Article

CONVERGENCE

Convergence: The International Journal of Research into New Media Technologies
1–19

DOI: 10.1177/1354856517735795
journals.sagepub.com/home/con
SAGE

Open-source projects as incubators of innovation: From niche phenomenon to integral part of the industry

Jan-Felix Schrape
Universität Stuttgart, Germany

Abstract
Over the last 20 years, open-source development has become an integral part of the software industry. Against this backdrop, this article 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 information technology companies. This is followed by the identification of four ideal-type 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 lost their subversive connotations while, in contrast to former cases of collective invention, remaining viable beyond the initial phase of innovation.

Keywords
Change, collective invention, decentralization, innovation, media induced, open-source, peer production, professionalization, sociotechnical software industry, transitions

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 (see, critically, Pomerantz and Peek, 2016). Today, projects of all kinds flaunt the attribute of openness and its associated promise of more decentralized and democratic coordination structures. More specifically, the promise entails that technology could break with the traditional distribution of social roles and empower once-passive citizens, users and consumers (e.g. Ritzer et al., 2012; West et al., 2014; see, for a critical overview, Dickel and Schrape, 2017; Fuchs, 2015).

Corresponding author:
Jan-Felix Schrape, Abteilung für Organisations und Innovationssoziologie (SOWI VI), Institut für Sozialwissenschaften, Universität Stuttgart, Seidenstr 36, 70174 Stuttgart, Germany.
Email: felix.schrape@sowi.uni-stuttgart.de

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 social sciences, accustomed to intellectual property rights as drivers of innovation, this increase was initially received with surprise (Lessig, 1999: 1411). Not long thereafter, however, open-source became acknowledged as an emerging production model that is based on voluntary and self-directed collaboration among equals and that could break with classical forms of socio-economic coordination (Lakhani and Hippel, 2003). In that context, the concept of 'commons-based peer production', introduced by Yochai Benkler in 2002, gained traction. Hailed as a technically effective 'collaboration among large groups of individuals [. . .] without relying on either market pricing or managerial hierarchies' (Benkler and Nissenbaum, 2006: 381), commons-based peer production was to be accompanied with 'systematic advantages [. . .] in identifying and allocating human capital/creativity' (Benkler, 2002: 381; see, critically Shaw and Hill, 2014). More recently, the concept has been applied in adjacent fields such as the service sector or the production of material goods (e.g. Kostakis et al., 2016; Sundararajan, 2016). In this very context, Jeremy Rifkin (2014: 1) even stated that 'the capitalist era is passing' and that 'a new economic paradigm – the Collaborative Commons – is rising [. . .]'.

However, studies of recent open-source software (OSS) projects have shown that the growth of the communities goes hand in hand with the formation of hierarchical decision-making routines, that leading information technology (IT) companies are increasingly gaining influence over important projects 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 based on the contributions of employed developers. For example, in the Linux kernel project, often referred to as a typical OSS project, 85% of the changes were made by programmers who 'are being paid for their work' (Corbet and Kroah-Hartman, 2016: 12).

Against this backdrop, this article seeks to develop a systematic overview of open-source communities and their socio-economic contexts. I begin with a causal reconstruction (Héritier, 2008; Mayntz, 2004)[1] of the genesis of OSS projects and their changing relationships to established IT companies on the basis of available literature, market statistics, documents and informal background talks with eight software engineers from Germany, Switzerland and the United States. Based on aggregated empirical data, this is followed by the identification of four ideal-type variants of current open-source projects that differ from each other in their modes of coordination and the degree of corporate involvement. I then examine from an organizational–sociological point of view why OSS projects have largely lost their subversive connotations while nevertheless, in contrast to previous types of collective invention, remaining viable beyond the emergence of predominant solutions and their commercial exploitation: In a software industry that is characterized by very short innovation cycles, OSS projects have proven to be important incubators for branch-defining standards and infrastructures.

The genesis and institutionalization of open-source projects

Soon after OSS projects became widely known, a number of articles were published that, offering initial explanations of their success and underlining their subversive character, essentially form the basis of the social sciences view of open-source to this day (e.g. Moody, 2002; Weber, 2000). These texts were primarily oriented towards narratives coming from the scene itself and, with few exceptions (e.g. Lerner and Tirole, 2002), dispensed with any socio-economic contextualization. As the following three-step reconstruction shows, however, the dividing line between free and

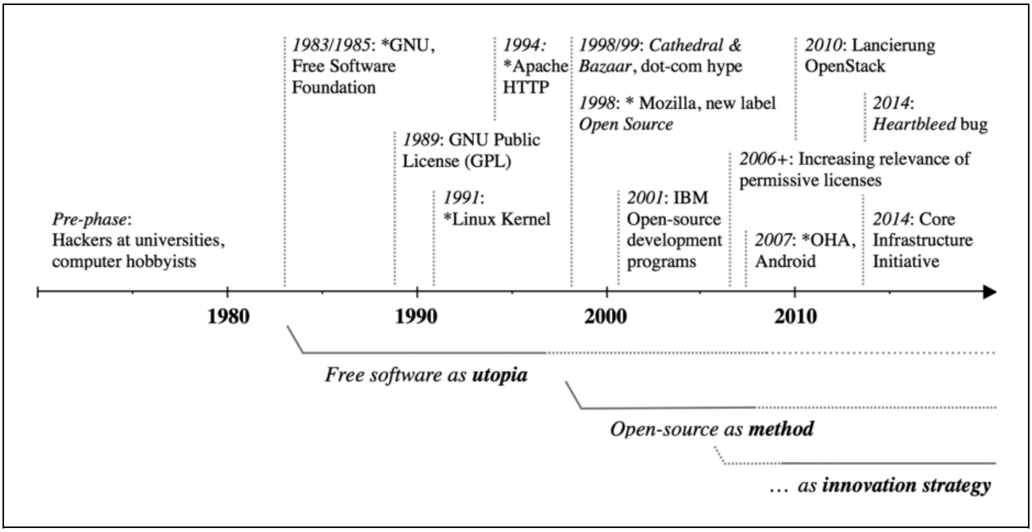

Figure 1. OSS as utopia, method and innovation strategy. OSS: Open-source software.

proprietary software development has never been clear-cut, and the involvement in open-source projects, ongoing for the last 20 years, has become an intentional, studied component of the innovation strategies of leading IT providers (Figure 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 had been developed in close cooperation between manufacturers and users, with computer programmes not yet perceived as a product that is independent of hardware but rather 'as a research tool to be developed and improved by all users' (Gulley and Lakhani, 2010: 6). Starting at the end of the 1960s, however, software began to be acknowledged as a separate product, prompted 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 – and the founding of the first specialized software companies (Fisher et al., 1983).

