

This is the author's version of a chapter accepted for publication in the *Handbook of Peer Production*. Changes resulting from the publishing process such as copy-editing, typesetting, and other quality control mechanisms may not be reflected in this document. This author manuscript version is available for personal, non-commercial and no derivative uses only.

Citation: Couture, S. (2021). Free & Open Source Software. In: M. O'Neil, C. Pentzold & S. Toupin (Eds.), *The Handbook of Peer Production* (pp. 155-168). Malden, MA: Wiley-Blackwell.

ISBN 9781119537106 Available at: <https://www.wiley.com/en-au/The+Handbook+of+Peer+Production-p-9781119537090>

The Handbook of Peer Production

Chapter 12 – Free & Open Source Software

Stéphane Couture, Université de Montréal, Canada

1. Introduction¹

Free software, often used interchangeably with the term “open source software” or “free and open software” (F/OSS), is software whose source code can be freely accessed, shared, modified, and improved. Because of these characteristics that enable the free access and sharing of knowledge, free and open source software has given rise to new models of software development based on peer production practices, that may involve hundreds, if not thousands, of developers. For many authors, free and open source software is often seen as a prime example, if not the archetype, of the large-scale communal production of information, knowledge, and culture (Bauwens et al., 2012; Benkler, 2006, p. 5).

Today, free and open source software is used in multiple applications and forms much of the backbone of Internet infrastructures, like web or mail servers, or the core of operating systems like Android and Mac OSX. Beyond technologies, free and open source software may also be seen as a sharing culture, a community of actors collaborating online, and as a political movement for the freedom of knowledge on the Internet. Introduced in the 1980s in opposition to the proprietary software model, the success of free and open source software has garnered commercial interest to such an extent that many companies such as IBM, RedHat, Oracle and more recently Google and even Microsoft have focused their business strategy on supporting and taking advantage of the forms of collaboration at work in open source software communities. Public administrations around the world have also embraced free and open source software as a way to reduce costs or assert their

¹ This chapter is based on previous work of the author on the subject (Couture, 2007, 2013, 2015). Thanks to Geneviève Szczepanik for the grammatical revisions.

sovereignty, and many social activists use it to develop autonomous infrastructures. Finally, these collaborative and moral dimensions of the free and open source movement have inspired many other “peer-production” projects, the most famous of them probably being Wikipedia, the “free encyclopedia.”

This chapter first considers free and open source software from a historical perspective, by examining how these terms came about and are still sometimes opposed to each other. It then describes how free and open source software “works,” that is, the collaborative and legal practices on which some of the most successful free and open source software projects rely. Then, it presents the mainstream adoption of free and open source software, and ends by addressing some of the challenges this peer production model faces, in particular when related to tensions arising from its marketization as well as from participants’ lack of diversity.

2. Genesis of a Movement

2.1 The hacker ethic at the origin of free software

It is generally recognized that the free and open source software movement originated in the hacker culture of the early 1960s. While the meaning of the term “hacker” is today often associated with illegal practices, the term originally referred – and still refers within many free and open source communities – to what Coleman (2012) describes as an ethic and aesthetics of tinkering (Coleman, 2012), similar to that described by the metaphor “art for art’s sake,” where technical activity is not a means or an instrument, but rather an end in itself (Riemens, 2006). Citing Levy (1985), Coleman writes that people

who called themselves hackers in the 1970s “placed the desire to tinker, learn, and create technical beauty above all other goals” (Coleman, 2012, p. 17)².

Raymond (2000), in his brief *history of hackerdom*, situates the beginnings of hacker culture in the 1960s, more specifically at the Tech Model Railroad Club (TMRC), an MIT student association still active today in which people tinker with model railroads and other technical objects, including computers. Many members of the TMRC at the time were also members of the MIT Artificial Intelligence Laboratory, which facilitated the spread of “hacker culture” in the realm of computer programming. Several authors (Coleman, 2012; Himanen, 2001; Levy, 1985) have attempted to explore this hacker culture whose importance to cooperation and information sharing has played, according to Manuel Castells (2003), a decisive role in the development of the Internet and its potential for innovation.

Steven Levy (1985) identified several values associated with the hacker ethic, one being the idea that “All information should be free.” Consistent with this value, hackers attached great importance to sharing the software and computer programs they created. However, with the reduction of hardware costs in the late 1970s, many computer corporations began to restrict this sharing of software, in order to increase their profits. Bill Gates, today at the head of one of the largest fortunes in the world, said: “The royalty paid to us, the manual, the tape and the overhead make it a break-even operation. One thing you do is prevent good software from being written. Who can afford to do professional work

² Beyond this basic characterization of hacking, Coleman insists however on the diversity of hacking genres (Coleman, 2012, p. 18; Coleman & Golub, 2008), the main one being the difference between F/OSS hackers, who are oriented towards transparency and collaboration, and the “hacker underground,” that is more oriented towards secrecy and spectacle.

for nothing?” (Gates, 1976). This gave rise to the “proprietary” software model we know today, software where use and sharing of source code are restricted.

While many hackers decided to subscribe, often unwillingly, to this view of proprietary software development, others sought to develop technological alternatives based on sharing software and its source code. The most radical initiative proposed as an alternative to “proprietary” software was the creation of the concept of free software by Richard Stallman, whom Steven Levy (1985) described in his book as the “last of the hackers.”

2.2 The GNU project and the free software foundation

In the late 1970s, hackers at MIT’s Artificial Intelligence Laboratory had formed their community around the handling of a minicomputer and its co-developed operating system. After purchasing a new printer in 1982, the Laboratory’s authorities decided to use a proprietary operating system, the license for which prohibited the modification of its source code. For Richard Stallman, a member of this hacker community, these new rules, being incompatible with the software-sharing ethic of the hacker community, were unacceptable and antisocial: “This meant that the first step in using a computer was to promise not to help your neighbor. A cooperating community was forbidden” (Stallman, Lessig, & Gay, 2002, p. 13). Rather than subscribing to this proprietary model, Stallman decided to create a new computer system, based only on free software, with the hope that a new sharing community would emerge around it. Three crucial milestones are attributed to Stallman, which significantly contributed to booting up the free software movement:

First, the project of a new completely free computer system was launched. The project, called *GNU* which stands for Gnu is Not Unix, a recursive acronym (popular in hacker humor) referring to the idea of a system that would be *like* a Unix system, but non-

proprietary. The goal was to build an alternative to the then very popular Unix operating system, but that could be freely modified, used, studied, or redistributed. Stallman started to build the system with some software he was already developing, and made a call to contribute to the project with other pieces of software. The initial announcement of this project was on September 27, 1983 on a Usenet Group.

