Open Source Projects as Incubators of Innovation: From Niche Phenomenon to Integral Part of the Software Industry

Schrape, Jan-Felix

Research Contributions to Organizational Sociology and Innovation Studies / Stuttgarter Beiträge zur Organisations- und Innovationssoziologie

SOI Discussion Paper 2017-03

Open Source Projects as Incubators of Innovation

From Niche Phenomenon to Integral Part of the Software Industry

Jan-Felix Schrape

University of Stuttgart
Germany

Institute for Social Sciences
Organizational Sociology and Innovation Studies

Jan-Felix Schrape

Open Source Projects as Incubators of Innovation. From Niche Phenomenon to Integral Part of the Software Industry.

SOI Discussion Paper 2017-03
University of Stuttgart
Institute for Social Sciences
Department of Organizational Sociology and Innovation Studies
Seidenstr. 36
D-70174 Stuttgart

Editor

Prof. Dr. Ulrich Dolata
Tel.: +49 711 / 685-81001
ulrich.dolata@sowi.uni-stuttgart.de

Managing Editor

Dr. Jan-Felix Schrape
Tel.: +49 711 / 685-81004
felix.schrape@sowi.uni-stuttgart.de

Research Contributions to Organizational Sociology and Innovation Studies
Discussion Paper 2017-03 (May 2017)
ISSN 2191-4990

Jan-Felix Schrape is senior researcher at the Department of Organizational Sociology and Innovation Studies, University of Stuttgart (Germany).

felix.schrape@sowi.uni-stuttgart.de

Abstract

Over the last 20 years, open source development has become an integral part of the software industry and a key component of the innovation strategies of all major IT providers. Against this backdrop, this paper seeks to develop a systematic overview of open source communities and their socio-economic contexts. I begin with a reconstruction of the genesis of open source software projects and their changing relationships to established IT companies. This is followed by the identification of four ideal-typical variants of current open source projects that differ significantly in their modes of coordination and the degree of corporate involvement. Further, I examine why open source projects have mainly lost their subversive potential while, in contrast to former cases of collective invention, remaining viable beyond the emergence of predominant solutions and their commercial exploitation: In an industry that is characterized by very short innovation cycles, open source projects have proven to be important incubators for new product lines and branch-defining infrastructures. They do not compete against classical forms of production but instead complement and expand these.

Zusammenfassung

In den letzten 20 Jahren ist die Open-Source-Entwicklung zu einem integralen Bestand der Softwareindustrie und zu einem zentralen Baustein der Innovationsstrategien aller großen IT-Anbieter geworden. Vor diesem Hintergrund entfaltet dieses Papier einen systematisierenden Überblick über Open-Source-Communities und ihre sozioökonomischen Kontexte. Nach einer historischen Rekonstruktion zur Ausdifferenzierung quelloffener Softwareprojekte und ihren sich wandelnden Relationen zu etablierten Unternehmen werden vier Varianten derzeitiger Open-Source-Projekte voneinander abgegrenzt, die sich in ihren Koordinationsweisen und dem Grad ihrer Unternehmensnähe signifikant voneinander unterscheiden. Daran anknüpfend wird herausgearbeitet, aus welchen Gründen Open-Source-Projekte inzwischen ihre subversive Formatierung weitgehend verloren haben, aber im Gegensatz zu früheren Ausprägungen kollektiver Invention überlebensfähig geblieben sind: In einer durch sehr kurze Innovationszyklen geprägten Softwareindustrie haben sich quelloffene Entwicklungsvorhaben als zentrale Inkubatoren für neue Produktlinien und branchenfundamentale Infrastrukturen erwiesen. Projektförmige Arbeitsweisen in Open-Source-Gemeinschaften und eingespielte Formen ökonomischer Koordination stehen nicht in einem konkurrierenden, sondern in einem komplementären Verhältnis zueinander.

Contents

1 Introduction

The term *open*, used in phrases from "open science" to "open innovation" and "open government," has become part of the standard vocabulary in the modern digital era. Today, projects of all kinds flaunt the attribute of openness and its associated promise of more decentralized and democratic organizational and coordination structures. More specifically, the promise entails that technology could break with the traditional distribution of social roles, override established boundaries of the production and consumption sphere and empower once-passive citizens, users, and consumers.

An important starting point for the popularity of the openness paradigm is the rapidly increasing relevance of open source projects in software development since the turn of the millennium. In the social sciences, accustomed to regarding intellectual property rights as primary drivers of innovation processes (Arrow 1962; Romer 1990), this increase was initially received with surprise (Lessig 1999). However, not long thereafter open source became acknowledged as an emerging production model that is based on voluntary and self-directed collaboration among equals and that could reduce the significance of traditional corporations in the working world and break with well-established forms of socio-economic coordination, such as the market or hierarchy (Lakhani & Hippel 2003). In that context, the concept of "commons-based peer production," introduced by Yochai Benkler (2002), gained traction. Hailed as a technically effective "collaboration among large groups of individuals […] without relying on either market pricing or managerial hierarchies to coordinate their common enterprise" (Benkler & Nissenbaum 2006: 381), commons-based peer production was to be accompanied with "systematic advantages […] in identifying and allocating human capital/creativity" (Benkler 2002: 381). More recently, the concept has been increasingly applied in adjacent fields, such as the production of material goods ("Maker Economy") or the service sector (e.g., Rifkin 2014).

However, long-term observations of open source software projects have shown that leading IT companies are gaining considerable influence over important projects; that the growth of the developer communities goes hand in hand with the formation of distinct hierarchical decision-making patterns; and that firmly established projects are not run by intrinsically motivated volunteers—"satisfying psychological needs, pleasure, and a sense of social belonging" (Benkler 2004: 1110)—but are mainly based on the contributions of employed developers. For example, in the Linux kernel development project, often referred to as a typical open source software project, more than 85 percent of the updates were made by programmers who "are being paid for their work" (Corbet & Kroah-Hartman 2016: 12). In light of this, the still oft-made claim that open source software communities are radical alternatives, or counter-projects, to industrial production (e.g., Kostakis et al. 2015; Bennett & Segerberg 2015: 183f.) are essentially blanket statements that do not hold.

Against this backdrop, this paper seeks to develop a systematic overview of open source communities and their socio-economic contexts on the basis of aggregated market data, an evaluation of publications from projects and companies, mailing lists, industry news, web content and literature from the past decades as well as informal background talks with software engineers from Germany, Switzerland and California. I begin with a reconstruction of the genesis of open source software projects and their changing relationships to established IT companies (*Section 2*). This is followed by the identification of four ideal-type variants of current open source projects that differ significantly from each other in their modes of coordination and the degree of corporate involvement—being corporate-led collaboration projects, elite-centered project groups, heterarchical infrastructure projects and egalitarian-oriented peer production communities (*Section 3*). I then examine why open source projects have mainly lost their subversive potential while, in contrast to former cases of collective invention, remaining viable beyond the emergence of predominant solutions and their commercial exploitation: In an international software industry that is characterized by very short innovation cycles, open source projects have proven to be important incubators for new product lines and branch-defining infrastructures (*Section 4*). The final section assesses broader societal implications of the developments under discussion.

2 The genesis and institutionalization of open source software projects

Soon after open source software projects became widely known, a number of books and articles were published that, offering initial explanations for their success and underlining their subversive character, essentially form the basis of the social sciences view of open source to this day (e.g., Weber 2000; Moody 2002). These texts were primarily oriented towards the narratives coming from the developer scene itself and, with very few exceptions (e.g., Lerner & Tirole 2002), dispensed with any socio-economic contextualization. As the following historical reconstruction shows, however, the dividing line between free and commercial software development has never been clear-cut, and the involvement in open source projects has become an intentional, studied component of the innovation strategies of all big IT companies.

2.1 Free software as utopia

The development of the free software movement in the 1980s can be seen as a direct response to the previously initiated commodification of software. The first digital computers from the 1950s had been developed in close cooperation between manufacturers and users, with computer programs not yet perceived as a product that is independent from hardware but rather “as a research tool to be developed and im-proved by all users” (Gulley & Lakhani 2010: 6). Starting with the end of the 1960s, however, software began being acknowledged as a separate product, prompted mainly by antitrust procedures—for example, against International Business Machines (IBM), which was criticized for pushing competitors out of business with its combined offer of hardware and software (Burton 2002)—and the founding of the first specialized software companies (Fisher et al. 1983).

The spread of mini-computers (e.g., PDP-1) also played an important role in the development of a stand-alone software sector. These types of computers differed from their predecessors, the larger mainframe systems, in that their operation was much less costly, due to which they were accessible to a greater number of people and applicable to a wider range of contexts. In addition, advanced input and output interfaces (e.g., cathode ray tubes and teleprinters) engendered the development of new software genres (e.g., word processing, graphic design). At North American universities, especially, mini-computers, often donated by their manufacturers to the institutes, offered a breeding ground for the formation of informal project groups, whose members called themselves “hackers” (Levy 1984). These groups sought to overcome the limitations of existing computer systems and paved the way for the amateur computing scene that developed from the mid-1970s alongside the emergence of the first home computers (e.g., Altair 8800). However, the shared problem of the software architectures developed in these contexts was their *lack of legal protection*: they were published as public goods yet were hardly protected against proprietarization. For example, the Unix operating system, co-developed at universities, was commodified by AT&T from 1983 on—as soon as permitted under antitrust law (Holtgrewe & Werle 2001). Or, the computer game *Spacewar!*, programmed by students from the Massachusetts Institute of Technology (MIT) in 1961/1962, was utilized as the basis of numerous commercial video arcade machines of the 1970s and 1980s (Lowood 2009).

What commercial software providers liked less about the computer hobbyist scene was its predilection to share and circulate programs without paying or charging for them. In an open letter addressed to that community, the then-young software entrepreneur Bill Gates (1976: 3) complained about this circumstance as follows:

> “As the majority of hobbyists must be aware, most of you steal your software. Hardware must be paid for, but software is something to share. Who cares if the people who worked on it get paid? Is this fair? […] One thing you do is prevent good software from being written. Who can afford to do professional work for nothing? What hobbyist can put 3-man years into programming, finding all bugs, documenting his product and distribute for free? […] Most directly, the thing you do is theft.”

As a result of this conflict, by the early 1980s most software products were sold solely as binary files that could not be changed and that had no accessible source code. At the same time, several amendments to copyright law in the United States increased the protection and excludability of software products (Menell 2002). As a socio-ethical statement about this turn of events, the MIT employee Richard Stallman announced in

1983, in the then still-young Usenet, his plan to develop a free and independent operating system to go by the recursive acronym GNU ("GNU's Not Unix"): "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 […]."