In addition, the spread of minicomputers in the 1960s played an essential role in the development of a stand-alone software sector. These types of computers differed from 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. At US universities, minicomputers, often donated by their manufacturers, offered a hotbed for informal project groups that sought to overcome the limitations of existing IT systems and paved the way for the amateur computing scene of the late 1970s (Levy, 1984). 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 onwards – as soon as permitted under antitrust law (Holtgrewe and Werle, 2001). Or, the computer game *Spacewar*!, programmed by students from the Massachusetts Institute of Technology (MIT) in 1962, was utilized as the basis for numerous arcade machines.

What commercial IT providers liked less about the computer hobbyist scene was its predilection to share and circulate programmes without paying for them. In an open letter, the software entrepreneur Bill Gates (1976) complained about this circumstance as follows: 'Hardware must be paid for, but software is something to share. [. . .] Who can afford to do professional work for nothing?' As a result, by the 1980s, most software products were sold only as binary files that had no accessible source code. At the same time, amendments to copyright law increased the protection and exclusivity of software products (Menell, 2002). As an ethical statement about this turn of events, the MIT employee Richard Stallman (1983) announced his plan to develop an 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 stand-alone 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 swiftly enlisted large-scale industrial sponsors such as the hardware manufacturers Sony and Hewlett-Packard (HP) who had an interest in inexpensively licensable software. The most relevant innovation, however, was the *introduction of licensing models*, like the General Public License (GPL) published in 1989, which ensure that any forks of free software remain free: '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). From 2001 onwards, violations of the GPL were the object of numerous court proceedings against companies such as Skype, Cisco and D-Link (Jaeger, 2010; Stiller, 2011). It should be noted, however, that the 'court of public opinion' played an equally important role in Usenet, and later on the 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 focusing on expensive workstations and its ideological connotations – two problems to which the Linux kernel project offered a solution. Linux was introduced in 1991 by then student Linus Torvalds as a free operating system kernel for the more affordable microcomputers 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 FSF: '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" [. . .]' (Torvalds, 2002). Another reason for the success of Linux was the spread of the World Wide Web from 1993 onwards, as it facilitated both access to and participation in the project and its coordination.

Nonetheless, the Linux project, too, was initially known only within expert circles and it was not until the publication of the widely read book *The Cathedral and the Bazaar* (Raymond, 1997, 1999) that the Linux kernel became publicly known. The main thesis of the book was: Whereas in traditional production models, a programme'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 groups are maintained by self-organization without central management – corresponding to a *bazaar*. Nonetheless, critics observed early on that while in both cases many suggestions came from the community, final changes were released by only one person (Bezroukov, 1999a; Connell, 2000). 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' (Torvalds, 1998: 36).

Overall, GNU and Linux stand as two main flagship projects for free software development of the 1980s and 1990s whose success was facilitated by the increased efficiency of communication on the Internet. This environment spurred the emergence of informal conventions as well as licensing models, which protect collective work results from being claimed by any one individual or entity. It was in this context that the first narratives circulated that hailed free software development as a radical new way to produce software without power asymmetries. These narratives gained, at least for some time, currency among social scientists (e.g. Benkler, 2002; Tapscott and Williams, 2006) – although informed observers marked them early as rather biased descriptions of the 'hacker culture' (e.g. Bezroukov, 1999b; Raymond, 1998a).

Open-source as a method

In the decade of the 2000s, then, OSS development became an increasingly recognized method within the industry. This may be attributed to the following dynamics.

First, a growing number of IT companies began outsourcing the development of software to the open-source field. Of those, Netscape Communications was a 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 *Mozilla community.* This open-source project, which engendered the browser Firefox in 2004, received financial and human resources support from AOL/Netscape until the founding of the Mozilla Foundation in 2003. With its announcement, Netscape Communications Inc (1998) aimed primarily to 'expand its client software leadership by [. . .] building a community that addresses markets and needs we can't address on our own [. . .]'.

Second, at the beginning of 1998, a group that had formed around Eric S. Raymond concluded that the term 'free software' could impede the spread of OSS in commercial contexts, given its possible political connotations. Therefore, they founded the open source initiative and introduced the new label 'open source', which they considered to emphasize the superiority of this software development model while deflecting from any ethical or sociopolitical aspects (Raymond, 1998b). However, to this day, this change in course has not been endorsed by the FSF: '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: 57). Ongoing to this day, some authors try to evade this disagreement by means of hybrid acronyms such as FLOSS (Free/Libre Open Source Software).

The third main factor that contributed to the recognition of open-source was the *stock market success* of some open-source companies in 1999 as a result of the dot-com boom. 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 architectures for enterprises. The initial public offerings of these three companies were, in fact, among the most spectacular of all time, resulting in public attention for the open-source scene as a whole (e.g. Gelsi, 1999).

Table 1. The most used open-source software licenses worldwide.

	E.g. used by	May 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
ISC License	OpenBSD, nhttpd	5	0	Permissive	2003
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 (May 2017).

These interrelated trends, combined with the continued expansion of the IT market, led to the rapid proliferation of OSS projects. Indeed, their number grew from several hundred in 1999 to the several million projects which can today be found on platforms such as GitHub and SourceForge. Given this increase, accompanied by the introduction of novel licensing models by companies and foundations, open-source licensing has been subject to strong diversification (Table 1). Alongside original ‘copyleft’ licences such as the GPL, which guarantee that free software must be forked under the same conditions (*strongly protective*), additional licences have been issued that permit the inclusion of free software in proprietary products as long as these elements remain open-source (*weakly protective*) or permit the publication of derivations under downright restrictive conditions (*permissive*). This diversity expands the strategic options, especially for commercial stakeholders (Lerner and Schankerman, 2010): After the GPL 3.0 was published, closing previous gaps, Apple replaced the GNU compiler collection (GCC) in its development environment Xcode with a solution with a permissive licence; Google decided from the outset to put project’s own code of Android under the Apache 2.0 License.