Second, the Free Software Foundation was created two years later, in 1985, foremost in order to finance and support the development of the GNU project and other “Free Software.” This Foundation still exists today, but its activities have broadened beyond software development, to encompass advocacy activities involved in free software and related issues.

Third, Stallman, along with the Free Software Foundation, published in 1989 the first version of the GNU General Public License (GPL), a licensing agreement guaranteeing the freedom for users to use, change, and modify software, unlike many proprietary software licenses that aim to restrict the use of software (to a single machine for instance). This provided a strong legal framing for the creating and sharing of free software. Since then, three major versions of the GPL have been published, and these different versions are still in use in many free and open source software projects.

So what exactly is “Free software?” While the formal definition has changed over time, its modern definition is often described using the ideas of four freedoms that users have in relation to software: using, studying, copying, improving (see table 1). The free software definition is often characterized by the idea of *Copyleft*, a play on words to signify an inversion of copyright: instead of using copyright to restrict the use and sharing of a work, it is used to guarantee this very possibility. One motto often stated by free

software advocates is that free software is “free as in freedom, not as in free beer”, to distinguish it from the idea of “zero price.”

Table 1: Free software definition³

A program is free software if the program's users have these four essential freedoms:

- The freedom to run the program as you wish, for any purpose (freedom 0).
- The freedom to study how the program works, and change it so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help others (freedom 2).
- The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.

An important aspect in this definition of free software is the idea of “access to source code,” as described in “freedom 1” and “freedom 3.” Source code can generally be described as the human-readable instructions that specify the functioning of software⁴. Source code is what programmers make and interact with when they create or modify

³ <https://www.gnu.org/philosophy/free-sw.html> (retrieved May 2nd, 2019)

⁴ For a more complex and situated analysis of this notion, see Couture (2019).

software. The idea of free software is not that the “executable” source code should be free (as in free beer), but that everybody should be allowed to access and modify the source

```
/**
 * Simple HelloButton() method.
 * @version 1.0
 * @author john doe <doe.j@example.com>
 */
HelloButton()
{
    JButton hello = new JButton( "Hello, wor
    hello.addActionListener( new HelloBtnList

    // use the JFrame type until support for t
    // new component is finished
    JFrame frame = new JFrame( "Hello Button"
    Container pane = frame.getContentPane();
    pane.add( hello );
    frame.pack();
    frame.show();           // display the fra
}
```

code. Indeed, source code is sometimes conceptualized by free software advocates as a form of speech and as such, sharing source code is considered as freedom of speech (Byfield, 2006; Coleman, 2009).

Figure 1: Source code (Source: Wikipedia)

It is important to emphasize that the idea of free software is foremost grounded in the moral principle of sharing. This quote from an interview given by Stallman captures the essence of this ethical principle:

For me, free software is above all a question of freedom and community. We need free software so that computer users are free to cooperate. It is for this reason alone that I decided to reject the nonfree software. That free software also leads to efficient and powerful software has been a surprise to me, and I am delighted. But it's a bonus. I would have chosen free software, even if it had been less efficient and less powerful – because I do not sell my freedom for simple matters of convenience (Gleizes & Papathéodou, 2000; free translation).

The development of free software continued throughout the 1980s and mobilized several hundreds of programmers who collaborated through various telecommunication networks of the early Internet. This gave rise to a new model of software development that later sparked the interest of commercial firms.

2.3 Disseminating to companies: The creation of “Open Source”

On August 25, 1991, Linus Torvalds, a Finnish student, launched a call for contributions for a new project he called *Linux*. In the months that followed, this call received growing interest from many free software developers, not least because Linux was the missing piece of the GNU project that started some years earlier and finally saw the establishment of a completely free computer system. Torvalds' leadership and charisma, combined with momentum in the development of the Internet catalyzed the free software community's efforts to develop a product that would soon reach a certain maturity.

Enthralled by the success of Linux, Eric Raymond wrote an essay in 1997, later published as a book (Raymond, 2001), titled *The Cathedral and the Bazaar*. It is considered by some to be the first sociological analysis of free and open source software

collaboration and development. In his essay, Raymond describes the software development style of Linux as a “bazaar” where the goal is foremost to create a community of contributors to a bazaar of ideas, and to choose from this bazaar the best ideas and integrate them in the new software. The metaphor of the ‘bazaar’ is contrasted with that of the ‘cathedral’ where software is developed in a top-down matter by a grand architect⁵.

This perspective is of great ideological significance. Unlike the previous legitimization of free software, Raymond does not attach any ethical importance to software freedom or sharing. It is to the extent that free software allows the open participation of a large number of programmers and users in the bazaar model that software must remain free to access. What is then at stake is simply the technological and managerial success of a new form of software development described as “the bazaar.” This discourse, devoid of moral concerns, was much more appealing to private firms, compared to Stallman’s moral variant. Inspired by Raymond’s perspective, Netscape announced in 1998 its intention to publish the source code of its web browser to obtain the collaboration of the free software movement, so they could destabilize their competitor Microsoft (William, 2002, p. 165). This would later become today’s Firefox browser. Following this decision, several business leaders and advocates decided to overcome the ambiguity, in the English language, of the terms “free” and “free software,” that are too often associated with lack of cost. This is why Christine Peterson, the chair of a Silicon Valley think tank, advanced the term “Open Source” to replace “Free Software” in order to make it more friendly to commercial business (Williams, 2002).

5 *The Cathedral and the Bazaar* is often understood as a comparison between free and open source software, and proprietary software. However, it is rather Stallman’s style of software development that is associated with the cathedral style in this essay.

The creation of open source has, however, created some controversies. For Stallman and many of his followers, “The Free Software movement and the Open Source movement are like two political camps within the free software community” (Stallman, Lessig, & Gay, 2002 p. 43). Many proponents still insist today on using the term “free software,” implying that open source, while proving technically successful, is erasing the moral dimension of their project. The acronyms FLOSS (Free/Libre Open Source Software), or FOSS (Free and Open Source Software) are sometimes used, especially by researchers, to grasp the phenomenon as a whole while avoiding taking a position in the controversy.

