugänglichmachung eine Rechtseinräumung erfolgen kann.

a) SaaS als neue Nutzungsart

Gem. § 31 Abs. 4 UrhG in der bis zum 31.12.2007 gültigen Fassung war die Einräumung von Nutzungsrechten an noch **nicht bekannten Nutzungsarten** unwirksam. Daher darf Code, der in der Zeit vor Ende 2007 und vor Bekanntwerden der Nutzungsart entwickelt wurde, in Deutschland nicht im Wege des SaaS genutzt werden. Das Problem der unbekanntenen Nutzungsarten ist damit praktisch relevant, obwohl die Änderung des § 31 UrhG und die dabei erfolgte Privilegierung von Open Source Lizenzen in § 31a Abs. 1 S. 2 UrhG das Problem für neuere Programmierungen weitgehend erledigt hat.⁷⁵³ Nur wenn der Rechteinhaber seine

⁷⁴⁹ Meeker, IFOSSLR 4 (1), 2012, 29 ff.; Meeker, Open Source for Business, S. 72.

⁷⁵⁰ Unklar ist, ob (auch) die Rechte zum „*public display*“ und „*public performance*“ einschlägig sind, die laut U. S. Copyright Office für das Streaming relevant sein sollen, vgl. U. S. Copyright Office, The Making Available Right in the United States, February 2016, https://www.copyright.gov/docs/making_available/making-available-right.pdf. Dazu auch SIA, Comments on the Right of Making Available, https://www.copyright.gov/docs/making_available/making-available-right.pdf.

⁷⁵¹ Vgl. Schricke/Loewenheim-Spindler, Vor § 69a ff., Rn. 67; Marly, Rn. 1101 ff.; Dreier/Schulze-Dreier, § 69c, Rn. 36a; ebenso OLG München, CR 2009, 500, 502. Anders Wandtke/Bullinger-Grützmacher, § 69c, Rn. 66, der SaaS auf die Übertragung von Grafikdaten reduziert; dabei wird aber übersehen, dass beim SaaS regelmäßig auch die Funktionalität der Software genutzt wird, etwa durch die Erstellung oder Verarbeitung von Daten.

⁷⁵² Die Details sind umstritten, vgl. Redeker-Gennen/Laue, Handbuch IT-Verträge, Kap. 1.17, Rn. 202; Dietrich ZUM 2010, 567, 570.

⁷⁵³ Die Übergangsregelung des § 137l UrhG ist nicht anwendbar, s. o. Rn. 188.

Software nach Bekanntwerden der neuen Nutzungsart des SaaS vertrieben oder gar bearbeitet und dadurch eine erneute Lizenzierung der Software erklärt hat, kann begründet werden, dass das Verbot der Lizenzierung für unbekanntes Nutzungsarten keine Rolle mehr spielt.⁷⁵⁴ Es bleibt aber die grundsätzliche Frage, wie weitgehend Open Source Lizenzen in dieser Hinsicht auszulegen sind.

b) Vertragsauslegung

- 198** In der klassischen Anwendung der Zweckübertragungsregel des § 31 Abs. 5 UrhG wird angenommen, dass die Rechte im Zweifel so weit wie möglich beim Urheber verbleiben.⁷⁵⁵ Es ist allerdings zweifelhaft, ob diese schematische Anwendung den Besonderheiten von Open Source Lizenzen gerecht wird. Denn Open Source Lizenzen bezwecken eine **möglichst weitgehende Rechtseinräumung**, von der – zumindest bei Copyleft Software – auch der Lizenzgeber profitiert, wenn er selbst Lizenznehmer von bearbeiteten Programmversionen wird.⁷⁵⁶ Wenn Nutzungsrechte aber möglichst weitgehend eingeräumt werden sollen, dann ist es auch naheliegend, dass die Nutzung im Wege des SaaS gestattet werden soll. Zudem ist im Rahmen der Zweistufenlehre⁷⁵⁷ zu berücksichtigen, dass ein SaaS nach Verständnis des US-Rechts, wie oben gezeigt, unproblematisch möglich ist. Dem entspricht auch das Verständnis in der Praxis: Open Source Software wird in erheblichem Umfang im Rahmen von SaaS genutzt und bislang ist nicht bekannt geworden, dass Rechteinhaber dagegen Einwände geltend gemacht haben. Nach alledem ist ein Verständnis vorzuzugungswürdig, wonach Lizenznehmer Freie Software auch im Wege des SaaS verwenden dürfen.⁷⁵⁸