Concurrently, we can observe a corporatization of open-source projects in two ways. On the one hand, major projects such as the Linux kernel, the Apache HTTP Server and the cloud platform OpenStack are today funded primarily by donations from companies or operate like the browser engine WebKit (Apple) and the operating system Android (Google) under the aegis of commercial providers. 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% of all contributions between 2000 and 2011 were made during standard 9–5 working hours. The Linux Foundation (Corbet and Kroah-Hartman, 2016; Corbet et al., 2009–2015), for its part, observed that the portion of independent programmers in kernel development (2009: 18%, 2016: 8%) is steadily declining compared to that of company-associated contributors (e.g. from IBM, Samsung, Intel).

Table 2. Estimated global market share of open-source software (in %, installed base).

	Open-source	2010	2016	Competitors	2010	2016
Operating system personal computer (a)	GNU/Linux	1	2	MS Windows	94	89
				Apple Mac OS X	5	11
Operating system mobile devices (b)	Android	11	72	Apple iOS	30	20
				Symbian/Nokia OS	33	1
				Windows Phone	—	1
				Blackberry	14	> 1
Web browser desktop (c)	Mozilla Firefox	31	15	MS IE, MS Edge	47	24
	Google Chrome*	14	59	Apple Safari	5	4
OS public servers (d)	Linux**	69	67	MS Windows	31	33
Web server [active sites] (e)	Apache	72	51	Microsoft IIS	21	12
	Nginx	4	32	Google Servers	1	1
Web content management system (g)	WordPress	51	59	Blogger (Google)	2	2
	Joomla	12	7	Bitrix	—	1
	Drupal	7	5	vBulletin	8	> 1

Source: (a, b) NetApplications; (c) StatCounter; (d, e, f) W3techs (Status: May 2017).
*Mainly based on the Chromium OSS project.
**includes Unix-like.

Open-source as innovation strategy

It is in this way that open-source development increasingly became enmeshed with the software industry. Today, OSS architectures are a constituent part of many operating systems and cloud services as well as they are predominating in the area of basic IT infrastructures (Table 2). In particular, in the enterprise software markets, which account for more than 80% of global software sales, 'a widespread use of open-source technology' can be observed (Driver, 2014; Miller and Nelson, 2016). Market researchers attribute this not only to the cost advantages but also to the 'inherent trialability' of OSS solutions (Spinellis and Giannikas, 2012: 667). Thus, it is not surprising that by now all the key IT companies are involved in relevant open-source projects: They use these working environments as a means to protect standards that are favourable to them and to expand their proprietary research and development (R & D) through 'controlled openings at the edges' (Dolata, 2017: 20).

Microsoft – the company which has once termed open-source 'an intellectual property destroyer' (Hayes, 2001: 78) – launched its subsidiary MS Open Technologies in 2012. Since then, it has put the .NET Framework, software development kits for its cloud computing service Azure as well as many other components under an OSS licence, namely, in order 'to achieve a strategic objective, such as promoting industry standards, advancing interoperability, or attracting and enabling our external development community' (Microsoft Inc, 2017: 20). 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 a standard practice in numerous manufacturer-specific architectures. *Apple*'s operating system packages macOS, iOS, tvOS and watchOS, for example, are at its core based on the Unix-like operating system Darwin and contain hundreds of other OSS components (e.g. WebKit, XQuartz).

IBM had already invested several hundred million US dollars in the development of Linux at the turn of the millennium, namely, as a means to counteract Microsoft's dominance in the enterprise sector and to set-up a service business around OSS. Today, IBM is involved in hundreds of OSS projects, among them is the cloud platform OpenStack, in which *Intel* and *HP* 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 [. . .])' (Lerner, 2012: 43). It is for similar reasons that *SAP*, *Oracle* and Adobe are participating in OSS projects. In addition, many consumer electronics products from Samsung and other leading companies are enabled with OSS. For smaller providers, in particular, their involvement in OSS projects also serves as a 'marketing tool to increase brand recognition' (Dahlander and Magnusson, 2008: 638).

Besides 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 endeavouring to build a business through support services. However, with the exception of the Linux distributor Red Hat, which had been cooperating early on with hardware vendors and which today is the market leader in commercial Linux systems, most of the companies that were launched during the dot-com boom quickly folded (Levine, 2014). Although the OSS environment has recently given rise to new start-ups (e.g. Hortonworks), most of these firms are characterized by a low level of identification with Stallman's ideals. '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' (OSS provider, in Bergquist et al., 2012: 8).

A special variant of corporate open-source exposures is the development of the Android mobile operating system by the Open Handset Alliance initiated and led by *Google*. Advertised as a pure open-source project 'to make sure there was no central point of failure, where one industry player could restrict or control the innovations of any other' (http://source.android.com), the development of the operating system is *de facto* controlled by Google alone: 'Google largely follows the "cathedral" model [. . .]. Because it fully controls the development of the OS, Google can determine the technological specifications to which Android partners must abide' (Spreeuwenberg and Poell, 2012). With the launch of Android, Google apparently succeeded above all in facilitating the seamless access to its services for as many devices as possible. Whereas Google generated 99% of its revenue from advertising in 2007, the sale of its digital content and services accounted for 11% of sales in 2016 (Alphabet Inc, 2017).

In that sense, many popular open-source communities by now have close financial ties with leading IT companies, which are investing in open-source projects as part of their overarching innovation strategies (Table 3).[2] In the case of corporate-initiated projects (e.g. Android), this entanglement is obvious. However, foundation-supported communities (e.g. GNU) grant their donors seats on the boards of their umbrella organizations. The latter, while not directly in control of the development activities, provides the technical infrastructures, distribute financial resources and define the orientation of the project. Together with their involvement in the code development as such, these leading IT companies are thereby securing a considerable influence on relevant communities while at the same time allowing for greater predictability in planning for these projects as regards both their human and financial resources.