Although GNU is to this day not suitable for everyday use as a standalone operating system, Stallman's project proved to be the breeding ground for free software development. In 1985, he established the Free Software Foundation (FSF), which, starting with 1988, enlisted the first large-scale industrial sponsors such as the hardware manufacturers Sony and Hewlett-Packard, who had an interest in inexpensively licensable software. The most important innovation, however, was the *introduction of robust licensing models*, like the General Public License (GPL) published in 1989, which legally ensure that any forks of the free software remain free:

> "Each time you redistribute the Program (or any work based on 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. You may not impose any further restrictions on the recipients' exercise of the rights granted herein." (FSF 1989)

From 2001 on, violations of the GPL were the object of numerous court proceedings against companies such as Skype, Cisco, and D-Link (Stiller 2011; Jaeger 2010). It should be noted, however, that "the court of public opinion" played an equally important role in Usenet, and later on the World Wide Web, for the establishment of the reciprocity principles in the GPL (O'Mahony 2003: 1189).

That said, the success of the GNU project remained limited at first due to its reliance on costly workstations and its strong ideological connotations—two problems to which the Linux kernel development project offered a solution. Linux was introduced in 1991 by then-student Linus Torvalds as a free operating system kernel for the more affordable micro-computers, and was, therefore, attractive to a larger number of developers. In addition, the Linux kernel project, or rather its founder, was characterized from the start by a much more liberal attitude than the Free Software Foundation: "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 and because others might find it useful, not because I got religion'." (Torvalds 2002) Another reason for the success of Linux was the spread of the World Wide Web from 1993 on, as it facilitated both access to and participation in the project and its coordination. Nonetheless, the Linux kernel project too was initially known only within expert circles.

Indeed, it was not until the publication of the much-anticipated book *The Cathedral and the Bazaar* in 1999, already presented as an essay by software developer Eric S. Raymond in 1997, that the Linux kernel became more widely known. The main thesis of the book was: Whereas in traditional production models a program's source code is only published for the final version, with developer groups being hierarchically organized—corresponding to the *cathedral*—the source code in projects like Linux or Fetchmail (then coordinated by Raymond) is always visible and their developer groups are horizontally structured as well as maintained by modular self-organization without central management—corresponding to a *bazaar*. Nonetheless, critical observers observed early on that while in both cases many suggestions came from the project community, the final changes were released by only one person, being Torvalds or Raymond (Connell 2000; Bezroukov 1999). In other words:

> "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. And even if somebody thought I was doing a bad job (which is fairly rare) and that somebody decides that 'I really want to fix this feature,' there's a really big hurdle to convince everybody else that he CAN fix that feature." (Torvalds 1998: 36)

GNU and Linux stand as two main flagship projects for free software development of the 1980s and 1990s whose success was greatly facilitated by the increased efficiency of communication brought about by the internet. This environment spurred the emergence of legal instruments such as the General Public License (GPL), which protect collective work results from being claimed or appropriated by any one individual or entity. It also gave rise to informal working conventions, whose uptake could be assessed more readily and directly than before given the increased visibility of communication on the internet. It was in this context that the first narratives circulated that hailed free software development as the new and upcoming way to produce software without asymmetries of power and that eventually gained, at least for some time, currency among social scientists (e.g., Benkler 2002; Tapscott & Williams 2006).

2.2 Open source as method

In the subsequent decade of the 2000s, open source became an increasingly recognized working method within the software industry. Apart from the continuing spread of the internet, this may be attributed to the following dynamics.

First, a growing number of companies began outsourcing the development of software products to the open source field. Of those, Netscape Communications was a rather conspicuous, and early, case in point. When it became evident that Microsoft would be crowding out Netscape Navigator with its Windows-integrated Internet Explorer, Netscape announced in 1998 that it would transfer large portions of its web browser code to the open source *Mozilla project*. This project, which engendered the popular web browser Firefox in 2004, received financial and human resources support from AOL/Netscape until the founding of the Mozilla Foundation in 2003. With its 1998 announcement, Netscape aimed primarily to build and diversify its clientele: "By making our source code available to the Internet community, Netscape can expand its client software leadership by […] building a community that addresses markets and needs we can't address on our own […]."

Secondly, at the beginning of 1998, a developer group that had formed around Eric S. Raymond concluded that the term "free software" could impede the spread of software with a GPL or similar license in commercial contexts given its possible political connotations. They therefore introduced the *new label "open source,"* which they considered to emphasize the superiority of this development model while deflecting from any socio-political aspects (Raymond 1998). As part of that process, they also founded the Open Source Initiative, namely with the help of protagonists such as Tim O'Reilly, who was to later coin the term "Web 2.0." However, to this day, this change of course has not been endorsed by the Free Software Foundation: "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 2001: 57) This disagreement was representative of the fundamental divide that had been fermenting and that, ongoing to this day, some try to evade by means of hybrid acronyms such as FLOSS (Free/Libre Open Source Software) or FOSS (Free & Open Source Software).

The *third* main factor that contributed to the recognition of open source as a working method was the *stock market success* of some open source companies in 1999 as a result of the dot-com boom of the late 1990s. Among these companies were the Linux-oriented hardware vendors VA Linux and Cobalt Networks as well as the software provider Red Hat, which specialized in Linux software architectures for enterprises. The initial public offerings (IPOs) of these three companies were, in fact, among the most spectacular of all time, resulting in mass media attention alongside the associated ripple effect on the open source scene as a whole (e.g., Gelsi 1999).

These interrelated trends and processes, combined with the continued expansion of the IT market, led to the rapid proliferation of open source projects. Indeed, the number of projects grew from only several hundred in 1999 to the several million projects which can today be found on platforms such as GitHub and SourceForge. Given this dramatic increase in the number of projects, accompanied by novel licensing models created by companies and foundations, open source licensing has also been subject to very strong diversification (Table 1). Alongside original "copyleft" licenses such as the General Public License (GPL), which guarantee that free soft-ware must be forked under the same conditions (*strongly protective*), additional licenses have been issued that permit the inclusion of free software in proprietary products as long as these elements remain open source (*weakly protective*) or even permit the publication of subsequent derivations or branches under downright restrictive conditions (*permissive*). This diversity greatly expands the strategic options, especially for commercial stakeholders (Lerner & Schankerman 2010; Lerner & Tirole 2005): After the third version of the GPL was published in 2007, closing previously gaps, Apple, for instance, replaced the GNU compiler collection (GCC) in its development environment Xcode with a solution with a more permissive license; and Google decided from the outset to put the bulk of project-own code from Android under a permissive license.

Table 1: The most commonly used open source licenses worldwide

	e.g., used by	2017 (%)	2010 (%)	Orientation	Publication
GNU Public License 2.0	Linux kernel, WordPress	18	47	strongly protective	1991
MIT License	jQuery, Ruby on Rails	32	6	permissive	1988
Apache License 2.0	Android, Apache HTTP	14	4	permissive	2004
GNU Public License 3.0	GNU	7	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	1	2	permissive	2007
Eclipse Public License	Eclipse	1	1	permissive	2004

Source: Black Duck Knowledgebase (4/2017)

Concurrently, we can observe a corporatization of open source projects in two ways. On the one hand, branch-defining development projects such as the Linux kernel, the Apache HTTP Server and the cloud computing architecture OpenStack are today funded primarily by donations from companies or operate like the browser engine WebKit (Apple) and the mobile operating system Android (Google) under the aegis of commercial providers (Fitzgerald 2006). On the other hand, the developer base of large-scale projects is increasingly financed by business circles. According to Kolassa et al. (2014), in the Linux kernel and 5000 other market-relevant projects, more than 50 percent of all contributions that occurred between 2000 and 2011 were made during standard 9-to-5 working hours. The Linux Foundation (Corbet & Kroah-Hartman 2016), for its part, observed that the portion of independent programmers in kernel development (2009: 18 percent, 2014: 12 percent, 2016: 8 percent) is steadily declining compared to that of company-associated contributors (Table 2).

It is in this way that open source development increasingly became enmeshed with the software industry over the past two decades—albeit not without losing, to a large extent, its initial force as a counterbalance to proprietary production.[1] Overall, apart from smaller projects (such as the Linux variants Arch or Parabola) that are still true

[1] Under the label "inner sourcing" (O'Reilly 2000), an increasing number of firms are adapting the development methods of open source software projects for their internal coordination structures; agile methods although had been in use in the IT industry as early as the 1990s (Martin 1991).

to the original maxims of free software, most open source projects today involve the participation of established IT companies. The latter use these working environments as a means to protect standards that are favorable to them and to expand their internal proprietary and undisclosed development activities through "controlled openings at the edges" (Dolata 2017: 20). In this respect, the blogger Mike Bulajewski (2011) finds, rightly so, the image of open source projects as communities "of volunteer programmers collaborating together in a gift economy" to be an illusion.

Table 2: Contributions to the Linux Kernel (changes, in %)

	2015–2016 (R 3.19–4.7)	**2013–2014** (R 3.11–3.18)	**2011–2013** (R 3.0–3.10)	**2010–2012** (R 2.6.36–3.2)	**2005–2009** (R 2.6.11–2.6.3)
independent	7.7 %	12,4 %	13,6 %	16,2 %	18,2 %
unknown	6.8 %	4,9 %	3,3 %	4,3 %	7,6 %
Intel	12.9 %	10,5 %	8,8 %	7,2 %	5,3 %
Red Hat	8.0 %	8,4 %	10,2 %	10,7 %	12,3 %
Linaro	4.0 %	5,6 %	4,1 %	0,7 %	n.a.
Samsung	3.9 %	4,4 %	2,6 %	1,7 %	n.a.
IBM	2,7 %	3,2 %	3,1 %	3,7 %	7,6 %
SUSE	3.2 %	3,0 %	3,5 %	4,3 %	7,6 %
Consultants	2,6 %	2,5 %	1,7 %	2,6 %	2,5 %
Texas Instruments	1.7 %	2,4 %	4,1 %	3,0 %	n.a.
Vision Engraving	1.3 %	2,2 %	2,3 %	n.a.	n.a.
Google	2.0 %	2,1 %	2,4 %	1,5 %	0,9 %
other companies	43,2 %	38,4 %	40,3 %	44,8 %	38,0 %
Intel, Red Hat, Samsung, IBM combined	27,5 %	26,5 %	24,7 %	23,3 %	25,2 %

Sources: Corbet et al. 2009–2015; Corbet & Kroah-Hartman 2016

2.3 Open source as innovation strategy

In particular in the enterprise software markets, which account for more than 80 percent of global software sales, "a widespread use of open-source technology" can now be observed (Driver 2014; Miller & Nelson 2016). In addition, open source solutions are predominating in the area of basic IT infrastructures such as web servers and content management systems (Table 3). Market researchers attribute this not only to the cost advantages but also to the adaptability and "inherent trialability" of open source solutions (Spinellis & Giannikas 2012: 667). As a result, it is not surprising that today most major IT companies (Table 4) are involved in open source projects.