3. Erfüllung von Lizenzpflichten

- 199** Wenn man davon ausgeht, dass Freie Software im Wege des SaaS genutzt werden darf, stellt sich die Folgefrage, ob dann auch die Lizenzpflichten erfüllt werden müssen, die zumeist an einen Vertrieb der Software anknüpfen. *Hilbert/Reintzsch* stellen zur Beantwortung der Frage darauf ab, „welchen Bedeutungsgehalt der Verfasser des Lizenztextes, dessen Rechtsauffassung sich der Lizenzgeber zu Eigen macht, den fraglichen Bestimmungen beigemessen hat“.⁷⁵⁹ Aus dem Umstand, dass die FSF als *License Steward* davon ausgeht, dass beim SaaS keine Lizenzpflichten zu erfüllen sind, wird eine entsprechende Auslegung der GPL-2.0 gefolgert. Dagegen sprechen allerdings eine Reihe von Gesichtspunkten: Zunächst ist festzustellen, dass zu dem Zeitpunkt, als die GPL-2.0 geschrieben wurde (1991), SaaS noch unbekannt war und die FSF dem entsprechend der Lizenz auch keinen Bedeutungsgehalt im Hinblick auf SaaS beigegeben hat. Zudem wurde bereits oben gezeigt, dass unter der Geltung von US-Recht keine besonderen Nutzungsrechte für SaaS benötigt werden und daher auch keine Pflichten an erforderliche zusätzliche Nutzungsrechte geknüpft werden konnten. Daher hat die FSF auch von einem „ASP-Loophole“ gesprochen.⁷⁶⁰

⁷⁵⁴ *Redeker-Jaeger*, Handbuch IT-Verträge, Kap. 1.20, Rn. 112 ff.

⁷⁵⁵ BGH, GRUR 1998, 680, 682 – *Comic-Übersetzungen*.

⁷⁵⁶ So auch *Jaeger/Mantz*, Anmerkung zu LG Köln, MMR 2014, 480, 482, für die insoweit vergleichbaren Creative Commons-Lizenzen. Dem folgt im Ergebnis auch OLG Köln, MMR 2015, 331.

⁷⁵⁷ S. u. Rn. 454.

⁷⁵⁸ *Redeker-Jaeger*, Handbuch IT-Verträge, Kap. 1.20, Rn. 123; offen gelassen bei *Hilbert/Reintzsch*, CR 2014, 697, 701 f.

⁷⁵⁹ *Hilbert/Reintzsch*, CR 2014, 697, 700.

⁷⁶⁰ Vgl. <http://www.zdnet.com/article/gpl-3-may-tackle-web-loophole-5000144830/> und <https://www.fsf.org/blogs/licensing/2007-03-29-gplv3-saas>.

Geht man hingegen von einer **ergänzenden Vertragsauslegung** aus,⁷⁶¹ dann ergibt sich zwanglos, dass die Rechteinhaber zwar einerseits möglichst weitgehend Nutzungsrechte einräumen wollen, andererseits aber auch Nutzungen, die nicht nur im internen Bereich des Lizenznehmer stattfinden, das Copyleft und alle anderen Lizenzpflichten auslösen.⁷⁶² Dagegen spricht auch nicht der Gesichtspunkt, dass es sich bei Open Source Lizenzen um AGB handelt. Denn vorliegend geht es nicht darum, unklare Vertragsregelungen zuungunsten des Verwenders auszulegen, sondern eine offenkundige Regelungslücke durch eine der Intention des Lizenzgebers angemessenen Weise zu schließen.⁷⁶³ Dies geht auch nicht zu Lasten des Lizenznehmers, da ohne ergänzende Vertragsauslegung die Software überhaupt nicht im Wege des SaaS genutzt werden dürfte.⁷⁶⁴ Es erscheint daher auch im Hinblick darauf, dass bei einem kollaborativen Entwicklungsmodell Lizenzgeber zugleich auch Lizenznehmer sind, angemessen, Rechte und Pflichten aus den Lizenzen in ihrer Wechselwirkung auszulegen.