Varieties of open-source projects

Over the last two decades, OSS development thus has seen an increasing corporate embracement. As a consequence, the array of open-source projects has become larger and broader: At one end of the spectrum, some communities are still committed to Richard Stallman's socio-ethical ideals,

Table 3. Popular projects on Open Hub (Web catalogue for open-source projects).

Project	Commits (last year)*	Umbrella organization	Primary funding source
Android	104,151	Google Inc., Open Handset Alliance (84+ companies)	
KDE	87,466	KDE e.V.	Patronages (includes Google, SUSE, Qt)
Chromium	77,562	Google Inc.	
OpenStack	76,130	OpenStack Foundation	Members (includes HP, IBM, Red Hat)
Linux Kernel	73,254	Linux Foundation	Members (includes HP, Intel, IBM, Red Hat)
Mozilla Firefox	53,255	Mozilla Foundation	Donations, royalties (until 2014: 90% Google)
Ubuntu (Touch)	52,128**	Canonical Ltd.	Canonical, partners (includes Intel, Cisco, HP)
Fedora	34,222	Fedora Project (Red Hat)	Red Hat Inc.
Debian Linux	26,782	Debian Project	Donations, partners (includes HP, 1&1)
LibreOffice	15,733	Document Foundation	Donations (includes Google, Red Hat, Intel)
WebKit	13,059	Apple Inc.	
Eclipse IDE	7715	Eclipse Foundation	Members (includes IBM, SAP, Oracle, Bosch)
GNU CC	7602	Free Software Foundation	Members, patronages (includes Google, IBM)
OpenSSL	3225	— (OpenSSL Foundation)	Since 2014: Core Infrastructure Initiative
Joomla!	2884	Open Source Matters NPO	Sponsors, advertising, affiliates
WordPress	2348	WordPress Foundation	Automattic Inc., donations, events
Apache HTTP	2103	Apache Foundation	Donations (includes Google, Microsoft)
Arch Linux	252	—	Smaller private donations
jEdit	178	—	Smaller private donations

Source: Open Hub (May 2017), Annual Reports.
*Updates January 2016–January 2017.
**2015–2016.

operate independently of corporate interests and are aligned with egalitarian organizational principles. At the other end, we find a large number of projects that follow hierarchical development models and that are under the direct control of leading technology corporations. From an organizational–sociological point of view (i.e. Ahrne et al., 2016; Scott, 2004; Van de Ven et al., 1976) and based on available empirical data (i.e. licensing documents, certificates, technical specifications, membership listings, mailing lists, wikis; see, for details, Schrape, 2016), 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 (Table 4).

Corporate-led collaboration projects

Corporate-led and -initiated 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 as well as Chromium (Web browser), WebKit (HTML rendering engine) and Fedora (Linux distribution), the strategic control clearly lies with Google, Apple and Red Hat, respectively. Android's own code, for instance, is run under permissive licences, which, in combination with further legal frameworks such as the 'contributor agreement', give Google comprehensive steering control. Most notably, the 'compatibility definition document' (source.android.com/compatibility) tightly defines the

Table 4. Ideal-type manifestations of open-source projects.

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

requirements of hardware devices running Android and how the application programming interfaces for third-party providers are developed and managed.

In the cloud computing project OpenStack, big sponsors likewise have considerable influence: As appointed or elected members of the technical committee and the board of directors, companies such as Rackspace, Intel, HP, IBM, Red Hat, AT&T and Cisco are able to steward the technical direction of the OpenStack project and define its overarching strategic orientation. Although such a community of companies 'admits individual contributions, it clearly prioritizes corporate interests, and participating companies, which can be commercial competitors, employ most of the developers' (Gonzalez-Barahona et al., 2013: 39; Teixeira et al., 2015).

This type of corporate collaboration under the terms of open-source licences allows to overcome two knowledge-sharing dilemmas (Larsson et al., 1998): Firstly, OSS licences prevent the direct proprietarization of the collectively developed code by any individual entity. Secondly, these same licences 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). In addition, in this day and age, it is often more feasible to create new software products by building on already existing open-source architectures than by developing a software from scratch.

Heterarchical infrastructure projects

Infrastructure projects, whose products are ever-present beneath the visible surface of IT systems, are also closely intertwined with corporate contexts. Some were initially based on architectures that were formerly proprietary (e.g. the IDE Eclipse, the content management system Joomla). Others (e.g. the Apache HTTP Server) were characterized by rapid organic growth, since they offered solutions to previously unaddressed problems, making them interesting to companies early on, particularly since OSS infrastructures do not carry any impetus for application code or hardware to be open itself (Greenstein and Nagle, 2014; Weinberg, 2015).

Today, infrastructure projects are primarily supported by medium and large IT companies that seek to adjust the code to their business needs. However, these communities are, although some are accepting targeted donations (e.g. Eclipse Foundation, 2016), not guided by corporate core circles, but they operate under the umbrella of non-profit organizations 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 freed from other tasks by their companies to work in the community, are more likely than lay programmers to advance to decision-making positions (Westenholz, 2012).

An infrastructure project that points to potential risks in the OSS model is the encryption software OpenSSL, which is used in many IT systems and Web platforms since the 1990s: Until 2014, OpenSSL was developed by one full-time programmer assisted by a small, voluntary team and received little financial support from the industry. In that context, ever new features were integrated into OpenSSL – yet without the according level of maintenance. In 2012, this culminated in an oversight that led to the major 'Heartbleed' bug, which was not discovered until 2014. In light of this 'worst vulnerability found [. . .] since commercial traffic began to flow on the Internet' (Steinberg, 2014), the Linux Foundation and leading companies formed the Core Infrastructure Initiative to fund projects that are critical to the functioning of the Web.

Elite-centric project communities

Elite-centric communities are likewise based to a large extent on the contributions of corporate developers, but their coordination takes place along a 'lieutenant system built around a chain of trust' (Kernel.Org, 2016) that is headed by an elected project manager (e.g. Debian Linux), a management team (e.g. Mozilla) or its founder as a 'benevolent dictator'. Linus Torvalds, for instance, 'is the final arbiter of all changes accepted into the Linux kernel' (Kernel.Org, 2016); Mark Shuttleworth as 'self-appointed benevolent dictator for life [. . .] plays a happily undemocratic role' as the sponsor of the Ubuntu project (Ubuntu Project, 2017). That said, neither Torvalds nor Shuttleworth would be well advised to regularly override the decisions of the technical boards, provided that they are interested in strong community involvement on the long run.