3. How it Works: The Economy and Sociology of Free Open Source Software

In an article addressing “bottom-up innovation”, French sociologist Dominique Cardon (2005) characterized free and open source software as having two axes of “coherence:” a normative axis, centered around hacker ethics and the value of sharing knowledge, and an organizational axis, first described by the “bazaar” metaphor, put forward by Raymond. In this section, I will go through these two “axes of coherence” and then present a survey of some of the research to date on free and open source software.

3.1 Sharing code: the legal and normative model of free and open Source software

As I have mentioned before, free software was born foremost within the hacker value of freely sharing information. While this ethos might seem less prominent today, it is still an important part of the engagement within free and open source software. This ethos is actualized legally through different software licenses that frame the possibilities to use and share source code, each of them being more or less permissive concerning the integration of free or open source software and proprietary software.

The most radical license innovation is the notion of “copyleft,” which refers to granting the right to use, modify or improve some works and distribute them, *under the same conditions as the original terms*. The most notable expression of the idea of copyleft is the Gnu General Public License, created by Richard Stallman, that I have described earlier, which allows the sharing of source code but forbids its integration in proprietary software.

Other permissive licenses have been created, in particular to favor collaboration with private firms. They are sometimes considered as “non-copyleft” licenses. For instance, in contrast to the GPL, the *MIT License* permits the modification of source code, but also its integration into proprietary software. This means that it is also possible to integrate MIT-licensed source code under the GPL, but not the other way around. The MIT license is currently the most used license on Github, a popular open source hosting platform (see next table).

Table 2: Most used licenses on Github (April 2nd, 2019)

Rank	License	% of projects
1	MIT	44.69%
2	Other	15.68%
3	GPLv2	12.96%
4	Apache	11.19%
5	GPLv3	8.88%

Source: (Balter, 2015)

Other innovative license models have appeared, inspired by the models of free and open source. For instance, the idea of adding a license granting access to source code, except for military use (a so-called “non-military clause”), has been discussed in the past⁶.

However, proponents of free and open source software have been reluctant to embrace this approach as it goes against the right to access and modify source code “for any use.” Similarly, the Creative Commons “non-commercial” license which permits redistribution of the work for non-commercial purpose only, is considered a non-free software license as it restricts the rights of some users (i.e. commercial ones).

3.2 The bazaar: The sociology of free and open source software

Several research projects, sometimes inspired by *The Cathedral and the Bazaar*, have explored the forms of network collaboration within free and open source software and sought to highlight its innovative character. Some works try to capture the motivations that lead people to become involved in its development (Lakhani & Wolf, 2003) or try to grasp the conditions allowing for the development of a network innovation. Eric von Hippel, a researcher at MIT, has extensively studied the development of free and open source software, which he characterizes as a case of “innovation by use” or “innovation by user networks” (von Hippel, 2005; 2007). Cardon (2005) has reused this model, which he calls “bottom-up innovation” (or “innovation ascendante,” in French), to distinguish it from the “top-down” innovation that would come from laboratories or large software firms.

Contrary to the usual business model of proprietary software in which developers are paid

6 <https://www.linux.com/news/open-source-project-adds-no-military-use-clause-gpl>

for their work through the sale of software, free and open source software is instead based on massive collaboration on the Internet, proceeding from contributions from users or volunteers.

In the most mature software, development is based on the sometimes minimal contributions (a comment, a “patch”) from what may be several thousands of developers. Cardon distinguishes three levels of contributors in bottom-up innovation networks. First, on the inside, the “core of innovators” who are often the ones who created the project or are leading it. Second, on the outside, the “nebula of contributors,” each of whom make small contributions. And, third, in between, the “circle of reformers,” who act as mediators between the previous categories by selecting the best contributions and rearranging them so they fit with the vision of the innovators. This is what Cardon calls the “organizational coherence” of free and open source software.

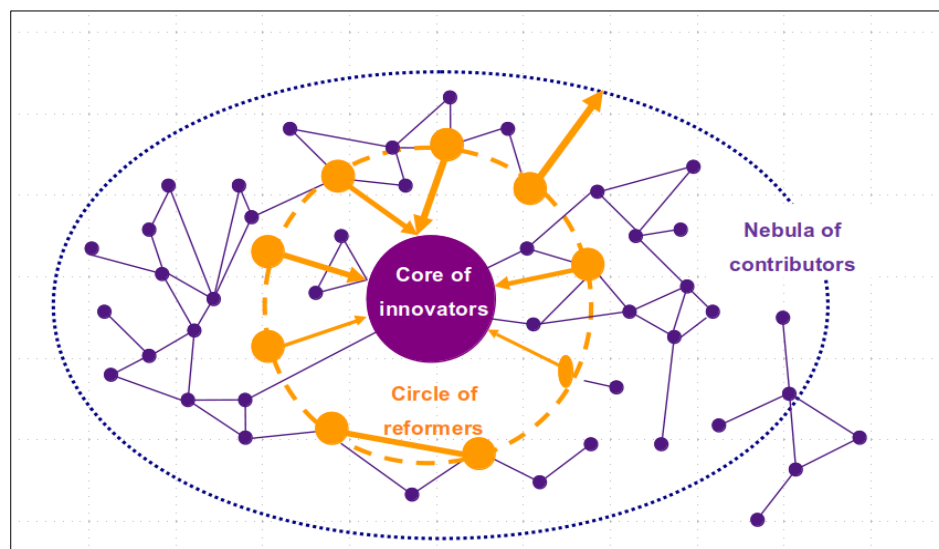


Figure 2: Bottom-up innovation in free software development (Cardon, 2005)

Some authors and analysts also characterize free software as *a Commons* (Aigrain, 2005; Benkler, 2003; Birkinbine, 2014; Boyle, 2008; Schweik, 2007). This

characterization sometimes refers to Elinor Ostrom's work on the economy and governance of commons, which studied institutional forms that rely on common resources rather than exclusive use, which typify property relations⁷. Indeed, Ostrom, along with Charlotte Hess, published an edited book titled *Understanding Knowledge as a Commons*, which includes a chapter that argues that free and open source software could serve as a model for collaboration built around intellectual property (Schweik, 2007). Philippe Aigrain, for his part, considers the free and open source software movement as a key moment in the development of the paradigm of "informational commons," as opposed to that of "information as property" (Aigrain, 2005). Benkler (2006), in his book *The Wealth of Networks*, coined the term *commons-based peer production* when analyzing emerging collaborative forms such as free software as well as Wikipedia. These commons practices, for Benkler, constitute a new economic paradigm that emerges at the heart of the capitalist economy.