C. Rechtsinhaberschaft

I. Wer ist Urheber bei der „Basar-Methode“?⁷⁶⁵

Betrachtet man Software, die auf herkömmliche Art entwickelt wird, so ist die Frage nach der Urheberschaft – abgesehen von mitunter schwierigen Beweisfragen im Einzelfall – zumeist einfach zu beantworten. Entweder ein einzelner Programmierer hat die Software geschrieben oder mehrere Urheber, die typischerweise bei einem Arbeitgeber angestellt sind, waren beteiligt. 200

Diese einfachen Strukturen der Softwareentwicklung existieren natürlich auch bei Freier Software. In diesem Bereich gibt es ebenfalls den einzelnen Urheber, der ein Programm schreibt, um es dann unter eine Open Source Lizenz zu stellen. Typisch sind auch Konstellationen, in denen ein Unternehmen alle Urheberrechte an einer Software hält und diese dann unter einer freien Lizenz verbreitet.⁷⁶⁶ Die erfolgreichsten freien Programme aber sind auf eine andere Art entstanden. Für die Entwicklung Freier Software ist die Mitarbeit vieler unabhängiger Programmierer typisch,⁷⁶⁷ die mitunter weltweit verstreut sind und via Internet kommunizieren

⁷⁶¹ Vgl. *Redeker-Jaeger*, Handbuch IT-Verträge, Kap. 1.20, Rn. 123.

⁷⁶² Das dies der Intention der Lizenzgeber entspricht, räumen auch *Hilber/Reintzsch*, CR 2014, 697, 701, ein.

⁷⁶³ Zu den Anforderungen an eine ergänzende Vertragsauslegung vgl. BGH, NJW 2012, 844 m. w. N.

⁷⁶⁴ Zu der Frage, ob durch Open Source Lizenzen überhaupt die Nutzung im Wege des SaaS gestattet wird, wenn das Rechte nicht explizit aufgeführt wird, vgl. Rn. 196.

⁷⁶⁵ In Anlehnung an *Raymond*, The cathedral and the bazaar, <http://www.catb.org/~esr/writings/cathedral-bazaar/>.

⁷⁶⁶ Prominente Beispiele sind Mozilla und Open/Libre Office, welche erst nach ihrer Fertigstellung und einer jahrelangen proprietären Nutzung als Netscape Navigator bzw. Star Office unter Open Source Bedingungen freigegeben wurden.

⁷⁶⁷ Die im Auftrag der Europäischen Gemeinschaft erstellte FLOSS-Studie aus dem Jahr 2002 geht davon aus, dass der Großteil der freien Softwareprojekte von kleinen Programmiergruppen betrieben wird. Die Studie berichtet jedoch auch von sieben Projekten mit mehr als 500 Entwicklern, 28 Projekten mit 101 bis 500 Entwicklern, 109 Projekten mit 51 bis 100 Entwicklern und 447 Projekten mit 21 bis 50 Entwicklern, vgl. Grafik 6 aus Part V, *Ghosh/Robles/Glott*, Free/Libre and Open Source Software: Survey and Study, Part V, http://www.flossproject.org/report/Final5all.htm#_ftnref12.

und zusammenarbeiten.⁷⁶⁸ Der Betriebssystemkernel Linux ist hier das beste Beispiel; zu diesem Projekt haben ca. 15.000 Entwickler beigetragen.⁷⁶⁹ Arbeiten mehrerer Programmierer im Rahmen eines solchen **Open Source Entwicklungsmodells** zusammen, sind unterschiedliche urheberrechtliche Konstellationen denkbar.⁷⁷⁰