Table 3: Estimated global market share of OSS (in %, installed base / usage share)

	Open source	2010	2016	Competitors	2010	2016
Operating system personal computer (a)	GNU / Linux	1	2	MS Windows Apple Mac OS X	94 5	84 11
Operating system mobile devices (b)	Android	11	72	Apple iOS Symbian / Nokia OS Windows Phone Blackberry	30 33 — 14	20 1 1 > 1
Web browser [desktop] (c)	Mozilla Firefox	31	12	MS IE (+Edge) [Google Chrome] [Apple Safari]	47 14 5	24 59 4
Operating system public servers (d)	Linux (inc. Unix-like)	69	67	MS Windows	31	33
Web server [active sites] (e)	Apache Nginx	72 4	51 32	Microsoft IIS Google Servers LiteSpeed	21 1 1	12 1 2
Web content management system (f)	WordPress Joomla Drupal	51 12 7	59 7 5	Blogger (Google) Bitrix vBulletin	2 — 8	2 1 > 1

Sources: (a, b) NetApplications; (c) StatCounter; (d, e, f) W3techs (Status: 3/2017)

Table 4: Largest public Internet, IT and software companies

	Sales FY 2016, in billion US-$	**Market cap** 5/2016, in billion US-$	**Employees** mid-2016
Apple	215.6	586	115,000
Samsung Electronics	177.6	162	392,305
Amazon	136.0	293	241,000
Alphabet (Google)	90.3	500	73,992
Microsoft	85.3	407	114,000
IBM	79.9	143	380,300
HP Inc.	48.2	22	49,000
Oracle	37.0	169	136,263
Facebook	27.6	315	15,724
SAP	24.1	98	84,183

Sources: Forbes 2000 (Status: 4/2017); Annual reports of the companies

Microsoft—the company, which has long termed open source as an “intellectual property destroyer” (Computerworld 3/2001: 78)—launched its subsidiary MS Open Technologies in 2012. Since then, it has put .NET Framework, software development kits for its cloud computing service Azure as well as many other components under a free license, namely in order “to achieve a strategic objective, such as promoting industry standards, advancing interoperability, or attracting and enabling our external development community” (Microsoft 2015: 13). It would be difficult to estimate what proportion of leading software companies’ R&D budgets goes to open source projects since the integration of open source elements is now standard practice in numerous manufacturer-specific architectures. Apple’s operating system packages macOS, iOS, tvOS and watchOS, for example, are in its core based on the Unix-like operating system Darwin and contain hundreds of other open source software components, e.g., WebKit (browser engine), CUPS (printing system), and XQuartz (window system).

At the turn of the millennium, IBM had already invested several hundred million US dollars in the development of Linux, namely as a means to counteract Microsoft’s dominance in the enterprise sector and to set up a service business around open source software. Today, IBM is involved in well over 100 open source projects, among them the cloud computing platform OpenStack, in which Intel and Hewlett-Packard also participate. However, that involvement results less from idealism than from pragmatic strategizing: “Such actions are comparable to giving away the razor (the code) to sell more razor blades (the related consulting services that IBM and HP hope to provide).” (Lerner 2012: 43) It is for similar reasons that SAP, Oracle, and Adobe are participating in open source projects. In addition, many consumer electronics products from Samsung and other leading companies—such as TVs, tablets, phones, cameras, etc.—are enabled with open source software. For smaller IT providers, in particular, an involvement in open source projects also serves as a “marketing tool to increase brand recognition” (Dahlander & Magnusson 2008: 638).

A special variant of corporate open source exposures is the development of the Android operating system for mobile devices by the Open Handset Alliance, initiated and led by Google. Advertised as a pure open source project and often presented in the literature along the same lines as projects such as the Linux kernel (e.g., Herstatt & Ehls 2015: XVII), the development of the operating system is de facto controlled by Google alone. The Android code is run under permissive licenses, which, in combination with further frameworks such as the “Compatibility Definition Document” (CDD), essentially give Google comprehensive steering control. “Because it fully controls the development of the OS, Google can determine the technological specifications to which Android partners must abide.” (Spreeuwenberg & Poell 2012) With the launch of Android, Google apparently succeeded above all in facilitating the seamless access to its own services and offers for as many IT devices as possible. For example, whereas Google generated approximately 99 percent of its revenue from advertising in 2007, the sale of its digital content and services accounted for 11 percent of sales (US-$ 90 billion) in 2016 (Alphabet 2017).

In addition, the end of the 1990s saw the emergence of a number of “open source companies,” which were giving away their core product, the software code, free of charge while endeavoring to build a business through support services. However, with the exception of the Linux distributor Red Hat, which had been cooperating early on with leading hardware vendors and which today is market leader in enterprise Linux systems, most of the companies that were launched during the dot-com boom quickly folded (Ante 2014; Levine 2014). And although the open source environment has recently given rise to new startups (e.g., Hortonworks *2011), most of these companies do not even emphasize “open source” in their self-presentation and are characterized by a low level of identification with Stallman’s ideals of reciprocity:

> “There is a tension between the GPL [Gnu General Public License] and business which has consequences for what we can do and what we want to do. At the end of the day, the company must earn money to survive. Richard Stallman has a very idealistic view of the world, which is admirable. But if one considers it from a business perspective one realizes that it is not feasible in practice.” (Open source service platform provider, in Bergquist et al. 2012: 8)

Indeed, today it is, in contrast, mainly established corporations such as IBM (“Open Source & Standards are key to making our planet smarter”) or Microsoft (“Openness builds bridges between platforms and people”) that are referring to certain selected maxims of free software in their public relations.

Table 5: Popular projects on OpenHub (catalog for open source projects) Q1/2017

Project	Commits*	Umbrella organization	Primary funding source
Android	104,151	Google, Open Handset Alliance (84+ companies)	
KDE	87,446	KDE e.V.	Patronages (includes Google, SUSE, Canonical)
OpenStack	76,130	OpenStack Foundation	Members (includes HP, IBM, Red Hat, AT&T)
Linux Kernel	73,254	Linux Foundation	Members (includes HP, Intel, IBM, Samsung)
Mozilla Firefox	53,255	Mozilla Foundation	Donations, royalties (includes Google, Yahoo)
Ubuntu (includes Touch)	52,128	Canonical Ltd.	Canonical, partners (Amazon, Intel, Asus etc.)
Debian Linux	** 26,782	Debian Project	Donations, partners (includes HP, 1&1)
LibreOffice	15,733	The Document Foundation	Donations (includes Google, Red Hat, Intel)
GNU Compiler Collection	7,602	Free Software Foundation	Members, patronages (includes IBM, HP)
Apache HTTP Server	2,103	Apache Foundation	Donations (Google, Microsoft, Facebook, etc.)

*Source: OpenHub (4/2017); *Code Updates Q1/2016–Q1/2017 (** 2015–2016)*

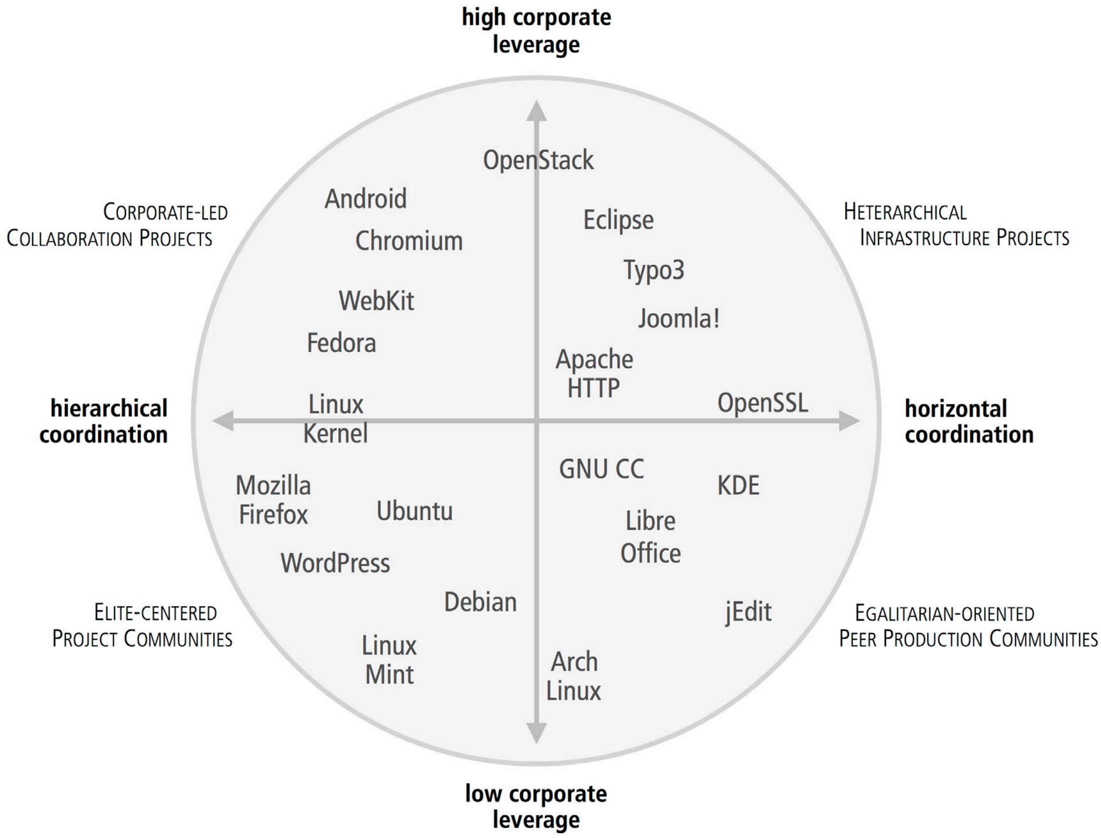
In that sense, many popular open source communities of today have close financial ties with leading IT companies, which are deliberately investing in open source projects as part of their overarching innovation strategies (Table 5). In the case of corporate-initiated projects (such as Android or OpenStack), this entanglement is obvious. However, even foundation-supported communities (such as the Apache HTTP Server or GNU) grant their donors seats on the boards of their umbrella organizations. The latter, while not directly in control of the development activities, nevertheless provide the technical infrastructures and distribute financial resources. Together with their involvement in the code development as such, these leading IT companies are thereby securing a considerable influence on relevant development projects while at the same time allowing for greater predictability in planning for these projects as regards both their human and financial resources.