With the launch of Firefox, Mozilla also installed a 'rather rigorously controlled model' (Stamelos, 2014: 328) – from 'super-reviewers' and 'stewards' to two 'ultimate decision makers'. Since 1998, former Netscape manager Mitchell Baker holds one of these positions and is an executive chairwoman of the Mozilla Cooperation and the Foundation, which has over 1000 employees. Although voluntary participants are welcomed, between September 2015 and September 2016 only 17 volunteered but 228 new hires were introduced in the project's weekly updates (Mozilla Foundation, 2016). The Ubuntu project, too, relies on the work of the employees of Shuttleworth's for-profit company Canonical. Debian, in turn, holds 1000 voluntary developers that collaborate under the terms of the Debian Constitution and elect the project leader who (quite similar to a CEO) is the public face of Debian, distributes resources and defines its direction (Krafft, 2010).

While such a top-down management curtails the scope of the participants in these projects, it also counteracts fragmentation (Coleman, 2013; O'Mahony and 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 [. . .], [because] the decisions we take remain consistent and are coherent with our overall vision'. However, while in Debian or Mozilla, the project guidelines are formally fixed; in the Linux kernel, 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' (Kreiss et al., 2011: 252).

Egalitarian-oriented peer production communities

In contrast, 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 and Williams, 2006: 70). However, as is apparent from KDE (Linux/Unix user interface), GNU or LibreOffice, when these communities grow, they usually feature distinct leadership structures as well.

The community of LibreOffice, a fork of the discontinued OpenOffice, is maintained by self-organization alongside working groups but operate under the strategical guidance of the Document Foundation, including a board of directors and an advisory board, Red Hat being a member in both entities and the most active contributor in development (Corbet, 2015). 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 KDE Project (2017). The GCC 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).

Intrinsically motivated communities such as Arch Linux or jEdit (text editor), in turn, target their products to specific user groups, are rather irrelevant to the general market and are run by small teams. For this reason, 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 those 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). Moreover, once such communities reach a certain size and their interactions with external market actors intensify, they too tend to adopt a stable roster of corporate stakeholders as well as 'cathedral-like' organizational modes.

Open-source projects as incubators of innovation

The preceding chapters debunk two assumptions: One, that the technical infrastructures of the Internet can, on their own, resist an 'ossification of power' (Benkler, 2013: 225) in OSS communities. And two, that there is a 'networked information economy' (Benkler, 2006: 3) in which corporate actors (companies, non-governmental organizations, research institutes) are losing ground in the face of 'non-proprietary, voluntaristic, self-assisted practices' (Benkler, 2013: 213; cf. Suddaby, 2013). These assumptions do not hold for two main reasons.

Firstly, although the technical infrastructures used in OSS projects lay the foundation for their coordination processes, they in no way lead to a disintermediation or loss of relevance of social structuring patterns. In open-source communities and other Web-based projects (such as Wikipedia), too, collectively accepted rules, guidelines and hierarchical decision-making structures emerge that are characterized by asymmetrical power distributions. Indeed, such social institutionalization dynamics are a fundamental requirement for an OSS 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 and Schrape, 2016).

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' individual preferences (Perrow, 1991). As a result, companies and other organizations are able to bring in their resources more continuously and consistently than individual contributors. They therefore significantly contribute to creating a reliable planning environment for open-source projects, in turn garnering them considerable clout and influence over the community.

In that context, open-source projects could be seen to be subject not only to corporatization but also to a steadily intensifying embracement by established market actors. Indeed, the reconstruction of the institutionalization of OSS development presented above shows that the ideal image of an independent commons-based peer production, in reality, 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 free software components, followed by the increasing involvement of larger IT companies in OSS projects.

From the point of view of innovation research, such a development does not seem extraordinary or unusual: Like other niche innovations, free software projects were initially 'carried and developed by small networks of dedicated actors', often fringe actors, outsiders or enthusiasts yet became subject to professionalization and appropriation on the part of established actors as soon as they caught the attention of the mainstream markets (Geels and Schot, 2007: 400). In fact, history has seen many episodes of collective invention (Table 5) during which organizations or individual actors shared their knowledge in niches that were decoupled from the general market, thereby benefiting from 'cumulative advance' (Allen, 1983: 23). However, in contrast to former cases of collective invention, OSS projects remain viable beyond the initial stages of innovation, that is, beyond the emergence of predominant solutions and their commercial exploitation (Osterloh and 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 novel *licensing models designed to prevent the proprietarization* of collective work results. Today, these models comprise the core framework of OSS projects, allowing for a reliable project-specific collaboration and exchange of knowledge between individual developers as well as companies that may be direct competitors otherwise.
- At the same time, the rapid *advance of online technologies* has allowed for much greater efficiency in the verification of compliance with these conditions and has facilitated the access to projects as well as the spread of their products. In addition, they have contributed to solving a long-term problem faced by the sector, namely, that of coordinating large projects with developers from different contexts and locations (Brooks, 1975).
- Finally, in an industry that has been expanding for decades and that is characterized by very short innovation cycles, OSS projects have proven to be pivotal *incubators for branch-defining infrastructures*, standards and platforms (e.g. the Apache HTTP Server, the Linux kernel, OpenStack). This applies all the more since OSS software can be tested by the developers themselves and adapted to their requirements with little administrative effort.

Thus, at the turn of the millennium, a novel form of collaboration and collective development that initially took place in niches was adapted by the industry and is today a key element of the

Table 5. Some historical episodes of collective invention.

