# Open-Source-Projekte: vom Nischenphänomen zum integralen Bestandteil der Softwareindustrie

„Open" ist zu einem ubiquitären Beiwort der digitalen Moderne geworden – von „Open Science" über „Open Innovation" bis hin zu „Open Government". Ein wesentlicher Ausgangspunkt für die Popularität des Offenheitsparadigmas liegt in dem Bedeutungszuwachs von Open-Source-Projekten in der Softwareentwicklung, der vielfältige Hoffnungen auf dezentralere Koordinationsweisen sowie eine technikinduzierte Auflösung eingespielter Rollenverteilungen und Machtasymmetrien in der Arbeitswelt provoziert hat. Aber eignen sich aktuelle Open-Source-Projekte überhaupt noch als Beispiel für offengehaltene und egalitäre Produktionszusammenhänge?

JAN-FELIX SCHRAPE

## 1. Einleitung: Das Versprechen der neuen Offenheit

Quelloffener Software kann inzwischen in vielen Segmenten des IT-Marktes eine hohe Relevanz zugeschrieben werden: Der Apache HTTP Server ist seit den 1990er-Jahren der weltweit meistgenutzte Webserver; linuxbasierte Architekturen dominieren das Feld der Server-Betriebssysteme; der Großteil der im Internet eingesetzten Datenbank- und Content-Management-Systeme steht unter freien Lizenzen (W3techs 2016). Marktforschungsunternehmen diagnostizieren, dass mittlerweile die meisten Unternehmen selbst in geschäftskritischen Bereichen auf quelloffene Elemente zurückgreifen (Miller/Nelson 2016; Driver 2014). Und auch auf dem Feld der Consumer-Software sind Produkte, die vollständig oder teilweise auf Open-Source-Entwicklungsvorhaben basieren, zu einem festen Bestandteil der Alltagwelt vieler Anwender geworden, so etwa der Browser Mozilla Firefox oder das mobile Betriebssystem Android.

Dieser Erfolg von Open-Source-Projekten in der Softwareentwicklung wurde in den Sozialwissenschaften angesichts klassischer Sichtweisen, die „intellectual property rights" als Treiber in Innovationsprozessen ansehen, zunächst mit Erstaunen zur Kenntnis genommen (Lessig 1999) und danach rasch als Beleg für die Emergenz eines neuen Produktionsmodells gedeutet, das auf freiwilliger wie selbstgesteuerter Kollaboration unter Gleichberechtigten beruht, den Stellenwert von Unternehmen in der Arbeitswelt schmälern und eingespielten Formen ökonomischer Koordination wie „Markt" oder „Hierarchie" auf Dauer überlegen sein könnte (Lakhani/Hippel 2003). Insbesondere die durch Yochai Benkler popularisierte Vorstellung der „commons-based peer production" als technisch effektivierte „collaboration among large groups of individuals [...] without relying on either market pricing or managerial hierarchies to coordinate their common enterprise" (Benkler/Nissenbaum 2006, S. 394), die mit „systematic advantages [...] in identifying and allocating human capital/creativity" einhergehen soll (Benkler 2002, S. 381), erfuhr eine intensive Reflexion und wurde zuletzt zunehmend auf angrenzende Kontexte wie die Herstellung materieller Güter oder den Dienstleistungssektor übertragen (z. B. Rifkin 2014).

Gerade in der Beobachtung von Open-Source-Softwareprojekten zeigt sich jedoch inzwischen deutlich, dass sich mit wachsender Größe der Entwicklergemeinschaften regelmäßig prägnante hierarchische Entscheidungsmuster herausbilden, führende IT-Konzerne mit steigender Relevanz der Vorhaben zumeist erheblichen Einfluss erlangen und dauerhaft aktive Projekte nicht durch intrinsisch motivierte Freiwillige – „satisfying psychological needs, pleasure, and a sense of social belonging" (Benkler 2004, S. 1110) – getragen werden, sondern auf den Beiträgen angestellter Entwickler fußen. Im häufig als typisches Beispiel genannten Linux-Kernel-Projekt etwa wurden zuletzt über 80 % ▶

der Aktualisierungen von Programmierern durchgeführt, „who are being paid for their work" (Corbet et al. 2015, S. 11). Angesichts dieser Verschränkungen reichen die oft üblichen pauschalen Verweise auf Open-Source-Communitys als Alternative zur proprietären (d. h. unternehmenseigenen) Entwicklung nicht mehr aus und es stellt sich die Frage, welche dieser Projekte den mit ihnen assoziierten „promises of openness, freedom, and democracy" (Kranich/Schement 2008, S. 563) heute noch gerecht werden können.

Dieser Aufsatz verfolgt daher das Ziel, auf der Basis von aggregierten Marktdaten, Dokumentenauswertungen und Hintergrundgesprächen einen systematisierenden Überblick über Open-Source-Communitys sowie ihre sozioökonomischen Kontexte zu entfalten.[1] Zunächst erfolgt eine Rekonstruktion zur Ausdifferenzierung quelloffener Softwareprojekte und ihren Relationen zu etablierten IT-Unternehmen (Abschnitt 2). Daran anknüpfend werden vier Varianten rezenter Open-Source-Projekte voneinander abgegrenzt – von korporativ geführten Kollaborationsprojekten und elitezentrierten Gemeinschaften über heterarchisch angelegte Infrastrukturvorhaben bis hin zu egalitär ausgerichteten Gruppierungen (3). Im folgenden Abschnitt 4 wird aus technik- bzw. organisationssoziologischer Sicht herausgearbeitet, warum quelloffene Softwareprojekte ihre Formatierung als Gegenentwurf zur kommerziellen Produktion mittlerweile weitgehend verloren haben, aber im Gegensatz zu früheren Spielarten kollektiver Invention überlebensfähig geblieben sind. Das Schlusskapitel (5) zieht Bilanz und diskutiert gesellschaftspolitische Implikationen.

## 2. Rekonstruktion: Ausdifferenzierung quelloffener Softwareprojekte

Kurz nachdem quelloffene Softwareprojekte in den allgemeinen Aufmerksamkeitsbereich gerückt sind, wurde eine Reihe an Abhandlungen veröffentlicht, die erste Erklärungen für deren Erfolg lieferten, ihren subversiven Charakter betonten und den sozialwissenschaftlichen Blick auf „Open Source" bis heute mitprägen (z. B. Weber 2000; Moody 2002). Diese Texte haben sich freilich primär an Narrativen aus der Szene selbst orientiert und mit wenigen Ausnahmen (z.B. Lerner/Tirole 2002) auf eine sozioökonomische Einordnung verzichtet. In der folgenden kontextualisierenden Rekonstruktion zeigt sich indes, dass freie und proprietäre Softwareentwicklung seit jeher ineinandergreifen und das Involvement in Open-Source-Projekte in den letzten 15 Jahren zu einem festen Baustein der Innovationsstrategien aller großen Anbieter avanciert ist.

### 2.1 Free Software als Utopie

Die Herausbildung des „free software movements" in den 1980er Jahren lässt sich als eine direkte Reaktion auf die zuvor angestoßene Kommodifizierung von Software verstehen: Während die ersten digitalen Computer in den 1950er Jahren in enger Kooperation zwischen Herstellern und Anwendern entwickelt und Computerprogramme noch nicht als von der Hardware unabhängige Güter wahrgenommen wurden, sondern „as a research tool to be developed and improved by all users" (Gulley/Lakhani 2010, S. 6), wurde Software Ende der 1960er Jahre durch mehrere kartellrechtliche Verfahren – z. B. gegen die International Business Machines Corporation (IBM), der vorgeworfen wurde, durch das kombinierte Angebot von Hardware und Software Mitbewerber aus dem Rennen werfen zu wollen – und der Gründung erster entsprechend spezialisierter Unternehmen zunehmend als separates Produkt sichtbar.