3 Varieties of open source software projects

Over the last 20 years, open source software has thus become an integral part of the IT industry. Against this background, the array of open source projects has become larger and broader. At one end of the spectrum, some communities are still committed to Stallman's socio-ethical ideals, operate independently of corporate interests, and are largely aligned with egalitarian organizational principles. At the other end of the spectrum, we find a large number of projects that are under the direct control of leading IT companies and that follow hierarchical development models. Based on existing case studies and empirical data (e.g., licensing documents, certificates, technical specifications, membership listings, self-descriptions, mailing lists, wikis) four ideal-type variants of recent open source projects can be distinguished according to their prevailing forms of coordination and the degree of corporate involvement (Fig. 1).

Corporate-led collaboration projects are characterized by clear work hierarchies and a strong market presence of its products. Their communities are composed primarily of programmers who are employed by the participating companies. In Android, WebKit (rendering engine for web browser) and Fedora (Linux distribution), the strategic control clearly lies with Google, Apple, and Red Hat, respectively. In the cloud computing project OpenStack, big sponsors (e.g., Rackspace, Intel, IBM, Red Hat, AT&T, Deutsche Telekom, Cisco) likewise have considerable influence: "This new kind of community […] is clearly driven by corporate interests. Participating companies, which may be commercial competitors, have clear strategies towards the project […]." (Gonzalez-Barahona et al. 2013: 39) Moreover, when this type of corporate collaboration takes place under the terms of open source projects, it allows to overcome two classic knowledge-sharing dilemmas (Larsson et al. 1998; Cabrera & Cabrera 2002): One, open source licenses prevent the proprietarization of the collectively developed code by any individual entity and, secondly, these same licenses prevent abuse from free riders given the traceability of which companies use which elements and whether they participated in the development (Henkel et al. 2014; Sydow et al. 2016: 233–252). In addition, in this day and age, it is often easier to create new software products by building on already existing open source elements than by developing the software from scratch (West & Bogers 2014, 2017).

Figure 1: Varieties of open source software projects



Heterarchical infrastructure projects, whose products are ever-present beneath the visible surface of IT architectures, are closely intertwined with corporate contexts. Some were initially based (like the integrated development environment Eclipse) on architectures that were formerly proprietary. Others were (like the Apache HTTP Server) characterized by rapid organic growth in their beginnings, since they offered solutions to previously unaddressed challenges, making them interesting to companies early on, especially as open infrastructures do not carry an impetus for application code or hardware to be open itself (Weinberg 2015).[2] Today, infrastructure pro-

[2] As Weinberg (2015b) notes, the mobile operation system Android therefore “provides an apt analogy. While the platform derives from hundreds of open source components […], the majority of the applications distributed through Google Play are closed and proprietary.”

jects are primarily supported by medium and large IT companies; however, their communities are not guided by corporate core circles but mostly operate under the umbrella of nonprofit organizations or foundations and are structured horizontally along working groups. Management positions are assigned on a meritocratic basis ("the more you contribute, the more responsibility you will earn"), but in these projects, too, employed developers, who are explicitly freed from other tasks by their companies to work in the community, are more likely than leisure or lay programmers to advance to decision-making positions. An infrastructure project that points to the potential risks of the open source model is the encryption software OpenSSL, which is used in many operating systems and platforms since the 1990s: Until 2014, OpenSSL was developed by one full-time programmer assisted by a very small, voluntary team and received little financial support from the industry. In that context, ever new features were continually integrated into OpenSSL—yet without bolstering the level of maintenance work accordingly. In 2012, then, this culminated in an oversight that led to the major "Heartbleed" vulnerability, which was not discovered until 2014 (Stokel-Walker 2014; Perlroth 2014).

Elite-centric project communities are likewise based to a large extent on the contributions of developers who are affiliated with companies, but these contributors are not under the direct control of a commercial actor. Rather, their coordination takes place along differentiated decision-making pyramids, or a "lieutenant system built around a chain of trust" (Kernel.Org 2016) that is often headed by its founder as a "benevolent dictator" (e.g., Linux kernel, WordPress, Ubuntu—"Shuttleworth, as self-appointed benevolent dictator for life […], plays a happily undemocratic role as sponsor of the project"), a democratically elected project manager (e.g., Debian Linux), or a long-term management team (e.g., Mozilla): "The ultimate decision-maker(s) are trusted members of the community who have the final say in the case of disputes. This is a model followed by many successful open source projects […]." (Mozilla Foundation 2017) While this top-down management curtails the scope of the participants, it also counteracts fragmentation (Coleman 2013; Snow 2014; O'Mahony & Ferraro 2007). In that sense, Linux Mint initiator Clement Lefebvre (in Byfield 2013) states: "The final decision comes from the top […]. Strong leadership is important and benefits Linux Mint, [because] the decisions we take remain consistent and are coherent with our overall vision." In Debian or Mozilla, the project guidelines are formally fixed; in the Linux kernel project, by contrast, Torvald's leadership style gave rise to "opaque governing norms" that risk counteracting the openness of the project in the event of a conflict: "[…] without the law or a clear mechanism of accountability those injured by or excluded from peer production processes have very limited recourse. The only alternative for these individuals is to not participate." (Kreiss et al. 2011: 252)

Egalitarian-oriented peer production communities are, based on their self-understanding, about market-independent, intrinsic and equitable collaboration among volunteers. "Basically, people who participate in peer production communities love it. They feel passionate about their particular area of expertise and revel in creating something new or better." (Tapscott &Williams 2006: 70) However, as is apparent from KDE (a community for desktop environments), GNU or LibreOffice, once these communities reach a certain size, they usually feature more classical leadership structures and a stable roster of corporate stakeholders. For instance, the KDE project does not have a single project manager but "The KDE Core Team" consisting of several dozen contributors that decide on the overall direction of the KDE platform (KDE 2017); the GNU Compiler Collection is managed by the "GCC steering committee […] with the intent of preventing any particular individual, group or organization from getting control over the project" (GNU 2017). By contrast, intrinsically motivated communities such as Arch (Linux distribution) or jEdit (text editor) target their products to very specific user groups, are rather irrelevant to the general market and are run by small teams, due to which they have so far been able to do without pronounced social structures or membership rules ("You can 'join' simply by subscribing to the […] mailing lists"). Still, even smaller developer communities are marked by technical and social contribution barriers, "including steep learning curve, lack of community support, and difficulties finding out how to start" (Steinmacher et al. 2015: 1380); and when such communities grow, alongside the intensity of their interactions with external market actors, they too tend to adopt "cathedral-like" organizational modes, regardless of any level of technical efficiency they may have attained.

Table 6: Ideal-type manifestations of open source projects

	Corporate-led collaboration projects	**Elite-centered project communities**	**Heterarchical infrastructure projects**	**Egalitarian-oriented peer production communities**
	e.g., Android, WebKit, OpenStack	*e.g., Linux Kernel, Firefox, Ubuntu*	*e.g., Apache HTTP, Eclipse, Joomla!*	*e.g., GNU CC, Arch Linux, KDE*
Work organization	Mainly hierarchical	Mainly hierarchical	Horizontal – meritocratic	Horizontal – egalitarian
Strategic management	Individual companies / consortium of firms	Project founder / long-term project management team	Board of directors of the foundation / Steering group	Steering committee / core team
Funding	Participating companies	Corporate donations / smaller private donations	Primarily contributions from companies	Primarily smaller private donations
Participant pool	Mainly staff from the participating companies	Employed and (few) voluntary developers	Employed developers, and explicit company representatives	Primarily voluntary developers

The common denominator of all four project variants is their underlying open source licensing models, which protect their products from direct proprietarization. Nonetheless, the "rebel code" spirit (Moody 2002) of these projects has, at this point, been significantly diluted. Despite the enhanced technological possibilities for coordination, all larger open source projects give rise to hierarchical decision-making routines as well as distinct management circles and tend to become enmeshed in market contexts if they operate in the long term and are able to reach larger target groups or to provide comprehensively used IT infrastructures (Table 6). Contrary to the notion that "organizations […] really don't matter as much as they used to" (Suddaby 2013: 1009), conventional companies and non-for-profits are not losing their influence over open source projects and are instead maintaining their status and role as the initiators and funders of open collaboration projects in the software industry.

4 Open source projects as incubators of innovation

The preceding chapters debunk two long-prevailing assumptions: One, that the technical infrastructures of the internet, seen to promote decentralized working methods and to offer "easier pathways to challenge oligarchy," can, on their own, effectively resist an "ossification of power" in open source projects (Benkler 2013: 225). And two, that there is a "networked information economy" (Benkler 2006: 3) in which corporate actors (companies, non-governmental organizations, research institutes) suffer a loss of their relevance in the face of "nonproprietary, voluntaristic, self-assisted practices" (Benkler 2013: 213). These assumptions do not hold for two main reasons.

First, although the infrastructures and services used in open source projects lay the foundation for the work processes and the optimized coordination of tasks, they in no way lead to a disintermediation or loss of relevance of *social structuring patterns*. In other words, in open source communities and similar web-based communities (such as Wikipedia), too, collectively accepted rules, guidelines and hierarchical decision-making structures emerge that are characterized by strong power asymmetries. Indeed, such social institutionalization dynamics are a fundamental requirement for an open source project to be perceived as an entity (by the project developers themselves as well as by external actors), to be capable of intentional and strategic action, and to gain broader momentum (Dolata & Schrape 2016; O'Mahony & Ferraro 2007).

Secondly, corporate players usually have more leverage than communities of interest to act systematically and reliably, namely because they have *formalized decision-making routines* as well as the discretion to utilize their resources regardless of their members' preferences (see, e.g., Perrow 1991; Blau & Scott 1962). In addition, companies and other organizations are able to bring in their resources more continuously and consistently than individual contributors. As a result, for-profit and not-for-profit organizations significantly contribute to creating a reliable, predictable planning environment for open source projects, in turn garnering them considerable clout and influence over the community. Moreover, independent projects are often linked to associated non-profit organizations that offer them an umbrella identity and that stabilize the community in the event of conflicts (Ahrne, Brunsson & Seidl 2016).