Episode	Knowledge exchange	Outcome
The Cornish Pumping Engine *ca. 1810–1850, Cornwall, England*	Exchange of technical know-how; comparison of progress via journals	Development of an efficient steaming engine for the mining industry
Furnace technologies *ca. 1850–1880, Cleveland Dis., England*	Exchange of knowledge via journals; collective trial-and-error process	Reduction of energy demand by height and temperature adjustments
Flat-panel displays *ca. 1969–1989, Japan/ Europe/USA*	Publication of proprietary research results in technical journals	Incremental development in the pre-commercial phase
Homebrew Computer Club *ca. 1975–1978 [1986], SF Bay area, USA*	Free exchange until the success of participating firms (e.g. Apple Inc.)	Development of the first personal computers for the mass market

Source: Allen, 1983; Spencer, 2003; Nuvolari, 2004; Powell and Giannella, 2010.

innovation strategies of leading IT providers. OSS licences, aided by the Internet, have contributed to ensuring the longevity of collective invention through sociotechnical means. Today, they comprise the legal and structural bases of collaboration projects that do not compete against classical forms of software production but instead complement and expand these.

Concluding remarks

Overall, the relationship between open-source projects and the well-established forms of socio-economic coordination (e.g. market, hierarchy) is not characterized by competition but by complementariness. As discussed in the previous sections, OSS projects have contributed to more flexibility in the collaboration between developers from divergent contexts, the task-specific cooperation between market actors as well as the modes of organization in software development at large – through which they evolved into industry-fundamental incubators of innovation. At the same time, however, freely available source code alone does not result in more transparent coordination patterns than elsewhere, in a disintermediation in the established societal resource and power distribution or a general democratization of innovation processes.

Therefore, the prospect that the original concept of commons-based peer production, which was rarely applied as such even in early OSS communities, could readily be adapted to neighbouring socio-economic fields such as three-dimensional printing (e.g. Rifkin, 2014) or sociopolitical 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 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) or the global hegemony of a few companies over the key infrastructures of communication and information retrieval to a degree unprecedented in media history.

Against this background, social scientists would do well to scrutinize popular catchwords such as Open Innovation, Web 2.0 or Open-source, often deliberately coined by dedicated 'visioneers' in the San Francisco Bay area or other high-tech hubs (McCray, 2013),[3] before adopting them as quasi-sociological terms. Instead, efforts should be made to examine to what degree the associated expectations might point to reoccurring semantic patterns and to assess their socio-cultural impacts. For example, even though the visions of open collaboration, participation and empowerment, which are associated with OSS projects and more recent phenomena such as the 'maker culture', have not been brought to fruition as intended, they nevertheless draw attention to new development paths, contribute to the creation of innovation niches, serve as a legitimizing basis in economic or political decision-making processes, and enhance the cohesiveness of the respective communities. In this spirit, the themed openness narratives can indeed be regarded as 'productive types of communication' (Dickel and Schrape 2017: 54) – provided they are not misinterpreted as objective descriptions of empirical facts and dynamics.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

Notes

1. 'Causal reconstruction does not look for statistical relationships among variables but seeks to *explain* a given social phenomenon [. . .] by identifying the processes through which it is generated' (Mayntz, 2004: 238). As a socio-scientific observation concept, it focuses on contextualization, 'organizes (scientific and prescientific) knowledge' and 'emphasizes the questions that are worthwhile asking' (Scharpf, 1997: 29).
2. For instance, a significant part of Mozilla's income arrives in the form of royalties from the Firefox search box, that is, contracts with major search engine providers. The main sponsors of the Apache Software Foundation include Google, Microsoft and Facebook as platinum members with donations of $100,000+ per year.
3. 'Visioneering means developing a broad and comprehensive vision for how the future might be radically changed by technology, doing research and engineering to advance this vision, and promoting one's ideas to the public and policymakers in the hopes of generating attention and perhaps even realization' (McCray, 2013: 13).

References

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

Allen R (1983) Collective Invention. *Journal of Economic Behavior and Organization* 4(1): 1–24.

Alphabet Inc (2017) Form 10-K 2016. Available at: https://abc.xyz/investor/ (accessed 30 May 2017).

Benkler Y (2002) Coase's penguin, or, Linux and the nature of the firm. *Yale Law Journal* 112: 369–446.

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

Benkler Y (2006) *The Wealth of Networks*. New Haven: Yale University Press.

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

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

Bennett WL, Segerberg A and Walker S (2014) Organization in the crowd: Peer production in large-scale networked protests. *Information, Communication and Society* 17(2): 232–260.

Bergquist M, Ljungberg J and Rolandsson B (2012) Justifying the value of open source. *ECIS 2012 Proceedings*. Avaialble at: http://aisel.aisnet.org/ecis2012/122/ (accessed 30 May 2017).

Bezroukov N (1999a) A second look at the cathedraland the bazaar. *First Monday* 4(12). Available at: http://firstmonday.org/article/view/708/618 (accessed 30 May 2017).

Bezroukov N (1999b) Open source software development as a special type of academic research. *First Monday* 4(10). Available at: http://journals.uic.edu/ojs/index.php/fm/article/view/696/606 (accessed 30 May 2017).

Black Duck Inc (2017): *Top Open Source Licenses*. Available at: https://www.blackducksoftware.com/top-open-source-licenses (accessed 30 May 2017).

Boes A, Kämpf T, Langes B, et al. (2017) Cloud & crowd: New challenges for labour in the digital society. *TripleC* 15(1): 132–147.

Brooks F (1975) *The Mythical Man-Month*. Reading: Addison-Wesley.

Byfield B (2013) What makes for a community distribution? *Linux Magazine*, 14 Jul 2013.

Coleman G (2013) *Coding Freedom*. Princeton: Princeton University Press.

Connell C (2000) Open source projects manage themselves? Dream on. In: *IBM/Lotus Developers Network*. Available at: http://www.chc-3.com/pub/manage_themselves.htm (accessed 30 May 2017).

Corbet J (2015) Development activity in LibreOffice and OpenOffice. In: *LWN.net*. Available at: https://lwn.net/Articles/637735/ (accessed 30 May 2017).

Corbet J and Kroah-Hartman G (2016) *Linux Kernel Development Report*. San Francisco: The Linux Foundation.

Corbet J, Kroah-Hartman G and McPherson A (2009–2015) *Linux Kernel Development Report*. San Francisco: The Linux Foundation.

Dahlander L and Magnusson M (2008) How do firms make use of open sourcecommunities? *Long Range Planning* 41(6): 629–649.

Dickel S and Schrape JF (2017) The Logic of Digital Utopianism. *Nano Ethics* 11(1): 47–58.