Für die Entstehung einer eigenständigen Softwarebranche spielte zudem die Verbreitung von Minicomputern eine wichtige Rolle: Zum einen waren sie im Betrieb günstiger als Mainframe-Systeme und nicht mehr auf eine möglichst effiziente Nutzung ausgelegt; zum anderen förderten erweiterte Ein- und Ausgabeschnittstellen (z. B. Bildschirme) die Herausbildung neuer Softwaregenres. Vor allem in nordamerikanischen Universitäten boten campusöffentlich erfahrbare Minicomputer, die oft von deren Herstellern an die Institute gespendet wurden, einen Nährboden für informelle Projektgruppen, die mit ihren Arbeiten die Grundlage für die sich ab 1975 entlang der ersten Heimcomputer herausbildende Amateur-Computing-Szene schufen. Das geteilte Problem der in diesen Kontexten entwickelten Architekturen lag jedoch in ihrer *mangelnden rechtlichen Absicherung*: Sie wurden als gemeinfreie Güter veröffentlicht und waren kaum vor Einzelaneignung geschützt. Das an Universitäten mitentwickelte Betriebssystem Unix etwa wurde durch AT&T ab 1983 – sobald es kartellrechtlich möglich war – kommodifiziert; das 1961 durch Studierende des Massachusetts Institute of Technology (MIT) programmierte Spiel „Spacewar!" wurde zur Basis zahlreicher kommerzieller Spieleautomaten.

Eine Schwierigkeit, die vice versa für gewerbliche Anbieter mit dieser Hobbyisten-Kultur einherging, bestand darin, dass Programme in diesen Kreisen zwar gerne weitergegeben, aber selten käuflich erworben wurden: „Hardware must be paid for, but software is something to share. [...] Who can afford to do professional work for nothing?" (Gates 1976) Infolgedessen wurden Softwareprodukte in den frühen 1980er Jahren meist nur noch als nicht mehr veränderbare Binärdateien ohne Quellcode verkauft. Gleichzeitig erhöhten mehrere Gesetzesnovellen in den USA deren

1 Methodisch basiert der Text neben der Aufarbeitung vorhandener Literatur auf einer systematischen Auswertung von Marktdaten, Branchennachrichten, Publikationen aus den Projekten bzw. Unternehmen sowie Mailinglisten aus den letzten Jahrzehnten. Zudem wurden acht Hintergrundgespräche mit Softwareingenieuren aus der BRD, Schweiz und Kalifornien geführt. Ausführliche Quellenangaben finden sich in Schrape (2016).

Schutz und Ausschließbarkeit. Als gesellschaftsethische Replik auf diese Schließungsprozesse kündigte der MIT-Mitarbeiter Richard Stallman in dem damals noch jungen elektronischen Kommunikationsnetzwerk Usenet an, unter dem rekursiven Akronym GNU (‚GNU's Not Unix') ein unabhängiges Betriebssystem entwickeln zu wollen: „I consider that the golden rule requires that if I like a program I must share it with other people who like it. [...] So that I can continue to use computers without violating my principles, I have decided to put together a sufficient body of free software [...]" (Stallman 1983).

Obgleich GNU als eigenständiges Betriebssystem bis heute nicht für den alltäglichen Einsatz geeignet ist, erwies sich Stallmans Projekt als Keimzelle für die freie Softwareentwicklung: 1985 gründete sich in seinem Kontext die Free Software Foundation; ab 1988 wurden erste industrielle Großspender wie Sony oder Hewlett-Packard angeworben. Die bedeutsamste Neuerung bestand aber in der Definition *rechtlich belastbarerer Lizenzmodelle*, die wie die 1989 publizierte General Public License (GPL) erzwingen, dass auch Derivate freier Software stets frei bleiben müssen: „Each time you redistribute the Program [...], the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions" (FSF 1989). Ab 2001 waren Verstöße gegen die GPL Gegenstand mehrerer Gerichtsverfahren gegen Firmen wie Skype oder D-Link (Jaeger 2010), wobei „the court of public opinion" im Usenet bzw. Web für die Etablierung der in der GPL angelegten Reziprozitätsprinzipien eine ebenso tragende Rolle gespielt hat (O'Mahony 2003, S. 1189).

Der Erfolg des GNU-Projektes an sich blieb aufgrund seines Zuschnitts auf kostenintensive Workstations und seiner ideologischen Konnotationen allerdings zunächst begrenzt. Auf beide Problemstellungen bot das Linux-Kernel-Projekt eine Antwort: Linux wurde 1991 durch den Studenten Linus Torvalds als freier Betriebssystemkern für die günstigeren Mikrocomputer vorgestellt und war daher für eine größere Zahl an Entwicklern attraktiv. Zudem zeichnete sich Torvalds von vornherein durch eine liberalere Haltung als die Free Software Foundation aus: „This world would be a much better place if people had less ideology and a whole lot more 'I do this because it's fun [...]'" (Torvalds 2002). Ein weiterer Grund für das Florieren der Linux-Kernel-Entwicklung bestand in der raschen Verbreitung des World Wide Web ab 1993, das durch seine kommunikationseffektivierenden Eigenschaften sowohl den Zugriff auf als auch die Beteiligung an dem Projekt und dessen Koordination erleichtert hat. Nichtsdestotrotz blieb auch der Linux Kernel in den ersten Jahren ein lediglich in Expertenkreisen bekanntes Vorhaben.

Dies änderte sich mit dem vielrezipierten Buch „The Cathedral and the Bazaar" (Raymond 1999), das von dem Softwareentwickler Eric S. Raymond 1997 zunächst als Essay vorgestellt wurde. Seine Kernthese lautete: Während in traditionellen Produktionsmodellen der Quellcode eines Programms nur für fertige Versionen veröffentlicht wird und die Entwicklergruppen hierarchisch organisiert sind (*cathedral*), sei der Source Code in Projekten wie Linux oder dem von Raymond initiierten Fetchmail stets einsehbar, ihre Gruppen seien horizontal strukturiert und geprägt durch modulare Selbstorganisation ohne zentrales Management (*bazaar*). Kritische Beobachter stellten jedoch früh fest, dass in beiden Fällen zwar viele Vorschläge aus der Community kamen, die letztlichen Änderungen aber nur durch eine Person – Torvalds oder Raymond – freigegeben wurden (Bezroukov 1999). Anders formuliert: „The only entity that can really succeed in developing Linux is the entity that is trusted to do the right thing. And as it stands right now, I'm the only person/entity that has that degree of trust" (Torvalds 1998, S. 36).

Mit GNU und Linux entstanden in den 1980/90er Jahren also zwei Flaggschiffprojekte freier Software, deren Erfolg durch die elektronische Effektivierung der Kommunikation erheblich befördert wurde. In ihrem Kontext bildeten sich rechtliche Instrumente wie die General Public License (GPL), welche die kollektiven Arbeitsresultate vor Einzelaneignung schützen, sowie informelle Arbeitskonventionen heraus, deren Anerkennung sich durch die erhöhte Sichtbarkeit der Kommunikation im Netz unmittelbarer überprüfen ließ als zuvor. Daneben verfestigten sich erste anschlussfähige Narrative, die freie Softwareentwicklung als revolutionären Produktionsmodus ohne Machtasymmetrien beschrieben und zeitweilig ohne weitere Rückfragen sozialwissenschaftlich weiterverarbeitet wurden (z. B. Benkler 2002; Tapscott/Williams 2006).