It should be noted here that although free and open source software can be characterized as commons-based, it doesn't mean that all contributors work for free or solely on a voluntary basis. In many cases, companies such as IBM, Sun or more recently Google or even Microsoft, play a leading role in securing the salaries of some contributors. In 2006, a study estimated that more than 500,000 *employees* worldwide were working on open source projects (Ghosh, 2006). However, in all cases, the software subsequently created remains "free" and is not subject – in theory – to any restrictions of use. The idea is then that sharing and commons-based production should serve the mutual interests of

7 Although Ostrom has most studied "natural commons" (such as water or land), one of her last books with Charlotte Hess addresses knowledge commons, including free and open source software (Hess & Ostrom, 2007).

concerned actors, whether they are individuals, companies, or non-profit organizations. At least, this is the ideal that, in reality, can create some tensions as we will see in section 5.1.

4. Beyond Experimentation: Mainstream Adoption and Social Significance

In this section, I will briefly discuss the current place of free software, which now extends to different social groups and fields of activity.

4.1 Free and open source as the “glue” of the Internet

Contemporary free and open source software has reached a level of maturity that makes it a good alternative to many popular end-user software programs: computer systems like Ubuntu or Red Hat (called “distributions”) instead of Windows or MacOS; LibreOffice instead of MS-Office, Chromium instead of Chrome. But most importantly, free and open source software is also an important part of the Internet infrastructure. This might be difficult to grasp, as infrastructure is by definition most of the time invisible to common users (Star, 1999). However, free and open source software is used in servers for sending emails, running websites or creating web applications.

For example, Table 3 shows that two free and open source web servers – Apache and Nginx – are used in more than 80% of all web servers. The same goes for “Server-side Programming Languages” for which PHP, licensed as free software, accounts for almost 80% of the market. Finally, we should also note that Android is, by far, the most popular Mobile Operating System. While we tend to associate Android with Google – and indeed Google strongly controls it, as we will see below – it is important to remember that Android is actually a piece of software that is, in part, licensed under the Gnu General Public License (GPL). In short, one can argue that free and open source software is the “glue” that holds the Internet together.

Table 3: Free and open source software as the Internet infrastructure

Server-side programming languages		Most popular web servers		Most popular mobile OS	
<u>PHP</u>	79.0%	<u>Apache</u>	43.8%	Android	86,8%
<u>ASP.NET</u>	11.4%	<u>Nginx</u>	41.6%	iOS	13,2%
<u>Java</u>	4.0%	<u>Microsoft-IIS</u>	8.7%		
<u>Ruby</u>	2.5%	<u>LiteSpeed</u>	4.0%		

Source: W3Techs.com. 1 April 2019 and (for Android):

<https://www.idc.com/promo/smartphone-market-share/os>

In the next sections, I will describe in more detail some of the aspects of free and open source software.

4.2 Public policies and governmental adoption

The success of the development of free and open source software has prompted many firms and public bodies to consider adopting them from an end-user perspective. From the beginning of the millennium, studies have found free and open source software to be economically and technologically viable, and several governments have established policies favoring free and open source software or compelling its adoption outright. In the 2000s, the Center for Strategic and International Studies routinely updated a report on these policies around the world. The last version (Lewis, 2010) listed 354 policy initiatives that had been proposed or approved worldwide since 2001 regarding free open source software, divided into four categories: research (looking at the possibilities of using open source), mandatory (requiring the use of open source software), preferences (preferring

open source software), and advisory (permitting its use). The significant savings offered by free and open source software are often the main political rationale for adopting these policies, but other arguments are put forward that emphasize public responsibility, such as the adaptability of free and open source software to meet the particular needs of public administration, the increased security of free and open source software that allows the possibility of analyzing its source code, and technological independence with regards to a particular technology or provider (e.g. Microsoft or Oracle) (Oram, 2011).

This last concept, *technological independence*, sometimes framed as “digital sovereignty,” is also emphasized in several studies insisting on the capacity of states to ensure their sovereignty. In 2013, the French Minister for the Digital Economy also noted that free and open source software was a real guarantee of the “digital sovereignty” of a country (Pellerin, 2013). This term is additionally being used to support the need to implement a digital industrial policy aimed at counteracting the hegemony of the United States of America (USA) in the sector.

Several so-called “mandatory” policies, sometimes involving massive migration to free and open source software, have been adopted. The following table summarizes several European experiences of free and open source software adoption. Note that several countries in Latin America have also adopted free and open source software, especially Brazil, which was for a few years recognized as a leader in the field, especially at the beginning of Lula da Silva’s leftist government (Benson, 2005; Bollier, 2011).

Table 4: Some experiences of migration to free and open source software in Europe

Year	Organization	Details

2003	Finnish Union of Practical Nurses	300 migrations to GNU/Linux and OpenOffice.
2003	French Ministry of Interior	15 000 migrations to OpenOffice.
1997 - 2005	The Spanish region of Extremadura	100 000 computers using <i>gnuLinEx</i> , a version of Linux
2005	The Dutch city of Haarlem	2000 migrations to OpenOffice.
2005	French Customs	16 000 migrations to OpenOffice
2005	Bristol City Council in the United Kingdom	5500 migrations to StarOffice
2003 - 2010	Ministry of Justice, Finland	10 000 migrations to OpenOffice
2005 - 2013	French Gendarmerie Nationale	70 000 computers partly using OpenOffice, Firefox and Thunderbird

Source: This table copied and adjusted from Couture (2013), itself inspired from Karjalainen (2010).