- 201** Selbstständige Werke, also Programme oder Programmteile, die unabhängig voneinander geschrieben wurden und unabhängig voneinander verwertbar sind, können gem. § 9 UrhG miteinander verbunden werden (**verbundenes Werk**). Ein typisches Beispiel dürfte die Verbindung einer Standard-Programmbibliothek mit einer Anwendung sein.⁷⁷¹ Jeder bleibt dann Urheber des von ihm geschaffenen Werkes.⁷⁷²
- 202** Programmierer können Software aber auch als **Miturheber** gem. § 8 UrhG entwickeln, wenn sie im Rahmen eines gemeinsamen Projekts zusammenwirken und dabei die Entwicklung nach einer **Gesamtidee** erfolgt.⁷⁷³ Dies kann auch örtlich getrennt über Netzwerke geschehen, auch wird man nicht verlangen müssen, dass alle Miturheber alle anderen Miturheber kennen.⁷⁷⁴ Miturheberschaft kann ausnahmsweise auch dann vorliegen, wenn die Beiträge zeitlich nacheinander erbracht werden, solange dies im Rahmen eines einheitlichen Entwicklungsprozesses und bei gegenseitiger Abstimmung geschieht.⁷⁷⁵ Die Miturheber bilden – zumindest⁷⁷⁶ – eine Gesamthandgemeinschaft, grundsätzlich müssen sie stets gemeinsam über die Verwertung entscheiden, also auch darüber, ob das von ihnen erstellte Programm unter eine Open Source Lizenz gestellt wird.⁷⁷⁷
- 203** Schließlich kann ein Werk aus der **Bearbeitung** eines vorbestehenden Werkes entstehen, vgl. §§ 3, 23, 69c Nr. 2 UrhG. In dieser Konstellation sind sowohl das vorbestehende als auch das abgeleitete Programm selbstständig durch das Urheberrecht geschützt. Zur Veröffentlichung und Weiterverbreitung bedarf der Autor des abgeleiteten Werkes der Gestattung des Urhebers des vorbestehenden Werkes. Die Abgrenzung zur Miturheberschaft kann mitunter schwierig sein und dürfte sich in erster Linie danach richten, ob die einzelnen Beiträge im Rahmen einer Gesamtkonzeption erbracht werden oder ob sie ohne Abstimmung mit den Urhebern des

⁷⁶⁸ Zur Entstehungsgeschichte von GNU/Linux s. o. Rn. 15.

⁷⁶⁹ Vgl. 2017 Linux Development Report, S. 11, https://go.pardot.com/l/6342/2017-10-24/3xr3f2/6342/188781/Publication_LinuxKernelReport_2017.pdf.

⁷⁷⁰ Vgl. zu den folgenden Überlegungen auch Koch, CR 2000, 273, 277; Koglin, S. 73 ff.; Meyer, passim, mit instruktiven Ausführungen zur Lage in den USA; Schäfer, S. 21 ff.; Teupen, S. 151 ff.

⁷⁷¹ Schäfer, S. 37.

⁷⁷² Koch, CR 2000, 273, 278, geht davon aus, dass die Zusammenfügung des GNU-Systems mit dem Linux-Kernel eine Werkverbindung i. S. d. § 9 UrhG darstellt. Die dafür erforderliche „gesonderte Verwertbarkeit“ des Kerns dürfte in der Tat gegeben sein, wie z. B. einige Embedded-Systeme zeigen, die ohne GNU-Komponenten auskommen, oder auch das Android-System, vgl. Stallman, Android and Users Freedom, <https://www.gnu.org/philosophy/android-and-users-freedom.html>.

⁷⁷³ Zur Gemeinschaftlichkeit der Werkschöpfung bei der Miturheberschaft allgemein vgl. Schrickler/Loewenheim-Loewenheim/Pfeifer, § 8, Rn. 8; eingehend zur Situation bei Open Source Spindler, FS Schrickler zum 70. Geburtstag, S. 539 ff.; Spindler-Spindler, S. 27 ff. sowie Schäfer, S. 25 ff.; Meyer, S. 11 ff. und v. Welsler, ZGE 2017, 570, 580.

⁷⁷⁴ Im Anschluss an Spindler, FS Schrickler zum 70. Geburtstag, S. 539 ff. (bei Fn. 20); ebenso im Ergebnis Plaß, GRUR 2002, 670, 672.

⁷⁷⁵ Hierauf weist zu Recht Marly, Rn. 135, hin; vgl. auch Teupen, S. 158 f. und v. Welsler, ZGE 2017, 570, 580, für den Linux-Kernel. Im Regelfall ist dann aber von einer Bearbeitung auszugehen, vgl. BGH, GRUR 2005, 860, 863 – *Fash 2000*.

⁷⁷⁶ Zur Frage, ob und wenn ja, unter welchen Voraussetzungen von einer Gesellschaft bürgerlichen Rechts auszugehen ist s. u. Rn. 271 ff.

⁷⁷⁷ Vgl. Schrickler/Loewenheim-Loewenheim/Pfeifer § 8, Rn. 10 ff.

ursprünglichen Programms erstellt werden. Spaltet sich eine Entwicklergruppe vom bisherigen Projekt ab, um eine unabhängige Fortentwicklung vorzunehmen (sog. „fork“), so wird das Ergebnis des Folgeprojekts als Bearbeitung einzuordnen sein. Stoßen dagegen neue Teilnehmer zu einem bereits angelaufenen Projekt dazu und suchen die Abstimmung mit den bisherigen Teilnehmern, so dürften sie als Miturheber einzuordnen sein, auch wenn ihre Beiträge für das Gesamtprojekt gegebenenfalls zunächst nicht besonders ins Gewicht fallen.