## 2.2 Open Source als Methode

Im nachfolgenden Jahrzehnt konnte sich die quelloffene Entwicklung als Arbeitsmethode zunehmend in der Softwarebranche durchsetzen, was sich neben der fortgesetzten Verbreitung des Internets vorwiegend auf drei Dynamiken zurückführen lässt:

*Zum ersten* lagerte eine wachsende Zahl an IT-Firmen die Entwicklung von Softwareprodukten in den quelloffenen Bereich aus, darunter Netscape Communications als ein besonders früher und aufsehenerregender Fall: Nachdem es absehbar war, dass Microsoft den Netscape Navigator durch den in Windows integrierten Internet Explorer aus dem Markt drängen würde, kündigte das Unternehmen Anfang 1998 an, große Teile des Codes seines Browsers in das *quelloffene Projekt Mozilla* zu überführen, das bis zur Gründung der Mozilla Foundation 2003 durch AOL/Netscape unterstützt wurde und aus dem 2004 der Webbrowser Firefox hervorging. Dabei ging es Netscape in erster Linie um die Erschließung neuer Kundenkreise „by [...] building a community that addresses markets and needs we can't address on our own [...]" (Netscape Communications 1998).

*Zum zweiten* kam 1998 eine Gruppe um Eric Raymond zu dem Schluss, dass sich der politisch belegte Begriff ▶

TABELLE 1

**Global meistgenutzte quelloffene Softwarelizenzen**

| | z. B. genutzt von | 2016 (in %) | 2010 (in %) | Ausrichtung | Publikation |
|---|---|---|---|---|---|
| GNU Public License 2.0 | Linux-Kernel, WordPress | 21 | 47 | strongly protective | 1991 |
| MIT License | jQuery, Ruby on Rails | 26 | 6 | permissive | 1988 |
| Apache License 2.0 | Android, Apache HTTP | 16 | 4 | permissive | 2004 |
| GNU Public License 3.0 | GNU | 9 | 6 | strongly protective | 2007 |
| BSD License 2.0 (3-clause) | Chromium, WebKit | 6 | 6 | permissive | 1999 |
| Artistic License 1 / 2 | Perl | 4 | 9 | permissive | 2000/2006 |
| GNU Lesser GPL 2.1 / 3.0 | VLC Media Player | 6 | 9 | weakly protective | 1999/2007 |
| Microsoft Public License | Microsoft Azure | 2 | 2 | permissive | 2007 |
| Eclipse Public License | Eclipse | 2 | 1 | permissive | 2004 |

Quelle: Black Duck Knowledgebase (Stand: 7/2016).

WSI Mitteilungen

„Free Software“ für die Verbreitung quelloffener Software in kommerziellen Kontexten als hinderlich erweisen könnte. Diese Gruppe schuf das *neue Label „Open Source“*, das die Überlegenheit des Entwicklungsmodells betonen sowie gesellschaftsethische Aspekte ausblenden sollte, und gründete mit Hilfe von Szeneprotagonisten wie Tim O’Reilly, der später auch den Begriff „Web 2.0“ prägen sollte, die Open Source Initiative. Allerdings unterstützt die Free Software Foundation diese Kursänderung bis heute nicht: „For the Open Source movement, non-free software is a suboptimal solution. For the Free Software movement, non-free software is a social problem and free software is the solution“ (Stallman 2002, S. 57).

Zu den Vermarktungsbemühungen der Open Source Initiative, welche die unternehmerischen Vorteile quelloffener Software betonten, kamen *zum dritten* die durch den allgemeinen Dotcom-Boom beförderten *Börsenerfolge einiger „open source companies“* im Jahr 1999 hinzu, darunter die Hardwareanbieter VA Linux und Cobalt Networks sowie der Linux-Distributor Red Hat. Deren Börsengänge gehörten zu den erfolgreichsten Debüts aller Zeiten und erregten eine entsprechend große mediale Aufmerksamkeit. Kurz darauf beschrieb etwa der Spiegel (33/1999, S. 78) Linux als „ernsthafte Konkurrenz zum Microsoft-Monopol“. Damit war „Open Source“ als Schlagwort im öffentlichen Bewusstsein angekommen.

Diese ineinandergreifenden Entwicklungsstränge führten im Verbund mit der weiteren Ausweitung des IT-Marktes zu einem raschen Wachstum freier Softwareprojekte: Während 1999 einige hundert quelloffene Vorhaben existierten, sind heute auf Plattformen wie GitHub und SourceForge mehrere Millionen Projekte zu finden. Angesichts dieser steigenden Anzahl und der Definition eigener Lizenzmodelle durch Firmen und Stiftungen unterlag die Open-Source-Entwicklung einer starken Diversifizierung (*Tabelle 1*): Neben originäre „Copyleft“-Lizenzen, die garantieren, dass auch Derivate freier Software stets unter gleichen Bedingungen distribuiert werden (*strongly protective*), sind Lizenzen getreten, welche die Einbindung freier Software in proprietäre Produkte gestatten, sofern eben diese Elemente quelloffen bleiben (*weakly protective*), oder wieder die Publikation von Fortentwicklungen unter restriktiveren Bedingungen erlauben (*permissive*). Diese Vielfalt erweitert die strategischen Optionen für kommerzielle Stakeholder: Google etwa entschied sich von vornherein dazu, Android unter permissive Lizenz stellen.

Daneben lässt sich in zweierlei Hinsicht eine „Korporatisierung“ von Open-Source-Projekten beobachten: Zum einen werden zentrale Vorhaben wie der Linux Kernel, der Apache HTTP Server und die Cloud-Computing-Architektur OpenStack heute vorrangig durch Spenden von Unternehmen finanziert oder operieren wie die Browser-Engine WebKit (Apple) und Android (Google) unter der expliziten Ägide kommerzieller Anbieter. Zum anderen speist sich die Entwicklerbasis großer Projekte zunehmend aus Firmenkontexten: Kolassa et al. (2014) kommen für den Linux Kernel sowie 5000 weitere marktrelevante Vorhaben zu dem Schluss, dass 2000 bis 2011 über 50 % aller Beiträge in der westlichen Kernarbeitszeit geleistet wurden; die Linux Foundation (Corbet et al. 2015) beobachtet, dass der Anteil unabhängiger Programmierer an der Kernel-Entwicklung (2009: 18 %; 2014: 12 %) gegenüber unternehmensaffiliierten Beiträgern (z. B. von IBM, Samsung oder Intel) kontinuierlich abnimmt.

In den letzten zwei Jahrzehnten konnte sich die Open-Source-Entwicklung insofern zunehmend in der Softwarebranche etablieren; sie hat dabei allerdings ihre Formatierung als Gegenentwurf zur proprietären Herstellung

weitgehend verloren.[2] Zwar lassen sich nach wie vor kleinere Vorhaben wie die Linux-Varianten Arch oder Parabola finden, die sich an den genuinen Maximen freier Software ausrichten. In viele relevante Open-Source-Projekte sind inzwischen indes etablierte IT-Unternehmen involviert, die diese Arbeitskontexte nutzen, um für sie förderliche Standards zu protegieren und ihre sonst eher abgeschotteten Entwicklungsaktivitäten durch „kontrollierte Öffnungen an den Rändern" (Dolata 2015, S. 17) zu erweitern. Insofern beschreibt der Blogger Mike Bulajewski (2011) das Bild von Open-Source-Projekten als Gemeinschaften „of volunteer programmers collaborating together in a gift economy" zurecht als Illusion.