4.3 Social movements, “tech activists,” and free and open source software

Social movements and so-called “tech activists” have also long been interested in the use of free and open source software. Since its inception, there has always been a discourse that associated – rightly or wrongly – free and open source software (especially the “free software” perspective) with a leftist or progressive ideology. In 1999, Eben Moglen – who later provided significant support and legal advising to the Free Software Foundation – characterized the emergence of free software as “Anarchism triumphant”

(Moglen, 1999). In the early 2000s, Indymedia, a global network of alternative media created as part of the anti-globalization movement, was technically built with free and open source software and influenced by its culture (Coleman, 2004). Free software activists were also always present within the World Social Forum, an annual meeting of progressive and environmentalist social movements around the world (Juris, Caruso, Couture, & Mosca, 2013), though not always in a straightforward way.

Many “tech activists” are still engaged alongside or in support of social movements around the world. Many “radical servers” or alternative infrastructures have been built, like riseup in North America, Koumbit in Canada or “dégooglisons Internet” (de-google Internet) in France, all of which are based on free software and aimed at social movements and non-profit organizations. While infrastructures built by these groups are not as extensively developed as Google’s commercial products, they serve as cultural resources or a “concrete utopia” (Broca, 2013) to participate in global discussions about technical choices. Activist-researcher Alex Haché has proposed to use the term “technological sovereignty” to describe efforts by civil society to develop autonomous infrastructures and protect itself from commercial and governmental surveillance, notably by using free and open source software (Haché, 2017; Couture & Toupin, 2019).

4.4 Studying free and open source software: The proliferation of academic work

Numerous academic articles and research projects have been produced about free and open source software. For instance, a search on ProQuest Central for the term “open source” between 2000 and 2018 yielded 73,956 peer-reviewed articles, while a search for the term “free software” for the same period yielded 9,093 peer-reviewed articles, and “free and open source software” yielded 1,121 articles. While many of these articles have

not been produced from a social science perspective, these numbers show the significant academic interest that this phenomenon evokes.

Early research on free and open source software has, for instance, studied the economics of free and open source software, and analyzed why developers and businesses sometimes contribute for free (Lerner & Tirole, 2000). Other research has focused on the cultural significance of free and open source software by looking at the hacker ethics and values that permeate free and open source software communities (Auray, 2000; Coleman, 2012; Broca, 2013). Others have looked at the phenomenon from the perspective of knowledge construction, for instance by analyzing conventions and forms of regulation that ensure their cohesion (Demazière, Horn, & Zune, 2007), or at the diversity of contexts in which free and open source software is deployed (Lin, 2005). In the mid-2000s, Tuomi (2004) and Lin (2005) argued – and criticized – that much of the research into free and open source software was based on a conception of a homogeneous free and open source software community with relatively shared values and motivations. Other authors have analyzed, from a critical perspective, the link between the production of free and open source software and contemporary capitalism (Birkinbine, 2014; Coris, 2006; Dafermos & Söderberg, 2009; Moulrier-Boutang, 2007). Similarly, other researchers have sought to capture the conditions that allow the appropriation of free and open source software in non-technical environments (Jullien & Zimmerman, 2007).

5. Challenges

The development of free and open source software faces several challenges. I will present two of them: the tension between capitalist appropriation and community sustainability; the weak participation of women and broader lack of diversity.

5.1 Capitalist appropriation vs community sustainability

An important challenge that faces free and open software is the opposition between capitalist appropriation of technology and community sustainability. It is important to look at this as a tension, since the problem is not so much that the free and open source software model is not sustainable, but rather that the “community form” of free and open source software is greatly struggling, while its “commercial form” is increasingly articulated in what can be characterized as informational capitalism.

Various analyses have critically studied the role of private companies in free and open source software development. For example, several analyses have looked at how companies succeed in channeling the free work (as in gratis work) of volunteers in order to extract some commercial profit (Broca, 2013; Dafermos & Söderberg, 2009; Terranova, 2000). For Moulier-Boutang, free and open source software can be apprehended as a “signifier” of cognitive and post-industrial capitalism where entrepreneurship consists in “converting the wealth already there in digital space, into economic value” (Moulier-Boutang, 2007, p. 167, as cited by Broca, 2013, p. 244).

Coris (2006, 2009) for her part sees free and open source software as an example of “absorption by the market sphere.” Coris focuses more specifically on the case, in the French context, of what she calls “Free Software Services Companies” (FSSCs) created in opposition to traditional information technology service companies (ITSC)⁸, that did not include free and open source software in their product offerings at the time of their creation. Coris (2009) notes that, on the one hand, “pure” free open software service companies have been unable to strive and change the software industry. At the same time,

⁸ In French : “Sociétés de services en ingénierie informatique” or SSII

free and open source software has been further integrated into pre-existing and more traditional IT service offerings, because of its benefits such as reduced costs and flexibility. In short, the emergence of an industrial context favorable to free and open source software has contributed to quell the early community-oriented approach rather than perpetuate it. In addition, free and open source software that is embedded in “traditional” enterprises is generally limited to that whose maturity and recognition is comparable to similar proprietary software. Essentially, Coris finds that free and open source software has managed to become dominant at the cost of a form of ‘industrialization’ that challenged the initial “community” model: “Born from opposition to the capitalist system – exemplified by the ‘proprietary’ model protection of intellectual property – yet free and open source software seems to succeed: the software companies end up being forced to integrate free software into their offerings. At the same time, free and open source software software services companies are, in turn, forced to industrialize” (Coris, 2006, p. 21; free translation).

5.2 Low participation of women and lack of diversity

Lack of diversity is another important challenge in free and open source communities. The low participation of women, in particular, has been noted several times. In 2001, a study funded by the European Union found that only 1,1% of free and open source software developers were women (Ghosh, Glott, Krieger, & Robles, 2002, p. 8)⁹. Since then, many other studies – whether informal or formal – have echoed this finding.

⁹ It has to be noted that this survey was aimed at “developers” and authors of Free and open source software. As I showed elsewhere (Couture, 2019), the way in which we define software development and contributions to source code can have implications for how we value and give more visibility to one type of work over others. In other words, if the survey had been aimed more broadly at “participants” rather than “developers,” it could have yielded a higher proportion of female participants.