In that context, open source projects could be seen to be subject not only to corporatization but also to a steadily intensifying embracement by the established software providers and IT companies. Indeed, the reconstruction of the history of open source software development presented above shows that the commonly portrayed ideal image of an independent commons-based peer production existed primarily in the early days of free software. However, as early as the end of the 1990s, the then internet-focused start-up scene relied heavily on inexpensively licensed open source components, followed by, starting with the turn of the millennium, the increasing involvement of other companies in open source projects—IBM being an early case in point. From the point of view of innovation research, such a development does not necessarily seem unusual: Like other (radical) niche innovations, free software projects were initially "carried and developed by small networks of dedicated actors, often outsiders or fringe actors," yet became subject to a professionalization and appropriation on the part of established economic actors as soon as they caught the attention of the mainstream markets (Geels & Schot 2007: 400; Dolata 2013: 68f.).

Table 7: Historical episodes of collective invention

Episode	Knowledge exchange	Outcome
The Cornish Pumping Engine ca. 1810–1850, Cornwall, England (Nuvolari 2004)	Exchange of technical data and know-how; comparison of individual progress via journals	Development of a fuel-efficient steaming engine for the mining industry
Paper manufacture ca. 1827–1857, New England, USA (McGaw 1987)	Stable community of mill owners; regular informal exchange of experiences	Increase in productivity by mechanization of the entire production process
Furnace technologies ca. 1850–1880, Cleveland District, England (Allen 1983)	Exchange of technical knowledge via journals and societies; collective trial-and-error process	Reduction of energy demand by raising the building heights and temperature adjustments
Flat-panel displays ca. 1969–1989, Japan / Europe / USA (Spencer 2003)	Publication of proprietary research results in technical journals	Incremental development in the pre-commercial phase
Homebrew Computer Club ca. 1975–1978 [1986], Silicon Valley, USA (Meyer 2003)	Free exchange of technical information until the success of participating firms (e.g., Apple Inc.)	Development of the first personal computers for the mass market

History has seen many episodes of collective invention (Table 7) during which corporate or individual actors openly shared their knowledge in niches that were decoupled from the general market, thereby benefiting from "cumulative advance" (Allen 1983; Powell & Giannella 2010; Lamoreaux & Sokoloff 2000; Scranton 1997):

> "To the degree that economists have considered this behaviour at all, it has been regarded as an undesired 'leakage' that reduces the incentives to invent. That firms desire such behaviour and that it increases the rate of invention […] are possibilities not yet explored." (Allen 1983: 21)

However, in sharp contrast to former cases of collective invention, open source software projects remain viable beyond the initial stages of the innovation process, i.e. beyond the emergence of predominant solutions and their commercial exploitation (Osterloh & Rota 2007). This may be attributed to the following interacting factors:

- Early on in its development, the free software scene gave rise to informal rules as well as *resilient licensing models* designed to prevent the proprietarization and commodification of collective work results. Today, these models comprise the core framework of open source software projects, allowing for a reliable project-specific exchange of knowledge and ensuring that collaboration between individual developers as well as companies that may be direct competitors otherwise (e.g., Apple and Samsung) can take place outside of formal cooperations.

- At the same time, the rapid *advance of online technologies* has allowed for much greater efficiency in the verification of compliance with these licensing conditions and has facilitated not only the access to projects but also the spread and use of their products. In addition, it has contributed to solving a long-term problem faced by the software sector, namely that of coordinating large projects with developers working in different contexts and from different geographical locations (Brooks 1975; Campbell-Kelly 2003).

- Finally, and most importantly, in a software industry that has been expanding for decades[3] and that is characterized by very short innovation cycles, open source projects have proven to be pivotal *incubators for branch-defining infrastructures*, standards and product platforms (such as the Apache HTTP Server, the Linux kernel, or OpenStack). This applies all the more since open source software can be tested by the developers themselves and adapted to their respective requirements with little administrative effort (Spinellis & Giannikas 2012).

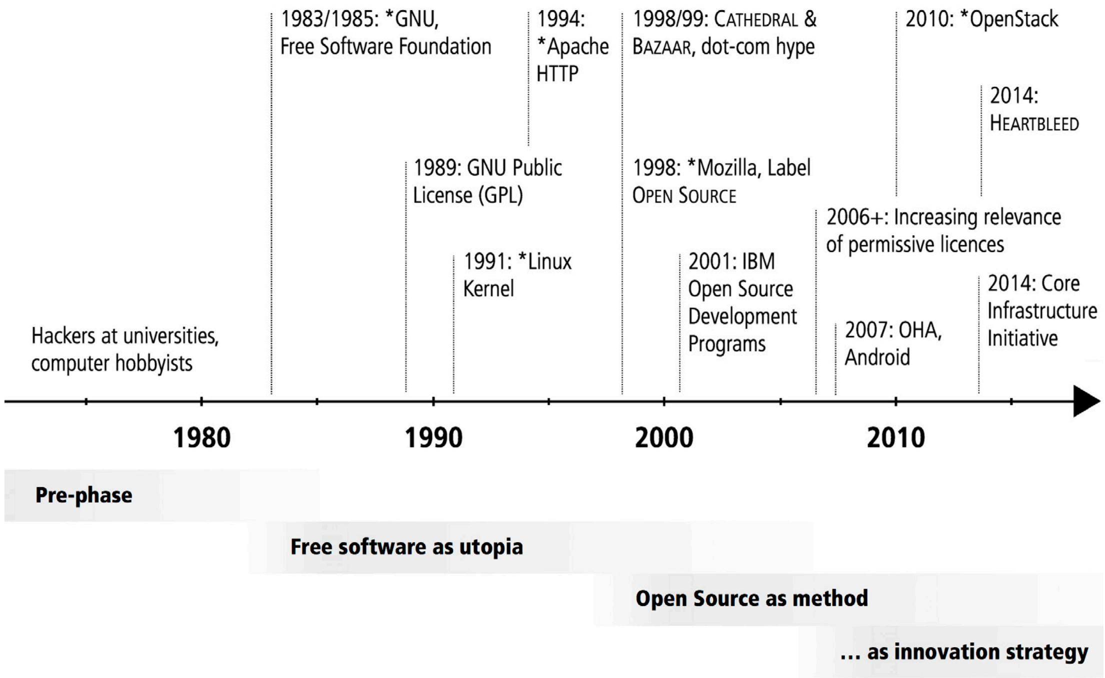
From the 1980s on, the ongoing success of open source projects was determined by the expansion of digital connectivity, the establishment of widely accepted informal rules and conventions for work and practice and, last but not least, the creation of reliable licensing models. Strongly protective "copyleft" licenses (e.g., the GNU Gen-

---

[3] Worldwide spending for software and IT services 2005: 885 billion US-Dollars; 2010: 1,092 billion US-Dollar; 2015: 1,532 billion US-Dollar (UNCTAD 2012; Accelerance 2017).

eral Public License) and their derivatives (e.g., permissive licenses such as the MIT License or the Apache License 2.0), aided by the online technologies, have contributed to ensuring the longevity of collective invention through socio-technical means. Thus, at the turn of the millennium, a novel form of collaboration that initially took place in subversive niches was adapted into a supplementary working method by the commercial software industry and is today a key element of the innovation strategies of all established IT providers (Fig. 2). As Dolata (2017: 21f.) notes, most of these corporations "continue to be characterized by a strong focus on proprietary developments," but are now "perceptive to their environments […]" and "observe very closely what is happening in the open source communities; collaborate in open source projects;" and "revert to […] software and know-how developed there […]."

Figure 2: Open source as utopia, method, and innovation strategy



In that sense, open source licensing models no longer can be seen as a "form of institutional jiujitsu" (Benkler 2002: 446) that aims for the total dissolution of intellectual property rights (see, for an incisive critical overview, Coleman 2013: 185–215). Instead, these licenses comprise the legal and structural basis of collaboration projects that, as incubators of innovation, do not compete against established forms of socio-economic production but instead complement and expand these (Allen & Potts 2016). Since open source software (and hardware) projects have proven to be more than just a flash in the pan, it may be worthwhile to recognize them as fourth enduring source of invention and technological change (cf. Jewkes et al. 1969; Allen 1983) —in addition to individual inventors, not-for-profit organizations (e.g., universities, public research centers), and the research and development departments of private firms.

5 Conclusions

Overall, the relationship between the task-oriented working structures of open source projects and the established forms of economic coordination is not characterized by competition but by complementariness. While open source software projects have largely lost their formatting as a counterpart to commercial production, they have, unlike previous types of collective invention, remained viable beyond the initial phase of innovation processes. Open source licenses, together with the coordination-facilitating features of the online technologies, have established the socio-technical framework for a permanent form of open and collaborative development; initially, it was applied in subversive niches and later adapted by the commercial software industry as a complementary development method. Today, open source projects have become important incubators for new product lines, standards and fundamental infrastructures in an international industry characterized by very short innovation cycles, as well as a fixed component of the innovation strategies of established IT companies.

In the last two decades, open source software projects thus have contributed to more flexibility in the collaboration between developers from divergent contexts, the project-specific cooperation between market actors, as well as inner-organizational production modi—through which the software market as a whole has become more permeable. At the same time, however, freely available source code does not necessarily result in more transparent coordination patterns than in other working contexts, in a disintermediation in the established modes of resource and power distribution over the long term, or a comprehensive democratization of innovation processes. In sum, open code alone does not guarantee open societal structures.

Therefore, the prospect that the original concept of commons-based peer production, which was rarely applied as such even in open source communities, could be adapted to neighboring socio-economic fields such as 3D printing (e.g., Rifkin 2014) and socio-political phenomena such as social movements (e.g., Bennett et al. 2014) remains at best misleading. Worse, these types of narratives deflect from the fact that some trends engendered by the digital transformation of society are not necessarily compatible with the ideal of a more open and democratic economy. We think only of the potential erosion of “the foundations of the system of work and labour regulation as it has developed historically, both on the company and on the society level” (Boes et al. 2017: 143), the restriction of fundamental consumer rights through the terms and conditions imposed on the users of many online services, or the global hegemony of a small number of multinational companies over the key infrastructures of communication, media distribution and information retrieval to a degree unprecedented in media history (Schrape 2017: 145–147; Dolata 2017).[4]

Against this background, social scientists would do well to scrutinize popular catchwords such as *Open Innovation*, *Web 2.0* or *Open Source,* often coined by professional "visioneers" in Silicon Valley or other high-tech hubs (McCray 2013), before adopting them at once as quasi-sociological terms. Instead, efforts should be made to examine to what degree the expectations associated with these terms might point to reoccurring semantic patterns and to assess their societal impacts. For example, even though the "narratives of openness and individual empowerment" (Ames et al. 2014: 1088), which are associated with open source software projects and more recent phenomena such as the "maker culture," have not yet been brought to fruition as intended, they nevertheless fulfill elementary functions in their communicative contexts. Namely, they draw attention to new (technological) development paths, channel the discourse, contribute to the creation of innovation niches, and serve as a legitimizing basis in economic or political decision-making processes as well as they enhance the internal cohesiveness of the respective fields. In this respect, the themed openness narratives could indeed qualify as "productive types of communication," provided they are not—as was done for a long time by observers of open source software projects—misunderstood as descriptions of empirical facts (Dickel & Schrape 2017: 172).