Bei freien Programmen komplexerer Art, wie z. B. dem Linux-Kernel, finden sich alle genannten Formen der Zusammenarbeit in **Kombination**. Bearbeitungen und in Miturheberschaft entstandene Programme werden mit anderen Werken verbunden oder in Miturheberschaft geschriebene Programme werden bearbeitet.⁷⁷⁸

II. Urheber in Arbeits- und Dienstverhältnissen

Entsprechend den allgemeinen urheberrechtlichen Vorschriften kommt dem Programmierer als Urheber das Urheberrecht an der Software mit deren Schöpfung zu. § 69b UrhG gewährt dem Arbeitgeber oder Dienstherren eines Programmierers aber eine **gesetzliche Lizenz**, nach der dem Arbeitgeber die ausschließlichen Nutzungsrechte ohne ausdrückliche Übertragung zukommen. Die gesetzliche Formulierung „zur Ausübung aller vermögensrechtlichen Befugnisse“ soll sicherstellen, dass die Rechte des Arbeitgebers umfassend sind, sich also aus der Zweckübertragungslehre keine Beschränkungen ergeben und eine Bearbeitung möglich ist.⁷⁷⁹ Liegen die vermögensrechtlichen Befugnisse beim Arbeitgeber, so kann dieser das Werk nach den Bestimmungen einer Open Source Lizenz lizenzieren, und zwar auch **ohne Einwilligung des Arbeitnehmers**.⁷⁸⁰ § 69b UrhG verschafft dem Arbeitgeber alle Rechte, die dieser benötigt, um auch eine solche weitgehende Einräumung von Nutzungsrechten an Dritte vorzunehmen. Theoretisch denkbar bleiben allenfalls urheberpersönlichkeitsrechtliche Ansprüche des Urhebers gegen eine Veränderung des Programms, dies jedoch nur im Rahmen der oben genannten Grenzen für entsprechende Rechte im Bereich von Software.⁷⁸¹

Verschiedentlich wurde angezweifelt, ob angestellte Programmierer Software weiterentwickeln dürfen, die unter der GPL oder Copyleft-Lizenzen im Allgemeinen lizenziert ist.⁷⁸² Nach dieser Auffassung soll hierin ein Verstoß des Arbeitnehmers gegen das Verbot von Lizenzgebühren vorliegen, da das Arbeitsentgelt auch für die Rechtseinräumung gezahlt werde. Dieser Ansicht kann nicht beigetreten werden und zwar unabhängig von der Frage, ob man das **Arbeitsentgelt** als Gegenleistung für die Rechtseinräumung ansieht oder ob man den Charakter als gesetzliche Lizenz betont und das Arbeitsentgelt als Entgelt für die Arbeitsleistung einordnet.⁷⁸³ Entscheidend ist allein, ob das **Lizenzgebührenverbot** so zu verstehen ist, dass die in Frage stehende Lizenz eine Weiterentwicklung durch Arbeitnehmer untersagen möchte. Die Antwort hat also nicht bei § 69b UrhG, sondern bei der Vertragsauslegung anzusetzen. Für ein so weitgehendes Verbot fehlt es jedoch an An-

⁷⁷⁸ Zu den prozessualen Folgen, die sich daraus ergeben, s. u. Rn. 226 ff.

⁷⁷⁹ Schrickel/Loewenheim-Loewenheim/Spindler, § 69b, Rn. 12.

⁷⁸⁰ So auch Teupen, S. 165. Eine Einwilligung kann aber wegen der §§ 32, 32a UrhG sinnvoll sein, s. o. Rn. 186.

⁷⁸¹ Vgl. oben Rn. 177 f.

⁷⁸² Siehe Deike, CR 2003, 9, 17; Koch, Informatik Spektrum 2004, 55, 58; Spindler in: Büllesbach/Dreier, S. 115, 124.