For instance, a survey around the Debian system showed that only 1 out of 1,000 software developers was a woman. Following the European Union study, another follow-up study was conducted to better understand the issue, this time from a qualitative and ethnographic point of view (Nafus, Leach, & Krieger, 2006a, 2006b). The findings of this study emphasized behaviors within free and open source software development that “actively discouraged” women to participate. More recently, in 2017, a survey was created by Github, a software collaboration platform very popular among free and open source practitioners, to better understand the practices and demography of these actors (Github, 2017). Again, this survey found out that only 5% percent of respondents were women, while 22.6% of professional computer programmers in the US are women (Finley, 2017).

Despite these numbers, the situation may have improved in different ways. First, the very fact that research was conducted to study the phenomenon is in itself a positive signal. Second, many initiatives were developed to raise awareness on the low participation of women, and potentially discriminatory or sexist behaviors. Indeed, many female-oriented or explicitly feminist groups have emerged throughout the years for women to mutually support each other, and to encourage others to participate in these projects. LinuxChix¹⁰, for instance, is a group founded in 1999, to bring together and support women interested in developing or using Linux. Another group, Debian-Women was founded in 2004 to actively engage women in the Debian Project¹¹. Feminist hackerspaces have also raised interest in recent years (Savic & Wuschitz, 2018; Toupin, 2014) as sites engaging women in hacker culture and free and open source software development. Many

10 <https://www.linuxchix.org/>

11 <https://www.debian.org/women/>

project leaders have also since taken a step back following accusations or recognition of sexist behaviors. The most famous of them is probably Linus Torvalds, the creator of Linux who, in 2017, decided to take a break from his leadership of the project to reflect on his aggressive behavior (Cohen, 2018). Another famous case is Jacob Applebaum who retired from his leadership in the TOR project after allegations of sexual abuse (Greenberg, 2016). More recently, Richard Stallman himself has resigned from his position as president of the Free Software Foundation and from its board of directors, following a much-criticized message he sent to a mailing list concerning underage victims of sex trafficking (Evangelho, 2019)¹². These actions could be interpreted as proof of enduring sexism and machismo, but they could also be signs of growing intolerance towards such behaviors and thus point to an improvement in terms of openness for women.

In terms of racial, ethnic or linguistic equity, however, much less work has been done to study or tackle the subject. First, there seems to be a hegemonic understanding that computer programming is done in English. Apart from some less popular software projects written in other languages (Couture, 2017), the vast majority of free and open source projects are written in English. Concerning racial and ethnic diversity, the previously cited survey conducted by Github found that “16 percent of respondents said they belonged to ethnic or national groups that are in the minority in the country they live in [...] as

12 Stallman’s message was sent to the MIT Computer Science and Artificial Intelligence Laboratory (CSAIL) mailing list, and concerned someone who was accused of assaulting an underage girl connected to the Epstein case. In his message, Stallman stated that “*the most plausible scenario is that she [the underage victim] presented herself to him as entirely willing*” which sparked much criticism on the mailing list and elsewhere. While Stallman did resign from his several roles following many requests to do so, he still considers the situation to be a “series of misunderstandings and mischaracterizations of what I have said”, and doesn’t recognize any wrongdoing on his part. It has to be noted however that Stallman’s statement is actually one of his many documented and questionable acts perceived as sexist, ableist or legitimizing child sexual abuse. See for instance https://geekfeminism.wikia.org/wiki/Richard_Stallman (accessed on October 24th, 2019).

compared with 34 percent of programmers in the US” (Finley, 2017; Github, 2017).

Dunbar-Hester, while praising initiatives advocating for diversity in free and open source software communities, also noted that these initiatives are often restricted to gender and miss the opportunity to tackle diversity more broadly. As she writes, this “serves to perpetuate the marginalized status of poor white women and women of color in technical cultures” (Dunbar-Hester, 2019, p. 93).

6. Conclusion

Most observers today recognize the undeniable ability of the free and open source software model to produce mature and stable technologies, many of which presently form a critical part of Internet Infrastructures, like mail and web servers, or significant parts of widely adopted devices, such as Android phones. Contrary to the so-called “proprietary” software, free and open source software can be characterized as a commons-based peer production model, founded on normative and organizational pillars. The normative pillar refers to the ethics of sharing and its surrounding legal arrangements, more specifically “copyleft” licenses that guarantee the right to share source code rather than limiting it. The organizational pillar refers to the “bazaar” form of making software, centered around the assemblage of a multitude of distributed contributions into a whole by the project leaders.

While free and open source software is today strongly backed by commercial interests, it must be recalled that its principles were first developed for ethical reasons, based on the desire to preserve a culture of sharing and self-help in computer programming, as opposed to restrictions of use imposed by commercial software editors. Although motivations to become involved in the development of free and open source software are now quite diverse, an ethical dimension continues to drive many participants

who often take part in peripheral causes, such as providing access to government data, the protection of privacy on the Internet or the right to share cultural works online. The cultures, philosophies and models of free and open source software development have also inspired similar models of collaborative production or the legal guarantee of the preservation of digital works in the public domain, such as the Creative Commons licenses or the well-known Wikipedia online encyclopedia. However, this model – and especially its ethical component – is in strong tension with commercial appropriation. Free and open source software development has also been criticized for its lack of diversity and its hostility towards women, which can in turn have implications in terms of software quality. In short, while free and open source software has succeeded in being adopted by dominant players in the field and is today used for much of the infrastructure of the Internet, struggles are still ongoing to address the lack of diversity within its contributors and to keep alive the ethical ideas at the origin of the movement.

References

- Aigrain, P. (2005). *Cause commune : l'information entre bien commun et propriété*. Paris: Fayard.
- Auray, N. (2000). Le savoir en réseaux et l'empreinte inventive. *Revue Alice*, 3, 78-97.
<https://www.multitudes.net/le-savoir-en-reseaux-et-l/>
- Balter, B. (2015, March 10). Open source license usage. <https://github.blog/2015-03-09-open-source-license-usage-on-github-com/>
- Bauwens, M., Lacomella, F., Mendoza, N., Burke, J., Pinchen, C., Léonard, A. & Mootoosamy, E. (2012). Synthetic Overview of the Collaborative Economy, P2P Foundation. <http://p2p.coop/files/reports/collaborative-economy-2012.pdf>
- Benkler, Y. (2003). Freedom in the commons : Towards a political economy of information. *Duke Law Journal*, 52, 1245-1276.
- Benkler, Y. (2006). *The Wealth of Networks: How Social Production Transforms Markets And Freedom*. Yale University Press.
- Benson, T. (2005, March 29). Brazil: Free Software's Biggest and Best Friend. *The New York Times*. <https://www.nytimes.com/2005/03/29/technology/brazil-free-softwares-biggest-and-best-friend.html>