## 2.3 Open Source als Innovationsstrategie

Insbesondere für das Segment der Unternehmensinformatik, in dem über 80 % der globalen Softwareumsätze generiert werden, lässt sich inzwischen „a widespread use of open-source technology" diagnostizieren (Driver 2014). Zudem kann Open-Source-Lösungen im Bereich der basalen informationstechnischen Infrastrukturen Marktführerschaft zugesprochen werden. Den Grund dafür sehen Marktforscher nicht nur in Kostenvorteilen, sondern auch in deren Anpassbarkeit und „inherent trialability" (Spinellis/Giannikas 2012, S. 667). Es verwundert daher nicht, dass heute die meisten großen IT-Konzerne in Open-Source-Projekte involviert sind.

*Microsoft* – das Unternehmen, das Open Source lange als „intellectual-property destroyer" bezeichnete (Computerworld 3/2001, S. 78) – hat 2012 die Tochterfirma MS Open Technologies lanciert und seitdem das Framework .NET, Software-Development-Kits für den Cloud-Computing-Dienst Azure sowie viele weitere Komponenten unter freie Lizenz gestellt „to achieve a strategic objective, such as promoting industry standards, advancing interoperability, or attracting and enabling our external development community" (Microsoft 2015, S. 13). Welche genauen Anteile ihrer Entwicklungsausgaben marktführende Konzerne wie Microsoft in Open-Source-Projekte investieren, lässt sich freilich kaum gesondert abschätzen, da quelloffene Elemente mittlerweile für zahlreiche herstellerspezifische Architekturen von Bedeutung sind. *Apples* Betriebssystempakete macOS und iOS etwa basieren auf dem freien unixoiden Betriebssystemkernel Darwin und tragen hunderte weitere Open-Source-Komponenten in sich.

*IBM* investierte bereits zur Jahrtausendwende mehrere 100 Mio. US$ in die Linux-Entwicklung, um Microsofts Dominanz im Enterprise-Bereich entgegenzusteuern und ein Servicegeschäft um quelloffene Software aufzubauen. Heute ist IBM in weit über 100 Open-Source-Projekte involviert, darunter die Cloud-Computing-Plattform OpenStack, an welcher auch *Intel* und *Hewlett-Packard* beteiligt sind. Ihr Involvement resultiert jedoch nicht aus Idealismus, sondern aus Kalkül: „Such actions are comparable to giving away the razor (the code) to sell more razor blades (the related consulting services [...])" (Lerner 2012, S. 43). Aus ähnlichen Gründen beteiligen sich *SAP*, *Oracle* und *Adobe* an quelloffenen Projekten. Insbesondere für kleinere Anbieter dient ein Involvement in wichtige Open-Source-Projekte darüber hinaus als „marketing tool to increase brand recognition" (Dahlander/Magnusson 2008, S. 638).

Eine spezielle Variante korporativen Open-Source-Engagements stellt die Entwicklung des mobilen Betriebssystems Android durch die Open Handset Alliance dar: Beworben als lupenreines Open-Source-Projekt, wird das Vorhaben de facto allein durch *Google* kontrolliert: Android-eigener Code steht unter permissiven Lizenzen, die Google in Kombination mit weiteren Rahmungen wie der „Compatibility Definition" umfassende Steuerungsmöglichkeiten einräumen. „Because it fully controls the development of the OS, Google can determine the technological specifications to which Android partners must abide." (Spreeuwenberg/Poell 2012) Mit der Lancierung von Android ging es Google mit Erfolg primär darum, den nahtlosen Zugriff auf eigene Dienste auf möglichst vielen Geräten zu ermöglichen: Während Google 2007 ca. 99 % seines Umsatzes mit Werbung generierte, war der Verkauf digitaler Inhalte und Services 2015 für ca. 10 % des Umsatzes (US$ 75 Mrd. ) verantwortlich (Alphabet 2016).

Überdies bildete sich Ende der 1990er Jahre eine Reihe an „open source companies" heraus, die ihr Kernprodukt – den Softwarecode – kostenfrei abgeben und mit Supportleistungen ein Geschäft aufzubauen suchten. Mit Ausnahme des Linux-Distributors *Red Hat*, der früh mit führenden Hardwareanbietern kooperiert hat, sind die meisten dieser im Fahrwasser des New-Economy-Hypes lancierten Firmen allerdings schnell wieder eingegangen. Zwar sind im Open-Source-Umfeld zuletzt erneut Startups entstanden (z. B. Hortonworks); in ihrer Außendarstellung verzichten diese Firmen aber meist auf „Open Source" als primäres Differenzierungsmerkmal und zeichnen sich durch eine geringe Verbundenheit mit Stallmans Reziprozitätsidealen aus (Bergquist et al. 2012). Vice versa sind es heute vor allem Konzerne wie IBM oder Microsoft, die in ihrer Öffentlichkeitsarbeit auf ausgewählte Maximen freier Software verweisen.

Viele marktrelevante Open-Source-Communitys stehen folglich in einem engen finanziellen Wechselverhältnis mit führenden IT-Konzernen, die im Rahmen ihrer übergreifenden Innovationsstrategien gezielt in ebensolche Entwicklungsvorhaben investieren. Kombiniert mit ihrem Involvement in die konkrete Code-Entwicklung sichern sich ▶

2 Immer mehr Firmen adaptieren unter dem Label „inner sourcing" (O'Reilly 2000) zudem die in Open-Source-Communitys üblichen Arbeitsweisen für die interne Entwicklung. Modulare und agile Methoden wurden jedoch unabhängig davon bereits Anfang der 1990er Jahre in der IT-Branche angewendet (Martin 1991).

TABELLE 2

**Populäre Projekte auf Open Hub (Webkatalog für Open-Source-Projekte), 2016**

| Projekt | Commits* | Dachorganisation | Primäre Finanzierungsquelle |
|---|---|---|---|
| Android | 79.137 | Google, Open Handset Alliance (90+ Unternehmen) | |
| Linux Kernel | 66.926 | Linux Foundation | Mitglieder (u. a. HP, Intel, IBM) |
| Mozilla Firefox | 54.524 | Mozilla Foundation | Spenden, Royaltys (bis 2014: 90% Google) |
| OpenStack | 53.128 | OpenStack Foundation | Mitglieder (u.a. HP, IBM, Red Hat) |
| KDE | 29.472 | KDE e.V. | Patronagen (u.a. Google, SUSE, Qt) |
| Debian Linux | 26.782 | Debian Project | Spenden, Partner (u. a. HP, 1&1, Simtec) |
| LibreOffice | 17.045 | Document Foundation | Spenden (u.a. Google, Red Hat, Intel) |
| GNU CC | 7.751 | Free Software Found. | Mitglieder, Patronagen (u. a. Google, IBM) |
| Apache HTTP Server | 2.774 | Apache Foundation | Spenden (u. a. Google, Microsoft, Facebook) |

*Aktualisierungen 5/2015–5/2016.

Quelle: Open Hub (Stand: 7/2016), Jahresberichte.