4 Furthermore, as vendor lock-in is still attractive to vendors, many IT companies are practicing one or another kind of "openwashing" for marketing purposes (Pomerantz & Peek 2016): "Openwashing describes situations where the term 'open' as a (generally positive) adjective actually obscures the fact that content, processes, platforms or institutions are in reality not 'open' or at least not in the ways others think they should be." (Smith & Seward 2017)

References

Accelerance Inc., 2017: *Global IT Market Size: Facts and Figures*. http://www.accelerance.com/research/global-it-market-size-facts-and-figures (5/2017).

Ahrne, Göran / Brunsson, Nils / Seidl, David, 2016: Resurrecting Organization by Going Beyond Organizations. In: *European Management Journal* 34(2), 93–101.

Allen, Darcy W.E. / Potts, Jason, 2016: How Innovation Commons Contribute to Discovering and Developing New Technologies. In: *International Journal of the Commons* 10(2). https://thecommonsjournal.org/articles/10.18352/ijc.644/ (5/2017).

Allen, Robert C., 1983: Collective Invention. In: *Journal of Economic Behavior & Organization* 4(1), 1–24.

Alphabet Inc., 2016: *Form 10-K 2016*. https://abc.xyz/investor/ (5/2017).

Ames, Morgan / Bardzell, Jeffrey / Bardzell, Shaowen / Lindtner, Silvia / Mellis, David / Rosner, Daniela, 2014: Making Cultures: Empowerment, Participation, and Democracy–or not? In: *Proceedings of the 32nd ACM Conference on Human Factors in Computing Systems*, 1087–1092.

Ante, Spencer, 2014: Red Hat Plays Hardball on OpenStack Software. In: *The Wall Street Journal* 5/13/2014. http://on.wsj.com/14qBpus (05/2017).

Arrow, Kenneth J., 1962: Economic Welfare and the Allocation of Resources for Invention. In: Nelson, Richard (Hg.): *The Rate and Direction of Inventive Activity. Economic and Social Factors*. Princeton: Princeton University Press, 609–626.

Benkler, Yochai, 2002: Coase's Penguin, or, Linux and 'The Nature of the Firm.' In: *Yale Law Journal* 112, 369–446.

Benkler, Yochai, 2004: Intellectual Property: Commons-based Strategies and the Problems of Patents. In: *Science* 305(5687), 1110–1011.

Benkler, Yochai, 2006: *The Wealth of Networks. How Social Production Transforms Markets and Freedom*. New Haven: Yale University Press.

Benkler, Yochai, 2013: Practical Anarchism, Peer Mutualism, Market Power, and the Fallible State. In: *Politics & Society* 41(2), 213–251.

Benkler, Yochai / Nissenbaum, Helen, 2006: Commons-based Peer Production and Virtue. In: *Journal of Political Philosophy* 14(4), 394–419.

Bennett, W. Lance / Segerberg, Alexandra / Walker, Shawn, 2014: Organization in the Crowd: Peer Production in Large-scale Networked Protests. In: *Information, Communication & Society* 17(2), 232–260.

Bennett, W. Lance / Segerberg, Alexandra, 2015: The logic of connective action: digital media and the personalization of contentious politics. In: Coleman, Stephen / Freelon, Deen (Eds.): *Handbook of Digital Politics*. Cheltenham: Edward Elgar, 169–198.

Bergquist, Magnus / Ljungberg, Jan / Rolandsson, Bertil, 2012: *Justifying the Value of Open Source*. ECIS Proceedings. http://aisel.aisnet.org/ecis2012/122/ (5/2017).

Bezroukov, Nikolai, 1999: A Second Look at the Cathedral and the Bazaar. In: *First Monday* 4(12). http://firstmonday.org/article/view/708/618 (5/2017).

Blau, Peter M. & Scott, Richard W., 1962: *Formal Organizations. A Comparative Approach*. San Francisco: Chandler.

Boes, Andreas / Kämpf, Tobias / Langes, Barbara / Lühr, Thomas / Ziegler, Alexander, 2017: Cloud & Crowd: New Challenges for Labour in the Digital Society. In: *tripleC: Communication, Capitalism & Critique* 15(1), 132–147.

Brooks, Frederick, 1975: *The Mythical Man-Month*. Reading: Addison-Wesley.

Bulajewski, Mike, 2011: The Peer Production Illusion, Part I. In: *MrTeaCup* 11/19/2011. http://www.mrteacup.org/post/peer-production-illusion-part-1.html (5/2017).

Burton, Grad, 2002: A Personal Recollection: IBM"s Unbundling of Software and Services. In: *IEEE Annals of the History of Computing* 24(3), 64–71.

Byfield, Bruce, 2013: What Makes for a Community Distribution? In: *Linux Magazine* 7/14/2013. http://www.linux-magazine.com/Online/Blogs/Off-the-Beat-Bruce-Byfield-s-Blog/What-makes-for-a-community-distribution (5/2017).

Cabrera, Angel / Cabrera, Elizabeth F., 2002: Knowledge-Sharing Dilemmas. In: *Organization Studies* 23(5), 687–710.

Campbell-Kelly, Martin, 2003: *From Airline Reservations to Sonic the Hedgehog. A History of the Software Industry*. Boston: MIT Press.

Coleman, Gabriella, 2013: *Coding Freedom. The Ethics and Aesthetics of Hacking*. Princeton: Princeton University Press.

Connell, Chuck, 2000: *Open Source Projects Manage Themselves? Dream On*. IBM/Lotus Developers Network. http://www.chc-3.com/pub/manage_themselves.htm (5/2017).

Corbet, Jonathan / Kroah-Hartman, Greg, 2016: *Linux Kernel Development Report*. San Francisco: The Linux Foundation.

Corbet, Jonathan / Kroah-Hartman, Greg / McPherson, Amanda, 2009–2015: *Linux Kernel Development Report*. San Francisco: The Linux Foundation.

Dahlander, Linus / Magnusson, Mats, 2008: How do Firms Make Use of Open Source Communities? In: *Long Range Planning* 41(6), 629–649.

Dickel, Sascha / Schrape, Jan-Felix, 2017: The Logic of Digital Utopianism. In: *Nano Ethics* 11(1), 47–58.

Dolata, Ulrich, 2013: *The Transformative Capacity of New Technologies. A Theory of Sociotechnical Change*. London: Routledge.

Dolata, Ulrich, 2017: *Apple, Amazon, Google, Facebook, Microsoft. Market Concentration – Competition – Innovation Strategies*. Research Contributions to Organizational Sociology and Innovation Studies 2017-01.

Dolata, Ulrich / Schrape, Jan-Felix, 2016: Masses, Crowds, Communities, Movements: Collective Action in the Internet Age. In: *Social Movement Studies* 15(1), 1–18.

Driver, Mark, 2014: *Within the Enterprise, Open Source Must Coexist in a Hybrid IT Portfolio*. Gartner Inc. Research Report. Stamford: Gartner Inc.

Fisher, Franklin M. / McKie, James W. / Mancke, Richard B., 1983: *IBM and the US Data Processing Industry. An Economic History*. Santa Barbara: Praeger.

Fitzgerald, Brian, 2006: The Transformation of Open Source Software. In: *MIS Quarterly* 30(3), 587–598.

Free Software Foundation, 1989: *GNU General Public License (GPL) Version 1.0*. http://www.gnu.org/licenses/old-licenses/gpl-1.0.en.html (5/2017).

Gates, Bill, 1976: An Open Letter to Hobbyists. In: *Computer Notes* 1(9), 3.

Geels, Frank W. / Schot, Johan W., 2007: Typology of Sociotechnical Transition Pathways. In: *Research Policy* 36(3), 399–417.

Gelsi, Steve, 1999: VA Linux Rockets 698%. In: *CBS Marketwatch* 12/10/1999. http://www.cbsnews.com/news/va-linux-rockets-698/ (5/2017).

GNU Project, 2017: *GCC Steering Committee*. https://gcc.gnu.org/steering.html (5/2017).

Gonzalez-Barahona, Jesus M. / Izquierdo-Cortazar, Daniel / Maffulli, Stefano, 2013: Understanding How Companies Interact with Free Software Communities. In: *IEEE Software* 30(5), 38–45.

Greenstein, Shane / Nagle, Frank, 2014: Digital Dark Matter and the Economic Contribution of Apache. In: *Research Policy* 43(4), 623–631.

Gulley, Ned / Lakhani, Karim, 2010: *The Determinants of Individual Performance and Collective Value in Private-collective Software Innovation*. Harvard BS TOMU Working Paper 10/065.

Henkel, Joachim / Schöberl, Simone / Alexy, Oliver, 2014: The Emergence of Openness: How and Why Firms Adopt Selective Revealing in Open Innovation. In: *Research Policy* 43(5), 879–890.

Herstatt, Cornelius / Ehls, Daniel, 2015: *Open Source Innovation: Phenomenon, Participant Behaviour, Business Implications*. New York: Routledge.

Holtgrewe, Ursula / Werle, Raymund, 2001: De-Commodifying Software? Open Source Software Between Business Strategy and Social Movement. In: *Science Studies* 14(2), 43–65.

Jaeger, Till, 2010: Enforcement of the GNU GPL in Germany and Europe. In: *Journal of Intellectual Property, Information Technology and E-Commerce Law* 1(1), 34–39.

Jewkes, John / Sawyers, David / Stillerman, Richard, 1969: *The Sources of Invention (2)*. London: Palgrave Macmillan.

KDE Project, 2017: *Project Management*. https://www.kde.org/community/whatiskde/management.php (5/2017).