- Birkinbine, B. (2014). *Incorporating the Commons: A Political Economic Analysis of Corporate Involvement in Free and Open Source Software* [Doctoral dissertation, University of Oregon]. <https://scholarsbank.uoregon.edu/xmlui/handle/1794/18513>
- Bollier, D. (2011). *Brazil: No Longer the “Free Culture” Nation*. David Bollier news and perspectives on the commons. Retrieved from <http://www.bollier.org/brazil-no-longer-free-culture-nation>
- Boyle, J. (2008). *The public domain: enclosing the commons of the mind*. New Haven: Yale University Press.
- Broca, S. (2013). *Utopie du logiciel libre : du bricolage informatique à la réinvention sociale*. Lyon: Le passager clandestin.
- Byfield, B. (2006, May 20). FSF reaches out to social activists. *Linux.com*.
<https://www.linux.com/news/fsf-reaches-out-social-activists/>
- Cardon, D. (2005). Innovation by Usage. In A. Ambrosi, V. Peugeot, & D. Pimienta (Eds.), *Word Matters. Multicultural perspectives on information societies*. Caen : C&F Editions.
- Castells, M. (2003). *The Internet Galaxy: Reflections on the Internet, Business, and Society* (1st edition). Oxford: Oxford University Press.
- Cohen, N. (2018, September 19). After Years of Abusive E-mails, the Creator of Linux Steps Aside. *The New Yorker*. <https://www.newyorker.com/science/elements/after-years-of-abusive-e-mails-the-creator-of-linux-steps-aside>
- Coleman, G. (2004). Indymedia’s independence: From activist media to free software. *Planetnetwork Journal*, 1(1).

https://mediastudiesfa12.files.wordpress.com/2012/12/coleman_indymedias-independence.pdf

Coleman, G. (2009). CODE IS SPEECH : Legal Tinkering, Expertise, and Protest among Free and Open Source Software Developers. *Cultural Anthropology*, 24(3), 420–454. <https://doi.org/10.1111/j.1548-1360.2009.01036.x>

Coleman, G. (2012). *Coding Freedom: The Ethics and Aesthetics of Hacking*. Princeton: Princeton University Press.

Coleman, G., & Golub, A. (2008). Hacker practice: Moral genres and the cultural articulation of liberalism. *Anthropological Theory*, 8(3), 255-277. <https://doi.org/10.1177/1463499608093814>

Coris, M. (2006). Chronique d'une absorption par la sphère marchande : les Sociétés de Services en Logiciels Libres. *Gérer & Comprendre*, 84, 12–24.

Coris, M. (2009). Free software on the market-side: the failure-story of free software services companies in France. *Economics of Innovation and New Technology*, 18(6), 547–564. <https://doi.org/10.1080/10438590802231556>

Couture, S. (2007). *Logiciel libre, activité technique et engagement politique: la construction du projet GNU en Argentine* [Master dissertation, Université du Québec à Montréal]. <https://archipel.uqam.ca/3301/1/M9669.pdf>

Couture, S. (2013). *Logiciels libres : réduction des coûts et souveraineté numérique*. Institut de recherche et d'informations socio-économiques. <https://iris-recherche.qc.ca/publications/logicielslibres>

- Couture, S. (2015). Le contrôle des communs numériques à des fins commerciales : le cas des logiciels libres. *Éthique Publique*, 17(2). doi:10.4000/ethiquepublique.2275
- Couture, S. (2017). Values and Configuration of Users in the Design of Software Source Code. *International Journal of Communication*, 11, 1112-1132. Retrieved from <http://ijoc.org/index.php/ijoc/article/view/4179>
- Couture, S. (2019). The Ambiguous Boundaries of Computer Source Code and Some of Its Political Consequences. In J. Vertesi & D. Ribes (Eds.), *digitalSTS: A Field Guide for Science & Technology Studies* (pp. 136–156). Princeton (N.J.): Princeton University Press.
- Couture, S., & Toupin, S. (2019). What does the notion of “sovereignty” mean when referring to the digital? *New Media & Society*, 21(10), 2305-2322.
- Dafermos, G., & Söderberg, J. (2009). The hacker movement as a continuation of labour struggle. *Capital & Class*, 33(1), 53-73.
<https://doi.org/10.1177/030981680909700104>
- Demazière, D., Horn, F., & Zune, M. (2007). The Functioning of a Free Software Community : Entanglement of Three Regulation Modes - Control, Autonomous and Distributed. *Science Studies*, 20(2), 34–54.
- Dunbar-Hester, C. (2019). If “Diversity” Is the Answer, What Is the Question? Understanding Diversity Advocacy in Voluntaristic Technology Projects. In J. Vertesi & D. Ribes (Eds.), *digitalSTS: A Field Guide for Science & Technology Studies* (pp. 81–98). Princeton (N.J.): Princeton University Press.

Evangelho, J. (2019, September 17). Richard Stallman Has Resigned As President Of The Free Software Foundation. *Forbes*.

<https://www.forbes.com/sites/jasonevangelho/2019/09/17/richard-stallman-has-resigned-as-president-of-the-free-software-foundation/>

Finley, K. (2017, June 2). Diversity in Open Source Is Even Worse Than in Tech Overall.

Wired. <https://www.wired.com/2017/06/diversity-open-source-even-worse-tech-overall/>

Gates, B. (1976). *Bill Gates – An Open Letter to Hobbyists*. <http://www.adp-gmbh.ch/personal/histoire/lettergates.html>

Ghosh, R. A. (2006). Economic impact of floss on innovation and competitiveness of the EU ICT sector. Maastricht: UNU-MERIT. https://la-rem.com/wp-content/uploads/2013/11/2006-11-20-flossimpact_en.pdf

Ghosh, R. A., Glott, R., Krieger, B., & Robles, G. (2002). *Free/Libre and Open Source Software. : Survey and Study*.