WSI Mitteilungen

die jeweiligen Firmen so einen nicht zu unterschätzenden Einfluss auf marktrelevante Projekte und tragen zugleich zu einer Erhöhung der finanziellen wie personellen Planungssicherheit in den Communitys bei.

## 3. Typologie: Varianten quelloffener Softwareprojekte

In den letzten 15 Jahren ist Open-Source-Software mithin zu einem integralen Bestandteil der IT-Branche geworden. Vor diesem Hintergrund hat sich ein breites Spektrum an unterschiedlich ausgerichteten Open-Source-Projekten herausgebildet (eingehender: Schrape 2016, S. 49ff.). Dabei lassen sich entlang ihrer vorherrschenden Koordinationsweisen und dem Grad ihrer Unternehmensnähe gegenwärtig vier typische Varianten unterscheiden (*Tabelle 3*).

*Korporativ geführte Kollaborationsprojekte* zeichnen sich durch prägnante Hierarchisierungen auf Arbeitsebene aus und erarbeiten häufig marktzentrale Produkte. Ihre Communitys bestehen primär aus in den beteiligten Unternehmen angestellten Programmierern. In Android und WebKit liegt die strategische Kontrolle eindeutig bei Google und Apple; im Cloud-Computing-Projekt OpenStack haben große Sponsoren ebenfalls einen hohen steuernden Einfluss. Eine solche Kollaboration in Open-Source-Projekten trägt

TABELLE 3

**Idealtypische Ausprägungen von Open-Source-Projekten**

| | **Korporativ geführte Kollaborationsprojekte**<br>**z.B. Android, WebKit, OpenStack** | **Elitezentrierte Projektgemeinschaften**<br>**z.B. Linux Kernel, Debian, Firefox** | **Heterarchisch angelegte Infrastrukturvorhaben**<br>**z.B. Apache HTTP, Eclipse, Joomla!** | **Egalitär ausgerichtete Peer Production Communitys**<br>**z.B. GNU CC, Arch Linux, KDE** |
|---|---|---|---|---|
| Arbeitsorganisation | hierarchisch | hierarchisch | horizontal – meritokratisch | horizontal – egalitär |
| strategische Führung | Einzelunternehmen/ Firmenkonsortium | Projektgründer/ Projektleitung | Stiftungsvorstand/ Steuerungsgruppe | Steuerungskomitee/ Kernteam |
| Finanzierung | beteiligte Unternehmen | korporative Spenden/ private Kleinspenden | vorrangig Zuwendungen von Unternehmen | vorrangig private Kleinspenden |
| Teilnehmerbasis | Mitarbeiter aus den beteiligten Unternehmen | angestellte und wenige freiwillige Entwickler | angestellte Entwickler und explizite Unternehmensvertreter | vorrangig freiwillige Entwickler |

Quelle: Zusammenstellung des Autors.

WSI Mitteilungen

zur Überwindung zweier klassischer „knowledge sharing dilemmas" bei: Zum einen verhindern quelloffene Lizenzen die Einzelaneignung des kollektiv erarbeiteten Codes; zum anderen stellen sie sich Trittbrettfahrern entgegen, da stets nachvollziehbar bleibt, welche Firmen sich auf welche Elemente stützen und inwieweit sie an deren Entwicklung teilhaben (Henkel et al. 2014). Daneben bietet es sich in der Schöpfung von Softwareprodukten heute ohnehin oft an, auf existenten quelloffenen Elementen aufzubauen.

*Elitezentrierte Projektgemeinschaften* stützen sich ebenfalls zu einem Gutteil auf die Beiträge unternehmensaffiliierter Entwickler; sie stehen aber nicht unter der Kontrolle eines gewerblichen Akteurs. Ihre Koordination erfolgt entlang ausdifferenzierter Entscheidungspyramiden bzw. einem „lieutenant system built around a chain of trust" (Kernel.Org 2016), an deren Spitze oft ihr Gründer als „benevolent dictator"" (z. B. Linux), ein langfristig installiertes Führungsteam (z. B. Mozilla) oder ein gewählter Projektleiter (z. B. Debian) steht. Dieses Top-Down-Management beschneidet die Spielräume der Beteiligten, wirkt aber auch einer Fragmentierung der Vorhaben entgegen. In Debian und Mozilla sind die Projektrichtlinien formal fixiert worden; im Linux-Kernel-Projekt haben sich entlang des Führungsstils Torwalds' hingegen lediglich „opaque governing norms" herausgebildet, die im Konfliktfall der proklamierten Offenheit entgegenlaufen können: „Without [...] a clear mechanism of accountability those injured by or excluded from peer production processes have very limited recourse" (Kreiss et al. 2011, S. 252).

*Heterarchisch angelegte Infrastrukturvorhaben*, deren Produkte verbreiteten Einsatz unter der „sichtbaren" Oberfläche von IT-Architekturen erfahren, sind eng mit korporativen Kontexten verwoben: Entweder sie fußen (wie die Programmierumgebung Eclipse) auf ehemals proprietären Architekturen oder sie waren (wie Apache) durch ein rasantes organisches Wachstum gekennzeichnet, da sie Lösungen für zuvor nicht adressierte Bereiche boten, und daher früh für Unternehmen interessant (Greenstein/Nagle 2014). Heute werden Infrastrukturvorhaben primär von mittleren und großen IT-Firmen getragen; ihre Communitys werden aber nicht durch korporative Kernzirkel angeleitet, sondern operieren unter dem Dach von Stiftungen und sind horizontal entlang von Arbeitsgruppen strukturiert. Funktionsträger werden meist meritokratisch (d.h. leistungsorientiert) designiert; allerdings können sich von Unternehmen dafür freigestellte Entwickler langfristig in der Regel intensiver als Freizeitprogrammierer in die Projekte einbringen.

*Egalitär ausgerichtete Peer-Production-Communitys* dienen qua Eigendefinition der marktunabhängigen und gleichberechtigten Kollaboration unter Freiwilligen; sie bilden allerdings – wie sich an KDE (Desktop-Umgebung), GNU oder LibreOffice zeigt – ab einer gewissen Größe in der Regel gleichermaßen herausgehobene Führungsstrukturen aus und verfügen überdies über einen stabilen Pool an korporativen Stakeholdern. Intrinsisch motivierte Communitys wie Arch (Linux-Distribution) oder jEdit (Texteditor) hingegen richten ihre Produkte auf spezifische Anspruchsgruppen aus, sind für den allgemeinen Markt irrelevant, werden von kleinen Teams getragen und konnten daher bis dato auf die Ausbildung ausgeprägter sozialer Strukturierungen verzichten. Sobald aber die Gemeinschaft wächst und sich ihre Interaktionen mit externen Akteuren erhöhen, werden offenbar trotz aller technischen Effektivierungen auch in gesellschaftsethisch fundierten Vorhaben „kathedralartige" Koordinationsmuster notwendig (vgl. zu LibreOffice Corbet 2015).

Der gemeinsame Nenner aller vier Projektvarianten besteht in den dahinterliegenden quelloffenen Lizenzmodellen, die ihre Produkte vor direkter Proprietarisierung schützen. Mit „Rebel Code" (Moody 2002) hat all dies allerdings nicht mehr viel gemein: Die Verschränkungen mit marktlichen Kontexten sind oft ausgeprägt; trotz der technisch erweiterten Austauschmöglichkeiten bilden sich regelmäßig klassische hierarchische Entscheidungsstrukturen heraus; entgegen dem Eindruck, „that organizations [...] really don't matter as much as they used to" (Suddaby 2013, S. 1009), verlieren klassische Unternehmen in Open-Source-Projekten keineswegs an Einfluss, sondern bleiben als deren Initiatoren und Financiers prominent im Spiel.