Dolata U (2017) Apple, Amazon, Google, Facebook, Microsoft. Market Concentration – Competition – Innovation Strategies. *Research Contributions to Organizational Sociology and Innovation Studies 2017-01*. Stuttgart: University of Stuttgart, Institute for Social Sciences.

Dolata U and Schrape JF (2016) Masses, crowds, communities, movements: collective action in the internet age. *Social Movement Studies* 15(1): 1–18.

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

Eclipse Foundation (2016) *2016 Annual Eclipse Community Report*. Available at: http://www.eclipse.org/org/foundation/reports/annual_report.php (accessed 30 May 2017).

Fisher FM, McKie JW and Mancke RB (1983) *IBM and the US Data Processing Industry*. Santa Barbara: Praeger.

Fuchs C (2015) *Culture and Economy in the Age of Social Media*. London: Routledge.

Free Software Foundation (1989) *GNU General Public License (GPL) Version 1.0*. Available at: http://www.gnu.org/licenses/old-licenses/gpl-1.0.en.html (accessed 30 May 2017).

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

Geels FW and Schot JW (2007) Typology of sociotechnical transition pathways. *Research Policy* 36(3): 399–417.

Gelsi S (1999) VA Linux rockets 698%. *CBS Marketwatch*, 10 Dec 1999. Available at: http://www.cbsnews.com/news/va-linux-rockets-698/ (accessed 30 May 2017).

GNU Project (2017) *GCC Steering Committee*. Available at: https://gcc.gnu.org/steering.html (accessed 30 May 2017).

Gonzalez-Barahona JM, Izquierdo-Cortazar D, Maffulli S, et al. (2013) Understanding how companies interact with free software communities. *IEEE Software* 30(5): 38–45.

Greenstein S and Nagle F (2014) Digital dark matter and the economic contribution of Apache. *Research Policy* 43(4): 623–631.
Gulley N and Lakhani K (2010) The Determinants of Individual Performance and Collective Value in Private-Collective Software Innovation. *Harvard Business School TOMU* Working Paper 10/065.
Hayes F (2001) The Microsoft way. *Computerworld* 3/19/2001: 78.
Henkel J, Schöberl S and Alexy O (2014) The emergence of openness: How and why firms adopt selective revealing in open innovation. *Research Policy* 43(5): 879–890.
Héritier A (2008) Causal explanation. In: Della Porta D and Keating M (eds) *Approaches and Methodologies in the Social Sciences*. New York: Cambridge University Press, pp. 61–79.
Holtgrewe U and Werle R (2001) De-commodifying software? *Science Studies* 14(2): 43–65.
Jaeger T (2010) Enforcement of the GNU GPL in Germany and Europe. *Journal of Intellectual Property, Information Technology and E-Commerce Law* 1(1): 34–39.
KDE Project (2017) *Project Management*. Available at: https://www.kde.org/community/whatiskde/management.php (accessed 30 May 2017).
Kernel.Org (2016) *How to Get Your Change Into the Linux Kernel*. Available at: https://www.kernel.org/doc/Documentation/SubmittingPatches (accessed 30 May 2017).
Kolassa C, Riehle D, Riemer P, et al. (2014) Paid vs. volunteer work in open source. *In:* Proceedings 47th Hawaii Conference on System Sciences, pp. 3286–3295.
Kostakis V, Roos A and Bauwens M (2016) Towards a political ecology of the digital economy. *Environmental Innovation and Societal Transitions* 18: 82–100.
Krafft MF (2010) *A Delphi Study of the Influences on Innovation Adoption and Process Evolution in a Large Open Source Project: The Case of Debian*. Dissertation. Limerick: University of Limerick.
Kreiss D, Finn M and Turner F (2011) The limits of peer production: Some reminders from Max Weber for the network society. *New Media and Society* 13(2): 243–259.
Lakhani K and Hippel E (2003) How open source software works. *Research Policy* 32(6): 923–943.
Larsson R, Bengtsson L, Henriksson K, et al. (1998) The interorganizational learning dilemma: Collective knowledge development in strategic alliances. *Organization Science* 9(3): 285–305.
Lerner J (2012) *The Architecture of Innovation*. Boston: Harvard Business Press.
Lerner J and Tirole J (2002) Some simple economics of open source. *Journal of Industrial Economics* 50(2): 197–234.
Lerner J and Schankerman M (2010) *The Comingled Code*. Cambridge: MIT Press.
Lessig L (1999) Open code and open societies. *Chicago Kent Law Review* 74: 1405–1420.
Levine P (2014) Why there will never be another Red Hat: The economics of open source. *Techcrunch*, 13 Feb 2014. Available at: http://on.tcrn.ch/l/pjXf (accessed 30 May 2017).
Levy S (1984) *Hackers: Heroes of the Computer Revolution*. Garden City: Anchor Press.
Mayntz R (2004) Mechanisms in the analysis of social macro-phenomena. *Philosophy of the Social Sciences* 34(2): 237–259.
McCray P (2013) *The Visioneers*. Princeton: Princeton University Press.
Menell PS (2002) Envisioning copyright law's digital future. *New York Law School Review* 46: 63–199.
Microsoft Inc. (2017) 2016 Annual Report. Available at: http://www.microsoft.com/investor/reports/ (accessed 30 May 2017).
Miller P and Nelson LE (2016) *Open source powers enterprise digital transformation*. Research report. Cambridge: Forrester Inc.
Moody G (2002) *Rebel Code*. New York: Basic Books.
Mozilla Foundation (2016) *Mozilla Wiki WeeklyUpdates*. Available at: https://wiki.mozilla.org/WeeklyUpdates (accessed 30 May 2017).
NetApplications Inc (2017) *Market share reports*. Available at: http://www.netmarketshare.com (accessed 30 May 2017).
Netscape Communications Inc (1998) *Netscape Announces Mozilla.org*. Press Release, 23 Feb 1998.

Nuvolari A (2004) Collective invention during the British industrial revolution: The case of the Cornish Pumping Engine. *Cambridge Journal of Economics* 28(3): 347–363.

O'Mahony S (2003) Guarding the commons. How community managed software projects protect their work. *Research Policy* 32(7): 1179–1198.