<https://www.math.unipd.it/~bellio/FLOSS%20Final%20Report%20-%20Part%204%20-%20Survey%20of%20Developers.pdf>

Github. (2017). *Open Source Survey*. <https://opensource-survey.org/2017/#about>

Gleizes J. & Papatheódou (2000). La passion du libre entretien avec Richard Stallman (interview by J. Gleizes et A. Papatheódou). *Multitudes* 1(1), 377–391.

Greenberg, A. (2016, June 6). Tor Developer Jacob Appelbaum Resigns Amid Sex Abuse Claims. *Wired*. <https://www.wired.com/2016/06/tor-developer-jacob-appelbaum-resigns-amid-sex-abuse-claims/>

- Haché, A. (2017). Technological Sovereignty. *Mouvements*, 79(3), 38–48.
<http://www.cairn-int.info/revue-mouvements-2014-3-p-38.htm>
- Hess, C., & Ostrom, E. (2007). *Understanding knowledge as a commons: from theory to practice*. Cambridge, Mass.: MIT Press.
- Himanen, P. (2001). *The hacker ethic and the spirit of the information age*. New York: Random House.
- Jullien, N., & Zimmerman, J.-B. (2007). Peut-on envisager une écologie du logiciel libre favorable aux nuls? *Cahier de Recherche, Môle Armoricaïn de Recherche Sur La Société de l'information et Les Usages d'Internet (MARSOIN)*, (9).
<https://bit.ly/31Faadi>
- Juris, J., Caruso, G., Couture, S., & Mosca, L. (2013). The Cultural Politics of Free Software and Technology within the Social Forum Process. In J. Juris & A. Khasnabish (Eds.), *Insurgent Encounters: Transnational Activism, Ethnography, and the Political*. Durham: Duke University Press.
- Karjalainen, M. (2010). *Large-scale migration to an open source office suite: An innovation adoption study in Finland* [Doctoral dissertation, Tampere University].
<http://tampub.uta.fi/handle/10024/66646>
- Lakhani, K., & Wolf, R. G. (2003). Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.443040>
- Lerner, J., & Tirole, J. (2000). The Simple Economics of Open Source. *Journal of Industrial Economics*, 52, 197–234. <http://www.nber.org/papers/w7600>

- Levy, S. (1985). *Hackers heroes of the computer revolution*. New York: Dell.
- Lewis, J. A. (2010). Government open source policies. *Center for Strategic and International Studies*.
http://csisdev.forumone.com/files/publication/100416_Open_Source_Policies.pdf
- Lin, Y. (2005). The Future of Sociology of FLOSS. *First Monday*, (Special Issue #2: Open Source). http://firstmonday.org/issues/special10_10/lin/index.html
- Moglen, E. (1999). Anarchism Triumphant : Free Software and the Death of Copyritht. *First Monday*, 4(8).
- Moulier-Boutang, Y. M. (2007). *Le capitalisme cognitif: La Nouvelle Grande Transformation*. Paris: Editions Amsterdam.
- Nafus, D., Leach, J., & Krieger, B. (2006a). Gender: Integrated Report of Findings. In *Free/Libre and Open Source Software: Policy Support*. <https://bit.ly/2odFVMK>
- Nafus, D., Leach, J., & Krieger, B. (2006b). Gender: Policy Recommendations. In *Free/Libre and Open Source Software: Policy Support*. <https://bit.ly/2pX2G87>
- Oram, A. (2011). Promoting Open Source Software in Government: The Challenges of Motivation and Follow-Through. *Journal of Information Technology & Politics*, 8(3), 240–252. <https://doi.org/10.1080/19331681.2011.592059>
- Pellerin, F. (2013, June 15). Le discours de Fleur Pellerin dans les nouveaux locaux Mozilla à Paris. *Framablog*. <https://framablog.org/2013/06/15/fleur-pellerin-logiciel-libre-mozilla/>

Raymond, E. S. (2000). A Brief History of Hackerdom.

<http://www.catb.org/~esr/writings/cathedral-bazaar/hacker-history/>

Raymond, E. S. (2001). *The Cathedral and the Bazaar*. Sebastopol (Calif.): O'Reilly Media, Inc.

Riemens, P. (2006). Some thoughts on the idea of 'hacker culture'. *Complément Numérique de La Revue Multitude*, 8. <http://www.multitudes.net/some-thoughts-on-the-idea-of/>

Savic, S. & Wuschitz, S. (2018). Feminist Hackerspace as a Place of Infrastructure Production. *Ada*, 13. <https://adanewmedia.org/2018/05/issue13-savic-wuschitz/>

Schweik, C. M. (2007). Free/Open-Source Software as a Framework for Establishing Commons in Science. In C. Hess & E. Ostrom (Eds.), *Understanding knowledge as a commons: from theory to practice* (pp. 277–309). Cambridge, Mass.: MIT Press.

Stallman, R., Lessig, L. & Gay, J.(2002). In *Free Software, Free Society: Selected Essays of Richard M. Stallman*. Boston, Ma. : Gnu Press Free Software Foundation.

Star, S. L. (1999). The Ethnography of Infrastructure. *American Behavioral Scientist*, 43(3), 377–391. <https://doi.org/10.1177/00027649921955326>

Terranova, T. (2000). Free labor: Producing culture for the digital economy. *Social Text*, 18(2), 33–58. <http://muse.jhu.edu/journals/soc/summary/v018/18.2terranova.html>

Toupin, S. (2014). *Feminist Hackerspaces: The Synthesis of Feminist and Hacker Cultures*. *Journal of Peer Production*, 10.

<http://peerproduction.net/editsuite/issues/issue-5-shared-machine-shops/peer->

[reviewed-articles/feminist-hackerspaces-the-synthesis-of-feminist-and-hacker-cultures/](#)

Tuomi, I. (2006). Evolution of the Linux Credits file: Methodological challenges and reference data for Open Source research. *First Monday*, 9(6).

http://firstmonday.org/issues/issue9_6/tuomi/index.html

von Hippel, E. (2005). *Democratizing Innovation*. The MIT Press.

von Hippel, E. (2007). Horizontal innovation networks—by and for users. *Industrial and Corporate Change*, 16(2), 293–315.

<http://icc.oxfordjournals.org/content/16/2/293.short>

Williams, S. (2002). *Free as in Freedom: Richard Stallman's Crusade for Free Software* (1st edition). Sebastopol, Calif. : Farnham: O'Reilly Media.