## 4. Einordnung: Open Source als soziotechnisch verstetigte kollektive Invention

Die Annahme, dass die technischen Infrastrukturen des Internets einer „ossification of power" in Open-Source-Projekten entgegenwirken, da sie dezentrale Arbeitsweisen beförderten und „easier pathways to challenge oligarchy" böten (Benkler 2013, S. 225), lässt sich insofern in ihrer Radikalität ebenso wenig halten wie das Postulat einer „networked information economy" (Benkler 2006, S. 3), in der korporative Akteure gegenüber „nonproprietary, voluntaristic, self-organized practices" (Benkler 2013, S. 213) schlechthin an Relevanz verlieren sollen. Aus technik- und organisationssoziologischer Sicht lassen sich dafür zwei Gründe herausstellen:

*Erstens* bilden die in den Projekten genutzten Infrastrukturen und Dienste zwar die handlungsorientierende Grundlage für die dortigen Arbeitsprozesse und effektivieren deren Koordination; sie führen aber keineswegs zu einer Marginalisierung sozialer Strukturierungsleistungen: Auch in Open-Source-Communitys und ähnlichen digitalen Gemeinschaften (z. B. Wikipedia) bilden sich mit der Zeit kollektiv akzeptierte Regeln, Leitorientierungen und abgestufte Entscheidungsstrukturen mit prägnanten Einflussasymmetrien heraus. Erst diese voraussetzungsreichen sozialen Institutionalisierungsdynamiken führen vice versa dazu, dass ▶

ein quelloffenes Softwareprojekt in der Eigen- wie Fremdbeobachtung als Einheit wahrgenommen wird und übergreifende Strategiefähigkeit entwickeln kann (Dolata/Schrape 2016; O'Mahony/Ferraro 2007).

*Zweitens* können korporative Akteure im Normalfall erheblich systematischer als interessenbasierte Gemeinschaften handeln, weil sie über formalisierte Entscheidungsroutinen verfügen und ihre Ressourcen losgelöst von den situativen Präferenzen ihrer Mitglieder einsetzen können (Perrow 1991). Dies zeigt sich auch in Open-Source-Communitys: Unternehmen und andere Organisationen können ihre Ressourcen auf lange Sicht erwartungssicherer als individuelle Beiträger in die Vorhaben einbringen, tragen so zur Erhöhung von deren Planungssicherheit bei und haben dort demzufolge oft einen nicht zu unterschätzenden Einfluss. Ferner verfügen eigenständige Projekte meist über assoziierte gemeinnützige Organisationen, die als Dachidentitäten dienen und die Gemeinschaft bei Konflikten stabilisieren (Ahrne et al. 2016).

Lässt sich demnach nicht nur von einer „Korporatisierung", sondern auch von einer stetig intensiveren „Kolonialisierung" von Open-Source-Projekten durch klassische Unternehmen sprechen?[3] Tatsächlich zeigt sich in der Langfristrekonstruktion, dass sich verbreitete Beispiele für das gerne kolportierte Idealbild einer unabhängigen „commons-based peer production" vorrangig in der Frühzeit freier Software finden lassen. Bereits Ende der 1990er Jahre wurden günstig lizenzierbare quelloffene Softwarekomponenten indes zu Eckpfeilern einer internetzentrierten Startup-Szene und seit der Jahrtausendwende involvieren sich zunehmend auch andere Firmen in Open-Source-Vorhaben. Ein solcher Entwicklungsverlauf erscheint aus Sicht der Innovationsforschung erst einmal nicht ungewöhnlich: Wie andere Nischeninnovationen wurden freie Softwareprojekte zunächst getragen „by small networks of dedicated actors, often outsiders or fringe actors" und unterlagen einer Professionalisierung und Aneignung durch etablierte Akteure, sobald sie für „mainstream markets" interessant wurden (Geels/Schot 2007, S. 400).

Im Unterschied zu früheren Episoden der „collective invention", in denen Organisationen oder individuelle Akteure ihre Wissensbestände in vom allgemeinen Markt abgekoppelten Nischen offen geteilt und so von „cumulative advance" profitiert haben (Allen 1983, S. 23) – z. B. in der Entwicklung von LCD-Displays (1970/80er Jahre) oder auf dem Feld der erneuerbaren Energien (1980/90er Jahre), sind quelloffene Softwareprojekte allerdings auch über die Initialphase von Innovationsprozessen hinaus bzw. nach der Herausbildung dominanter Lösungen und deren kommerzieller Verwertung überlebensfähig geblieben (Osterloh/Rota 2007), was sich aus Sicht der angestellten Rückbetrachtungen auf folgende interagierende Faktoren zurückführen lässt:

(1) Zum einen haben sich in der freien Softwareentwicklung neben informellen Regeln früh rechtlich belastbare *Lizenzmodelle* herausgebildet, die eine Proprietarisierung und direkte Kommodifizierung der kollektiven Arbeitsresultate verhindern. Sie sind heute die elementare Geschäftsgrundlage aller Open-Source-Projekte und bieten einen erwartungssicheren institutionellen Unterbau für den punktuellen Wissensaustausch sowie die zielorientierte Kollaboration zwischen Unternehmen wie Einzelentwicklern außerhalb formaler Kooperationen.

(2) Zum anderen haben die sich zeitgleich verbreitenden *Onlinetechnologien* nicht nur die Überprüfung der Einhaltung dieser lizenzrechtlichen Bedingungen (ähnlich wie bei wissenschaftlichen Plagiaten) erheblich effektiviert, den Zugang den Entwicklungsvorhaben sowie die Diffusion und den Einsatz ihrer Produkte erleichtert, sondern auch zu der Lösung eines branchenzentralen Problems beigetragen: der Koordination großer Projekte mit örtlichen verteilten Entwicklern aus verschiedenen Arbeitskontexten (Brooks 1975).

(3) Und darüber hinaus haben sich Open-Source-Entwicklungsvorhaben in einer seit 30 Jahren beständig expandierenden und durch sehr kurze Innovationszyklen geprägten Softwareindustrie als wichtige *Inkubatoren* für neue Produktlinien und branchenfundamentale Infrastrukturen (wie z. B. die Cloud-Computing-Architektur OpenStack) erwiesen, gerade auch weil sich quelloffene Software ohne administrativen Aufwand durch die ausführenden Entwickler selbst erproben und an ihre jeweiligen Anforderungskontexte anpassen lässt.

Insoweit waren ab den 1980er Jahren nicht nur die erweiterten Formen der elektronischen Vernetzung, sondern ebenso die Kristallisation übergreifend akzeptierter Arbeitskonventionen und vor allem anderen die Definition rechtlich tragfähiger Lizenzmodelle für den anhaltenden Erfolg quelloffener Entwicklungsvorhaben von zentraler Bedeutung. „Copyleft-Lizenzen" und ihre Ableitungen haben im Verbund mit den kommunikationserleichternden Eigenschaften der Onlinetechnologien in einer stetig umfassender informatisierten Arbeitswelt den soziotechnischen Rahmen für eine *auf Dauer gestellte Form kollektiver Invention* aufgespannt, die zunächst in subversiven Nischen Anwendung fand, nach der Jahrtausendwende als ergänzende Arbeitsmethode von der kommerziellen Softwareindustrie adaptiert wurde und heute zu einem festen Baustein der Innovationsstrategien aller etablierten IT-Unternehmen geworden ist.