Kernel.Org, 2016: *How to Get Your Change Into the Linux Kernel*. https://www.kernel.org/doc/Documentation/process/submitting-patches.rst (5/2017).

Kolassa, Carsten / Riehle, Dirk / Riemer, Philip / Schmidt, Michael, 2014: Paid vs. Volunteer Work in Open Source. In: *Proceedings 47th Hawaii Int. Conference on System Sciences*, 3286–3295.

Kostakis, Vasilis / Niaros, Vasilis / Giotitsas, Christos, 2015: Production and Governance in Hackerspaces: A Manifestation of Commons-based Peer Production in the Physical Realm? In: *International Journal of Cultural Studies* 18(5), 555–573.

Kreiss, Daniel / Finn, Megan / Turner, Fred, 2011: The Limits of Peer Production: Some Reminders from Max Weber for the Network Society. In: *New Media & Society* 13(2), 243–259.

Lakhani, Karim R. / Hippel, Eric von, 2003: How Open Source Software Works. In: *Research Policy* 32(6), 923–943.

Lamoreaux, Naomi R. / Sokoloff, Kenneth L., 2000: The Geography of Invention in the American Glass Industry 1870–1925. In: *Journal of Economic History* 60(3), 700–729.

Larsson, Rikard / Bengtsson, Lars / Henriksson, Kristina / Sparks Judith, 1998: The Interorganizational Learning Dilemma: Collective Knowledge Development in Strategic Alliances. In: *Organization Science* 9(3), 285–305.

Lerner, Joshua, 2012: *The Architecture of Innovation*. Boston: Harvard Business Press.

Lerner, Joshua / Schankerman, Mark, 2010: *The Comingled Code. Open Source and Economic Development*. Cambridge: MIT Press.

Lerner, Joshua / Tirole, Jean, 2002: Some Simple Economics of Open Source. In: *Journal of Industrial Economics* 50(2), 197–234.

Lerner, Joshua / Tirole, Jean, 2005: The Scope of Open Source Licensing. In: *Journal of Law, Economics & Organization* 21(1), 20–56.

Lessig, Lawrence, 1999: Open Code and Open Societies. In: *Chicago Kent Law Review* 74, 1405–1420.

Levy, Steven, 1984: *Hackers. Heroes of the Computer Revolution*. Garden City: Anchor Press.

Lewine, Peter, 2014: Why There Will Never Be Another Red Hat: The Economics Of Open Source. In: *Techcrunch* 2/13/2014. http://tcrn.ch/1bs5yMQ (5/2017).

Lowood, Henry, 2009: Videogames in Computer Space: The Complex History of Pong. In: *IEEE Annals of the History of Computing* 31(3): 5–19.

Martin, James, 1991: *Rapid Application Development*. Indianapolis: Macmillan.

McCray, Patrick W., 2013: *The Visioneers. How a Group of Elite Scientists Pursued Space Colonies, Nanotechnologies, and a Limitless Future*. Princeton: Princeton University Press.

McGaw, Judith A., 1987: *Most Wonderful Machine: Mechanization and Social Change in Berkshire Paper Making 1801–1885*. Princeton: Princeton University Press.

Menell, Peter S., 2002: Envisioning Copyright Law's Digital Future. In: *New York Law School Review* 46, 63–199.

Meyer, Peter B., 2003: *Episodes of Collective Invention*. BLS Working Paper 368. Washington: U.S. Bureau of Labor Statistics.

Microsoft Inc., 2015: *2014 Annual Report*. http://www.microsoft.com/investor/reports/ (5/2017).

Miller, Paul / Nelson, Lauren E., 2016: *Open Source Powers Enterprise Digital Transformation*. Research Report. Cambridge: Forrester Inc.

Moody, Glyn, 2002: *Rebel Code. The Inside Story of Linux and the Open Source Revolution*. New York: Basic Books.

Mozilla Foundation, 2015: *Mozilla Roles and Leadership*. https://www.mozilla.org/en-US/about/governance/roles/ (5/2017).

Netscape Communications, 1998: *Netscape Announces Mozilla.org*. Press Release 2/23/1998.

Nuvolari, Alessandro, 2004: Collective Invention during the British Industrial Revolution: The Case of the Cornish Pumping Engine. In: *Cambridge Journal of Economics* 28(3), 347–363.

O'Mahony, Siobhán, 2003: Guarding the Commons. How Community Managed Software Projects Protect their Work. In: *Research Policy* 32(7), 1179–1198.

O'Mahony, Siobhán / Ferraro, Fabrizio, 2007: The Emergence of Governance in an Open Source Community. In: *Academy of Management Journal* 50(5), 1079–1106.

O'Reilly, Tim, 2000: *Re: Open Source and OpenGL*. In: Ask Tim Forum 12/15/2000. http://archive.oreilly.com/pub/a/oreilly/ask_tim/2000/opengl_1200.html (5/2017).

Osterloh, Margit / Rota, Sandra, 2007: Open Source Software Development: Just Another Case of Collective Invention? In: *Research Policy* 36(2), 157–171.

Perlroth, Nicole, 2014: Heartbleed Highlights a Contradiction in the Web. In: *The New York Times* 4/18/2014. http://nyti.ms/1hb6uBd (5/2017).

Perrow, Charles, 1991: A Society of Organizations. In: *Theory & Society* 20, 725–762.

Pomerantz, Jeffrey / Peek, Robin, 2016: Fifty Shades of Open. In: *First Monday* 21(5). http://dx.doi.org/10.5210/fm.v21i5.6360 (5/2017).

Powell, Walter W. / Giannella, Eric, 2010: Collective Invention and Inventor Networks. In: Hall, Bronwyn H. / Rosenberg, Nathan (Eds.): *Handbook of the Economics of Innovation Volume 1*. Oxford: Elsevier, 575–605.

Raymond, Eric S., 1998: *Goodbye, "free software"; Hello, "open source"*. Announcement 11/22/1998. ftp://ftp.lab.unb.br/pub/computing/museum/esr/open-source.html (5/2017).

Raymond, Eric S., 1999: *The Cathedral and the Bazaar. Musings on Linux and Open Source by an Accidental Revolutionary*. Sebastopol: O'Reilly.

Rifkin, Jeremy, 2014: *The Zero Marginal Cost Society*. New York: Palgrave Macmillan.

Romer, Paul, 1990: Endogenous Technological Change. In: *Journal of Political Economy* 98(5), 71–102.

Schrape, Jan-Felix, 2017: Reciprocal Irritations: Social Media, Mass Media and the Public Sphere. In: Paul, Regine / Mölders, Marc / Bora, Alfons / Huber, Michael / Münte, Peter (Hg.): *Society, Regulation and Governance: New Modes of Shaping Social Change?* Cheltenham: Edward Elgar Publishing, 138–149.

Scranton, Philip, 1997: *Endless Novelty: Specialty Production and American Industrialization 1865–1925*. Princeton: Princeton University Press.

Smith, Longshore / Seward, Ruhiya, 2017: Openness as Social Praxis. In: *First Monday* 22(4). https://firstmonday.org/ojs/index.php/fm/article/view/7073/6087 (5/2017).

Snow, Shane, 2014: How Matt's Machine Works. In: *Fast Company* 9/11/2014. http://www.fastcompany.com/3035463/how-matts-machine-works (5/2017).

Spencer, Jennifer W., 2003: Firms' Knowledge-sharing Strategies in the Global Innovation System. Evidence from the Flat Panel Display Industry. In: *Strategic Management Journal* 24(3), 217–233.

Spinellis, Diomidis / Giannikas, Vaggelis, 2012: Organizational Adoption of Open Source Software. In: *Journal of Systems and Software* 85(3), 666–682.

Spreeuwenberg, Kimberley / Poell, Thomas, 2012: Android and the Political Economy of the Mobile Internet. In: *First Monday* 17(7). http://dx.doi.org/10.5210/fm.v17i7.4050 (01/2017).

Stallman, Richard, 1983: *New UNIX Implementation*. http://bit.ly/1DSDoXW (5/2017).

Stallman, Richard, 2002: *Free Software, Free Society*. Boston: GNU Press.

Steinmacher, Igor / Conte, Tayana / Redmiles, David / Gerosa, Marco, 2015: Social Barriers Faced by Newcomers Placing their First Contribution in Open Source Software Projects. In: *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, 1379–1392.

Stiller, Ase, 2011: The Open Source Trials: Hanging in the Legal Balance of Copyright and Copyleft. In: *Vision Mobile Blog* 3/15/2011. http://www.visionmobile.com/blog/2011/03/the-open-source-trials-hanging-in-the-legal-balance-of-copyright-and-copyleft/ (5/2017).

Stokel-Walker, Chris, 2014: The Internet Is Being Protected By Two Guys Named Steve. In: *Buzzfeed* 4/24/2014. http://www.buzzfeed.com/chrisstokelwalker/the-internet-is-being-protected-by-two-guys- named-st#.ooP0j2q2z (5/2017).

Suddaby, Roy, 2013: Book Review: The Janus Face of Commercial Open Source Software Communities. In: *Organization Studies* 34(7), 1009–1011.

Sydow, Jörg / Schüssler, Elke / Müller-Seitz, Gordon, 2016: *Managing Inter-organizational Relations: Debates and Cases*. London: Palgrave Macmillan.

Tapscott, Don / Williams, Anthony D., 2006: *Wikinomics. How Mass Collaboration Changes Everything*. New York: Portfolio.

Torvalds, Linus, 1998: LINUX Manifesto. Interview. In: *Boot Magazine* 1998(7–8), 32–37.

Torvalds, Linus, 2002: *Re: [PATCH] Remove Bitkeeper Documentation from Linux Tree*. Linux Kernel Mailinglist 4/20/2002. http://lwn.net/2002/0425/a/ideology-sucks.php3 (5/2017).

UNCTAD – United Nations Conference of Trade and Development, 2012: *Information Economy Report 2012*. New York / Genf: United Nations.

Wasserman, Anthony, 2013: Community and Commercial Strategies in Open Source Software. In: *Information Technology* 55(5), 181–188.

Weber, Steven, 2000: *The Political Economy of Open Source*. BRIE Working Paper 140. Berkeley: University of California.

Weinberg, Bill, 2015: The Internet of Things and Open Source. In: Podnar Žarko, Ivana / Pripužić, Krešimir / Serrano, Martin (Eds.): *Interoperability and Open-Source Solutions for the Internet of Things*. Lecture Notes in Computer Science No. 9001. Cham: Springer, 1–5.