O'Mahony S and Ferraro F (2007) The emergence of governance in an open source community. *Academy of Management Journal* 50(5): 1079–1106

Osterloh M and Rota S (2007) Open source software development: Just another case of collective invention? *Research Policy* 36(2): 157–171.

Perrow C (1991) A society of organizations. *Theory and Society* 20: 725–762.

Pomerantz J and Peek R (2016) Fifty shades of open. *First Monday* 21(5). Available at: http://dx.doi.org/10.5210/fm.v21i5.6360 (accessed 30 May 2017).

Powell W and Giannella E (2010) Collective invention and inventor networks. In: Hall BH and Rosenberg N (eds) *Handbook of the Economics of Innovation*, Vol.1. Oxford: Elsevier, pp. 575–605.

Raymond ES (1999) *The Cathedral and the Bazaar*. Sebastopol: O'Reilly.

Raymond ES (1998a) Homesteading the Noosphere. *First Monday* 3(10). Available at: http://firstmonday.org/article/view/621/542 (accessed 30 May 2017).

Raymond ES (1998b) Goodbye, "free software"; hello, "open source." Announcement, 22 Nov 1998. Avaialble at: //ftp.lab.unb.br/pub/computing/museum/esr/open-source.html (accessed 30 May 2017).

Raymond ES (1997) *The Cathedral and the Bazaar, Revision 1.16*. Available at: https://archive.org/stream/CathedralAndTheBazaar/cathedral-bazaar_djvu.txt (accessed 30 May 2017).

Rifkin J (2014) *The Zero Marginal Cost Society*. New York: Palgrave Macmillan.

Ritzer G, Dean P and Jurgenson N (2012) The coming of age of the prosumer. *American Behavioral Scientist* 56(4): 379–398.

Scharpf FW (1997) *Games Real Actors Play*. Boulder: Westview Press.

Schrape JF (2016) *Open-Source-Projekte als Utopie, Methode und Innovationsstrategie. Historische Entwicklung – sozioökonomische Kontexte – Typologie*. Glückstadt: Hülsbusch.

Scott WR (2004) Reflections on a half-century of organizational sociology. *Annual Review of Sociology* 30(1): 1–21.

Shaw A and Hill BM (2014) Laboratories of oligarchy? How the iron law extends to peer production. *Journal of Communication* 64(2): 215–238.

Spencer JW (2003) Firms' Knowledge-sharing Strategies in the global innovation system. Evidence from the flat panel display industry. *Strategic Management Journal* 24(3): 217–233.

Spinellis D and Giannikas V (2012) Organizational adoption of open source software. *Journal of Systems and Software* 85(3): 666–682.

Spreeuwenberg K and Poell T (2012) Android and the political economy of the mobile internet. *First Monday* 17(7). Avaialble at: http://dx.doi.org/10.5210/fm.v17i7.4050 (accessed 30 May 2017).

Stallman R (2002) *Free Software, Free Society*. Boston: GNU Press.

Stallman R (1983) *New UNIX implementation*. Available at: http://bit.ly/1DSDoXW (accessed 30 May 2017).

Stamelos I (2014) Management and coordination of free/open source projects. In: Ruhe G and Wohlin C (eds) *Software Projekt Management in a Changing World*. New York: Springer, pp. 321–341.

StatCounter Inc (2017) *Global Stats*. Available at: http://gs.statcounter.com/ (accessed 30 May 2017).

Steinberg J (2014) Massive Internet security vulnerability. *Forbes*, 10 Apr 2014. Available at: https://www.forbes.com/sites/josephsteinberg/2014/04/10/massive-internet-security-vulnerability-you-are-at-risk-what-you-need-to-do/ (accessed 30 May 2017).

Steinmacher I, Conte T, Redmiles D, et al. (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 and Social Computing*: 1379–1392.

Stiller A (2011) The open source trials. In: *Vision Mobile Blog*. Available at: http://www.visionmobile.com/blog/2011/03/the-open-source-trials-hanging-in-the-legal-balance-of-copyright-and-copyleft/ (accessed 30 May 2017).

Suddaby R (2013) Review: The Janus face of commercial open source software communities. *Organization Studies* 34(7): 1009–1011.

Sundararajan A (2016) *The Sharing Economy*. Cambridge: MIT Press.

Tapscott D and Williams AD (2006) *Wikinomics*. New York: Portfolio.

Teixeira J, Robles G and Gonzalez-Barahona JM (2015) Lessons learned from applying social network analysis on an industrial free/libre/open source software ecosystem. *Journal of Internet Services and Applications* 6(1): 1–27.

Torvalds L (1998) LINUX manifesto. Interview. *Boot Magazine* 1998(7/8): 32–37.

Torvalds L (2002) Re: [PATCH] Remove bitkeeper documentation from Linux tree. In: *Linux Kernel Mailinglist*. Available at: http://lwn.net/2002/0425/a/ideology-sucks.php3 (accessed 30 May 2017).

Ubuntu Project (2017) *Governance*. Available at: https://www.ubuntu.com/about/about-ubuntu/governance (accessed 30 May 2017).

Van de Ven AH, Delbecq AL and Koenig Jr. R (1976) Determinants of coordination modes within organizations. *American Sociological Review* 41(2): 322–338.

W3techs Surveys (2017) *Technologies Overview*. Available at: http://w3techs.com/technologies (accessed 30 May 2017).

Weber S (2000) *The political economy of open source*. BRIE Working Paper 140. Berkeley: University of California.

Weinberg B (2015) The Internet of Things and open source. In: Podnar Žarko I, Pripužić K and Serrano M (eds.) *Interoperability and Open-Source Solutions for the Internet of Things*. Cham: Springer, pp. 1–5.

West J, Salter A, Vanhaverbeke W, et al. (2014) Open innovation: The next decade. *Research Policy* 43(5): 805–811.

Westenholz A (ed) (2012) *The Janus Face of Commercial Open Source Software Communities*. Copenhagen: Copenhagen Business School Press.

Author biography

Jan-Felix Schrape is Senior Researcher at the Department of Organizational Sociology and Innovation Studies, Stuttgart University (Germany). He has published on changes in media structures and the actors of innovation processes.