⁷⁸³ So insb. Koch, CR 2000, 333, 341; wie hier Meyer, S. 29.

haltungspunkten im Lizenztext der GPL, aber auch der anderen Lizenzen.⁷⁸⁴ Im Gegenteil: Eine solche Interpretation stünde im Widerspruch zu dem eindeutigen Zweck der Lizenzen, einen möglichst weitgehenden Austausch von Softwarebestandteilen zu erreichen. Der scheinbare Konflikt ist deswegen durch eine an dem Zweck der Lizenzen ausgerichtete Auslegung aufzulösen: Die Entlohnung des Arbeitnehmers stellt keine Lizenzgebühr im Sinne von Open Source Lizenzen, insbesondere der GPL dar.

- 206 Der gesetzliche Rechtsübergang des § 69b UrhG gilt nicht für **freiberufliche Programmierer** („Freelancer“), da Auftragswerke, anders noch als im ursprünglichen Entwurf der Kommission, nicht in die Regelung aufgenommen wurden.⁷⁸⁵ Dies ist gerade im Open Source Bereich von Bedeutung, da dort viele Programmierer freiberuflich arbeiten und die Abgrenzung der Programmerstellung im betrieblichen und privaten Bereich dann keine Rolle spielt. Bei angestellten Entwicklern muss dagegen im Einzelfall geprüft werden, ob die Software von dem Programmator „in Wahrnehmung seiner Aufgaben“ erstellt wurde und damit der gesetzlichen Lizenz unterliegt oder ob dies nicht der Fall ist und damit eine Freigabe unter einer Open Source Lizenz, z. B. im Rahmen von **Contributions** zu Open Source Projekten, nur durch den Programmierer selbst erfolgen kann.⁷⁸⁶

III. Rechtswahrnehmung durch Projekte und Organisationen („contributor agreements“)

- 207 Schließlich kann die Inhaberschaft der (ausschließlichen) Nutzungsrechte auch aufgrund vertraglicher Vereinbarung auf ein Projekt, ein Unternehmen oder eine Organisation übergehen, die das Projekt leitet oder organisatorisch unterstützt. Zum Teil verlangen freie Softwareprojekte oder Unternehmen, dass die Teilnehmer vor der Übernahme eines Beitrags in den „offiziellen“ Release ein sog. „*copyright assignment*“, „*contributor agreement*“ oder „*contributor license agreement*“ zugunsten des Projekts unterzeichnen, zum Teil nehmen Dachorganisation die Rechte von verschiedenen Projekten treuhänderisch wahr.⁷⁸⁷

Ein solcher Ansatz wird insbesondere von der FSF und der FSF Europe verfolgt. So verlangt die Free Software Foundation von Programmierern, die sich an Softwareprojekten der Free Software Foundation beteiligen, die Abgabe einer entsprechenden Rechtseinräumung.⁷⁸⁸ Diese sieht die Einräumung des ausschließlichen Rechts an die Stiftung vor. Die FSF verpflichtet sich ihrerseits, dem bisherigen Rechtsinhaber eine unbegrenzte Anzahl einfacher Nutzungsrechte zurück zu übertragen. Auch muss die Software nach den Bedingungen einer freien Soft-

⁷⁸⁴ Die Mozilla Public License geht in Ziffer 1.1 ausdrücklich davon aus, dass als „Contributor“ auch juristische Personen in Frage kommen.

⁷⁸⁵ Dreier, CR 1991, 577, 579.

⁷⁸⁶ Vgl. zur Auslegung des Merkmals „in Wahrnehmung seiner Aufgaben“ OLG München, CR 2000, 429, 430; OLG Köln, CR 2005, 557, zur Entwicklung außerhalb der Arbeitszeit.

⁷⁸⁷ Siehe zu den unterschiedlichen Strategien von FOSS-Projekten eingehend Jakob, 5 JIPITEC 105 (2014); Maracke, 10 SCRIPTed 140 (2013); Metzger, 10 SCRIPTed 177 (2013); zu den verschiedenen rechtlichen Gestaltungen siehe Engelhardt, 10 SCRIPTed 149 (2013); zum anwendbaren Recht siehe Metzger, 10 SCRIPTed 177 (2013). Für den weiteren Zusammenhang der Governance freier Projekte und der Rolle entsprechender Assignments siehe Wielsch, 1 JIPITEC 96 (2010) para. 32–34.

⁷⁸⁸ Der Text des „Assignments“ findet sich unter <https://web.archive.org/web/20030131061457/http://ftp.xemacs.org/old-beta/FSF/assign.changes>, vgl. auch <http://www.gnu.org/licenses/why-assign.en.html>.