Weinberg, Bill (2015b): *Open Source and The Internet of Things: A Reality Check*. http://blog.blackducksoftware.com/open-source-and-the-internet-of-things-a-reality-check/ (5/2017).

West, Joel / Bogers, Marcel, 2014: Leveraging External Sources of Innovation. A Review of Research on Open Innovation. In: *Journal of Product Innovation Management* 31(4), 814–831.

West, Joel / Bogers, Marcel, 2017: Open Innovation: Current Status and Research Opportunities. In: *Innovation* 19(1), 43–50.

Further Publications

Research Contributions to Organizational Sociology and Innovation Studies

Dolata, Ulrich, 2017: *Social Movements and the Internet. The Sociotechnical Constitution of Collective Action*. SOI Discussion Paper 2017-02.

Dolata, Ulrich, 2017: *Apple, Amazon, Google, Facebook, Microsoft. Market Concentration – Competition – Innovation Strategies*. SOI Discussion Paper 2017-01.

Kungl, Gregor / Geels, Frank W., 2016: *The Destabilisation of the German Electricity Industry (1998-2015)*. SOI Discussion Paper 2016-02.

Schrape, Jan-Felix, 2016: *Social Media, Mass Media and the 'Public Sphere'. Differentiation, Complementarity and Co-existence*. SOI Discussion Paper 2016-01.

Schrape, Jan-Felix, 2015: *Open Source Softwareprojekte zwischen Passion und Kalkül*. SOI Discussion Paper 2015-02.

Radig, Ann-Kathrin, 2015: *Der Wandel des deutschen Videoverleihmarktes durch Digitalisierung und Internet*. SOI Discussion Paper 2015-01.

Dolata, Ulrich, 2014: *Märkte und Macht der Internetkonzerne. Konzentration – Konkurrenz – Innovationsstrategien*. SOI Discussion Paper 2014-04.

Kungl, Gregor, 2014: *The Incumbent German Power Companies in a Changing Environment*. SOI Discussion Paper 2014-03.

Dolata, Ulrich / Schrape, Jan-Felix, 2014: *Masses, Crowds, Communities, Movements. Collective Formations in the Digital Age*. SOI Discussion Paper 2014-02.

Neukirch, Mario, 2014: *Konflikte um den Ausbau der Stromnetze. Status und Entwicklung heterogener Protestkonstellationen*. SOI Discussion Paper 2014-01.

Dolata, Ulrich / Schrape, Jan-Felix, 2013: *Zwischen Individuum und Organisation. Neue kollektive Akteure und Handlungskonstellationen im Internet*. SOI Discussion Paper 2013-02.

Kosche, Robert, 2013: *Kollektive Identitäten in Industrial Cultural Districts*. SOI Discussion Paper 2013-01.

Fuchs, Gerhard / Hinderer, Nele / Kungl, Gregor / Neukirch, Mario, 2012: *Adaptive Capacities, Path Creation and Variants of Sectoral Change*. SOI Discussion Paper 2012-02.

Fuchs, Gerhard / Wassermann, Sandra, 2012: *Organising a Market. Photovoltaics in Germany*. SOI Discussion Paper 2012-01.

Werle, Raymund, 2011: *Institutional Analysis of Technical Innovation. A Review*. SOI Discussion Paper 2011-04.

Dolata, Ulrich, 2011: *Radical Change as Gradual Transformation. Characteristics and Variants of Socio-technical Transitions*. SOI Discussion Paper 2011-03.

Dolata, Ulrich, 2011: *The Music Industry and the Internet*. SOI Discussion Paper 2011-02.

Schrape, Jan-Felix, 2011: *Der Wandel des Buchhandels durch Digitalisierung und Internet*. SOI Discussion Paper 2011-01.

Books

Dolata, Ulrich, 2013: *The Transformative Capacity of New Technologies. A Theory of Sociotechnical Change*. London: Routledge.

Dolata, Ulrich, 2011: *Wandel durch Technik. Eine Theorie soziotechnischer Transformation*. Frankfurt / New York: Campus.

Dolata, Ulrich / Schrape, Jan-Felix (Hg.), 2013: *Internet, Mobile Devices und die Transformation der Medien. Radikaler Wandel als schrittweise Rekonfiguration*. Berlin: Edition Sigma.

Fuchs, Gerhard (Hg.), 2017: *Lokale Impulse für Energieinnovationen*. Heidelberg: Springer.

Fuchs, Gerhard / Shapira, Philip (Eds.), 2014: *Rethinking Regional Innovation. Path Dependency or Regional Breakthrough?* Springer: Shanghai (chinese edition).

Schrape, Jan-Felix, 2016: *Open-Source-Projekte als Utopie, Methode und Innovationsstrategie*. Glückstadt: Hülsbusch.

Schrape, Jan-Felix, 2015: *Kommunikation und Partizipation im Social Web. Eine Übersicht*. Studienbrief der FernUniversität in Hagen.

Schrape, Jan-Felix, 2012: *Wiederkehrende Erwartungen. Prognosen, Visionen und Mythen um neue Medien seit 1970*. Boizenburg: VWH.

Schrape, Jan-Felix, 2011: *Gutenberg-Galaxis Reloaded? Der Wandel des deutschen Buchhandels durch Internet, E-Books und Mobile Devices*. Boizenburg: VWH.

Schrape, Jan-Felix, 2010: *Neue Demokratie im Netz?* Bielefeld: Transcript.

Recent Journal Articles and Contributions in Edited Volumes

Dolata, Ulrich, 2016: Forschung und Entwicklung in der Wirtschaft. In: Simon, Dagmar et al. (Hg.): *Handbuch Wissenschaftspolitik*. Heidelberg: Springer, 609–626.

Dolata, Ulrich, 2016: Apple, Amazon, Google, Facebook. Konzentration, Konkurrenz und Macht im Internet. In: *Z – Zeitschrift Marxistische Erneuerung* 108, 55–68.

Dolata, Ulrich, 2016: Internetökonomie und Internetkonzerne. Märkte – Expansion – Macht. In: Schröder, Lothar / Urban, Hans-Jürgen (Hg.): *Gute Arbeit. Digitale Arbeitswelt. Ausgabe 2016*. Frankfurt (Main): Bund, 148–155.

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

Dolata, Ulrich / Schrape, Jan-Felix, 2016: Masses, Crowds, Communities, Movements: Collective Action in the Internet Age. In: *Social Movement Studies* 18(1), 1–18.

Dolata, Ulrich / Schrape, Jan-Felix, 2015: Zwischen Ermöglichung und Kontrolle. Kollektive Formationen im Web. In: *Forschungsjournal Soziale Bewegungen* 28(3), 17–25.

Dolata, Ulrich / Schrape, Jan-Felix, 2014: Kollektives Handeln im Internet. Eine akteurtheoretische Fundierung. In: *Berliner Journal für Soziologie* 24(1), 5–30.

Fuchs, Gerhard, 2015: Building the Agenda for Carbon Dioxide Capture and Storage: Limits of EU-Activism. In: Tosun, Jale et al. (Eds.): *Energy Policy Making in the EU*. London: Springer, 205–223.

Fuchs, Gerhard, 2014: The Governance of Innovations in the Energy Sector: Between Adaptation and Exploration. In: *Science & Technology Studies* 27(1), 34–53.

Fuchs, Gerhard / Fettke, Ulrike, 2017: Incumbent-Challenger-Interaktionen und die Veränderungen im Markt für Stromerzeugung in Deutschland. In: Giacovelli, Sebastian (Hg.): *Die Energiewende aus wirtschaftssoziologischer Sicht*. Wiesbaden: Springer VS, 15–43.

Fuchs, Gerhard / Hinderer, Nele, 2016: One or many Transitions: Local Electricity Experiments in Germany. In: *Innovation: European Journal of Social Science Research* 29(3), 320–336.

Fuchs, Gerhard / Hinderer, Nele, 2014: Situative Governance and Energy Transitions in a Spatial Context: Case Studies from Germany. In: *Energy, Sustainability and Society* 4:16.

Geels, Frank W. / Kern, Florian / Fuchs, Gerhard / Hinderer, Nele / Kungl, Gregor / Mylan, Josephine / Neukirch, Mario / Wassermann, Sandra, 2016: The Enactment of Socio-technical Transition Pathways. In: *Research Policy* 45(4), 896–913.

Kungl, Gregor, 2015: Stewards or Sticklers for Change? Incumbent Energy Providers and the Politics of the German Energy Transition. In: *Energy Research & Social Science* 8, 13–23.

Neukirch, Mario, 2016: Protests against German electricity grid extension as a new social movement? A journey into the areas of conflict. In: *Energy, Sustainability and Society* 6:4.

Neukirch, Mario, 2015: Mehr Netz mit weniger Kohle? Zwei ökologische Perspektiven auf ‚Korridor D'. In: *politische ökologie* 141, 132–135.

Schrape, Jan-Felix, 2017: Reciprocal Irritations: Social Media, Mass Media and the Public Sphere. In: Paul, Regine / Mölders, Marc / Bora, Alfons / Huber, Michael / Münte, Peter (Hg.): *Society, Regulation and Governance*. Cheltenham: Edward Elgar Publishing, 138–149.

Schrape, Jan-Felix, 2016: Big Data: Informatisierung der Gesellschaft 4.0. In: *Berliner Debatte Initial* 27(4), 12–21.

Schrape, Jan-Felix, 2016: Open-Source-Projekte: Vom Nischenphänomen zum integralen Bestandteil der Softwareindustrie. In: *WSI Mitteilungen* 8/2016, 603–612.

Schrape, Jan-Felix, 2016: Soziologie als ‚Marke'. In: *Soziologie* 45(3), 279–293.

Schrape, Jan-Felix, 2016: ‚Big Data' als Erwartungsraum. In: *Soziopolis* vom 20.10.2016.

Schrape, Jan-Felix, 2015: Social Media, Massenmedien und Öffentlichkeit. In: Imhof, Kurt et al. (Hg.): *Demokratisierung durch Social Media?* Wiesbaden: Springer VS, 199–212.

Schrape, Jan-Felix / Dickel, Sascha, 2017: The Logic of Digital Utopianism. In: *Nano Ethics*, DOI: 10.1007/S11569-017-0285-6.

Schrape, Jan-Felix / Dickel, Sascha, 2015: Dezentralisierung, Demokratisierung, Emanzipation. Zur Architektur des digitalen Technikutopismus. In: *Leviathan* 43(3), 442–463.

http://www.uni-stuttgart.de/soz/oi/publikationen/