3 Klaus Dörre (2012) beschreibt ähnliche Dynamiken als „kapitalistische Landnahme" im Sinne einer Ökonomisierung vormals von Wirtschaftslogik befreiter Felder – so z. B. die Kommunikationssphäre des Internets, die zunächst als von marktlichen Zwängen abgekoppelter „Cyberspace" (Barlow 1996) verstanden wurde.

## 5. Resümee: Eine offenere und demokratischere Arbeitswelt?

Projektförmige Arbeitsweisen in quelloffenen Entwicklungsvorhaben und eingespielte Formen ökonomischer Koordination stehen also nicht in einem konkurrierenden, sondern in einem komplementären Verhältnis zueinander. Open-Source-Projekte haben in den letzten zwei Jahrzehnten die Kollaboration zwischen Entwicklern aus divergenten Kontexten, die sachbezogene Zusammenarbeit ansonsten fallweise in Konkurrenz stehender Marktteilnehmer wie auch die innerorganisationalen Produktionsmodi flexibilisiert und den Softwaremarkt insgesamt durchlässiger gemacht. Gleichzeitig zeigt sich mit Blick auf dauerhaft aktive quelloffene Softwareprojekte aber auch, dass frei verfügbarer Quellcode nicht zwangsläufig in transparenteren Koordinationsmustern als in anderen Arbeitszusammenhängen, einer Disintermediation langfristig kristallisierter Ressourcen- und Einflussverteilungen oder einer generellen Demokratisierung von Innovationsprozessen resultiert. Pointiert formuliert: Offener Code alleine mündet noch nicht in offenen Gesellschaftsstrukturen.

Der ungefilterte Übertrag der Vorstellung einer „commons-based peer production" in Reinform, die sich auch in Open-Source-Communitys nur selten aufspüren lässt, auf angrenzende sozioökonomische Bereiche (z. B. Rifkin 2014) oder gesellschaftspolitische Felder (z. B. Bennett et al. 2014) bleibt insofern bestenfalls irreführend. Schlechtestenfalls überdecken entsprechende Narrative indes mit der digitalen Rekonfiguration der Gesellschaft einhergehende Entwicklungstendenzen, die dem Ideal einer offeneren und demokratischeren Wirtschaftswelt entgegenstehen – etwa eine schleichende Erosion „des historisch gewachsenen Systems der Regulation von Arbeit" (Boes et al. 2014, S. 71), die Einschränkung fundamentaler Verbraucherrechte durch die Nutzungsbedingungen vieler Onlinedienste oder eine medienhistorisch singuläre globale Anbieterkonzentration auf dem Feld der kommunikationstechnischen Infrastrukturen (vgl. zu dem verbreiteten Phänomen des „Openwashings" überdies: Pomerantz/Peek 2016).

Vor diesem Hintergrund erscheint es für sozialwissenschaftliche Beobachter wenig voranbringend, oft direkt in Kalifornien geschöpfte Schlagworte wie „Open Innovation", „Web 2.0" oder auch „Open Source" unmittelbar als „quasi-soziologische Fachbegriffe" (Süssenguth 2015, S. 99) zu adaptieren. Ertragreicher könnte es sein, die mit diesen Vokabeln verknüpften Erwartungen auf generalisierbare Muster abzuklopfen und deren gesellschaftliche Wirkungen nachzuzeichnen. Denn obgleich die mit Open-Source-Projekten und jüngeren Phänomenen wie der „Maker"-Kultur verknüpften „narratives of openness and individual empowerment" bis dato übergreifend nicht eingelöst wurden (Ames et al. 2014, S. 1088), erfüllen sie in ihren Anwendungskontexten elementare Funktionen: Sie erzeugen Aufmerksamkeit für neue Entwicklungspfade, kanalisieren den Diskurs, tragen zur Konstitution von Innovationsnischen bei und dienen als Legitimationsbasis in wirtschaftlichen, wissenschaftlichen wie politischen Entscheidungsprozessen. Insofern lassen sich die thematisierten Openness-Narrative durchaus als „produktive Kommunikationstypen" betrachten (Dickel/Schrape 2015, S. 455) – sofern sie nicht als empirische Tatsachenbeschreibungen missverstanden werden, wie das im Softwarebereich lange der Fall war. ■

### LITERATUR


**Ahrne, G./Brunsson, N./Seidl, D.** (2016): Resurrecting organization by going beyond organizations, in: European Management Journal 34 (2), S. 93–101

**Allen, R.** (1983): Collective invention, in: Journal of Economic Behavior & Organization 4 (1), S. 1–24

**Alphabet Inc.** (2016): Form 10-K 2015, https://abc.xyz/investor/ (letzter Zugriff: 15.07.2016)

**Ames, M./Bardzell, J./Bardzell, S./Lindtner, S./Mellis, D./Rosner, D.** (2014): Making cultures: empowerment, participation, and democracy–or not?, in: Proceedings of the 32nd annual ACM Conference on Human Factors in Computing Systems, S. 1087–1092

**Barlow, J. P.** (1996): A declaration of independence of cyberspace, https://www.eff.org/de/cyberspace-independence (letzter Zugriff: 15.07.2016)

**Benkler, Y.** (2002): Coase's Penguin, or, Linux and ‚the nature of the Firm', in: Yale Law Journal 112 (3), S. 369–446

**Benkler, Y.** (2004): Intellectual property: commons-based strategies and the problems of patents, in: Science 305 (5687), S. 1110–1011

**Benkler, Y.** (2006): The wealth of networks, New Haven

**Benkler, Y.** (2013): Practical anarchism, peer mutualism, market power, and the fallible state, in: Politics & Society 41 (2), S. 213–251

**Benkler, Y./Nissenbaum, H.** (2006): Commons-based peer production and virtue, in: Journal of Political Philosophy 14 (4), S. 394–419

**Bennett, W. L./Segerberg, A./Walker, S.** (2014): Organization in the crowd: peer production in large-scale networked protests, in: Information, Communication & Society 17 (2), S. 232–260

**Bergquist, M./Ljungberg, J./Rolandsson, B.** (2012): Justifying the value of open source. ECIS proceedings, http://aisel.aisnet.org/ecis2012/122/ (letzter Zugriff: 15.07.2016)

**Bezroukov, N.** (1999): A second look at the cathedral and the bazaar, in: First Monday 4 (12), http://firstmonday.org/article/view/708/618 (letzter Zugriff: 15.07.2016)

**Boes, A./Kämpf, T./Langes, B./Lühr, T./Steglich, S.** (2014): Cloudworking und die Zukunft der Arbeit – Kritische Analysen am Beispiel der Strategie „Generation Open" von IBM, Kassel

**Brooks, F.** (1975): The mythical man-month, Reading

**Bulajewski, M.** (2011): The peer production illusion, part I, in: MrTeaCup vom 19.11., http://www.mrteacup.org/post/peer-production-illusion-part-1.html (letzter Zugriff: 15.07.2016)

**Corbet, J.** (2015): Development activity in LibreOffice and OpenOffice, in: LWN.net vom 25.3., https://lwn.net/Articles/637735/ (letzter Zugriff: 15.07.2016)

**Corbet, J./Kroah-Hartman, G./McPherson, A.** (2009–2015): Linux Kernel development report, http://www.linuxfoundation.org/publications/linux-foundation (letzter Zugriff: 15.07.2016)

**Dahlander, L./Magnusson, M.** (2008): How do firms make use of Open Source communities?, in: Long Range Planning 41 (6), S. 629–649

**Dickel, S./Schrape, J.-F.** (2015): Dezentralisierung, Demokratisierung, Emanzipation. Zur Architektur des digitalen Technikutopismus, in: Leviathan 43 (3), S. 442–463

**Dörre, K.** (2012): Landnahme, das Wachstumsdilemma und die ‚Achsen der Ungleichheit', in: Berliner Journal für Soziologie 22 (1), S. 101–128

**Dolata, U.** (2015): Volatile Monopole. Konzentration, Konkurrenz und Innovationsstrategien der Internetkonzerne, in: Berliner Journal für Soziologie 24 (4), S. 505–529

**Dolata, U./Schrape, J.-F.** (2016): Masses, crowds, communities, movements: collective action in the internet age, in: Social Movement Studies 15 (1), S. 1–18

**Driver, M.** (2014): Within the enterprise, Open Source must coexist in a hybrid IT portfolio. Gartner Inc. Research Report, Stamford

**FSF (Free Software Foundation)** (1989): GNU General Public License (GPL) Version 1.0, http://www.gnu.org/licenses/old-licenses/gpl-1.0.en.html (letzter Zugriff: 15.07.2016)

**Gates, B.** (1976): An open letter to hobbyists, in: Computer Notes 1 (9), S. 3

**Geels, F. W./Schot, J. W.** (2007): Typology of sociotechnical transition pathways, in: Research Policy 36 (3), S. 399–417

**Greenstein, S./Nagle, F.** (2014): Digital dark matter and the economic contribution of apache, in: Research Policy 43 (4), S. 623–631

**Gulley, N./Lakhani, K.** (2010): The determinants of individual performance and collective value in private-collective software innovation, Harvard Business School TOMU Working Paper 10/065

**Henkel, J./Schöberl, S./Alexy, O.** (2014): The emergence of openness: How and why firms adopt selective revealing in open innovation, in: Research Policy 43 (5), S. 879–890 ▶

**Jaeger, T.** (2010): Enforcement of the GNU GPL in Germany and Europe, in: Journal of Intellectual Property, Information Technology and E-Commerce Law 1 (1), S. 34–39
**Kernel.Org** (2016): How to get your change into the Linux Kernel, https://www.kernel.org/doc/Documentation/SubmittingPatches (letzter Zugriff: 15.07.2016)
**Kolassa, C./Riehle, D./Riemer, P./Schmidt, M.** (2014): Paid vs. volunteer work in Open Source, in: Proceedings 47th Hawaii Conference on System Sciences, S. 3286–3295
**Kranich, N./Schement, J. R.** (2008): Information commons, in: Annual Review of Information Science and Technology 42 (1), S. 546–591
**Kreiss, D./Finn, M./Turner, F.** (2011): The limits of peer production: some reminders from Max Weber for the network society, in: New Media & Society 13 (2), S. 243–259
**Lakhani, K./Hippel, E. v.** (2003): How open source software works, in: Research Policy 32 (6), S. 923–943
**Lerner, J.** (2012): The architecture of innovation, Boston
**Lerner, J. /Tirole, J.** (2002): Some simple economics of Open Source, in: Journal of Industrial Economics 50 (2), S. 197–234
**Lessig, L.** (1999): Open code and open societies, in: Chicago Kent Law Review 74 (3), S. 1405–1420
**Martin, J.** (1991): Rapid application development, Indianapolis
**Microsoft Inc.** (2015): 2014 annual report, http://www.microsoft.com/investor/reports/ (letzter Zugriff: 15.07.2016)
**Miller, P./Nelson, L. E.** (2016): Open source powers enterprise digital transformation, Forrester Inc. Report, Cambridge
**Moody, G.** (2002): Rebel code. The inside story of Linux and the Open Source revolution, New York
**Netscape Communications** (1998): Netscape announces Mozilla.org, Press Release vom 23.2.
**O'Mahony, S.** (2003): Guarding the commons. How community managed software projects protect their work, in: Research Policy 32 (7), S. 1179–1198
**O'Mahony, S./Ferraro, F.** (2007): The emergence of governance in an Open Source community, in: Academy of Management Journal 50 (5), S. 1079–1106
**O'Reilly, T.** (2000): Re: Open Source and OpenGL, in: Ask Tim Forum vom 15.12., http://archive.oreilly.com/pub/a/oreilly/ask_tim/2000/opengl_1200.html (letzter Zugriff: 15.07.2016)
**Osterloh, M./Rota, S.** (2007): Open source software development: just another case of collective invention?, in: Research Policy 36 (2), S. 157–171
**Pomerantz, J./Peek, R.** (2016): Fifty shades of Open, in: First Monday 21 (5), http://dx.doi.org/10.5210/fm.v21i5.6360, (letzter Zugriff: 15.07.2016)
**Perrow, C.** (1991): A society of organizations, in: Theory & Society 20 (6), S. 725–762
**Raymond, E. S.** (1999): The cathedral and the bazaar, Sebastopol
**Rifkin, J.** (2014): The zero marginal cost society, New York
**Spinellis, D. /Giannikas, V.** (2012): Organizational adoption of Open Source software, in: Journal of Systems and Software 85 (3), S. 666–682
**Schrape, J.-F.** (2016): Open-Source-Projekte als Utopie, Methode und Innovationsstrategie, Glückstadt
**Spreeuwenberg, K./Poell, T.** (2012): Android and the political economy of the mobile internet, in: First Monday 17 (7), http://dx.doi.org/10.5210/fm.v17i7.4050 (letzter Zugriff: 15.07.2016)
**Stallman, R.** (1983): New UNIX implementation, http://bit.ly/1DSDoXW (letzter Zugriff: 15.07.2016)
**Stallman, R.** (2002): Free software, free society, Boston
**Suddaby, R.** (2013): Book review: the Janus face of commercial Open Source software communities, in: Organization Studies 34 (7), S. 1009–1011
**Süssenguth, F.** (2015): Die Organisation des digitalen Wandels, in: Ders. (Hrsg): Die Gesellschaft der Daten, Bielefeld, S. 93–121
**Tapscott, D./Williams, A. D.** (2006): Wikinomics, New York
**Torvalds, L.** (1998): LINUX manifesto. Interview, in: Boot Magazine 7–8/1998, S. 32–37
**Torvalds, L.** (2002): Re: [PATCH] Remove bitkeeper documentation from Linux tree, in: Linux Kernel mailinglist vom 20.4., http://lwn.net/2002/0425/a/ideology-sucks.php3 (letzter Zugriff: 15.07.2016)
**W3techs Surveys** (2016): Technologies overview, http://w3techs.com/technologies (letzter Zugriff: 15.07.2016)
**Weber, S.** (2000): The Political economy of Open Source, BRIE Working Paper (140), Berkeley

## AUTOR

**JAN-FELIX SCHRAPE**, Dr. phil., forscht und lehrt am Institut für Sozialwissenschaften der Universität Stuttgart. Arbeitsschwerpunkte: Innovations- und Techniksoziologie, Medien- und Kommunikationswissenschaft, Kollektivität in der digitalen Gesellschaft.

@ felix.schrape@sowi.uni-stuttgart.de