HAL
open science

A Historical Account of the Value of Free and Open Source Software: From Software Commune to Commercial Commons

Magnus Bergquist, Jan Ljungberg, Bertil Rolandsson

To cite this version:

Magnus Bergquist, Jan Ljungberg, Bertil Rolandsson. A Historical Account of the Value of Free and Open Source Software: From Software Commune to Commercial Commons. 9th Open Source Software (OSS), Oct 2011, Salvador, Brazil. pp.196-207, 10.1007/978-3-642-24418-6_13 . hal-01570762

HAL Id: hal-01570762

https://inria.hal.science/hal-01570762v1

Submitted on 31 Jul 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

A Historical Account of the Value of Free and Open Source Software: From Software Commune to Commercial Commons

Magnus Bergquist, Jan Ljungberg, and Bertil Rolandsson

University of Gothenburg
{magnus.bergquist, jan.ljungberg, bertil.rolandsson}@gu.se

Abstract. Free and open source software has transformed from what has been characterized as a resistance movement against proprietary software to become a commercially viable form of software development, integrated in various forms with proprietary software business. In this paper we explain this development as a dependence on historical formations, shaped by different ways of justifying the use of open source during different periods of time. These formations are described as arrangements of different justificatory logics within a certain time frame or a certain group of actors motivating the use of free and open source software by referring to different potentialities. The justificatory arrangements change over time, and tracing these changes makes it easier to understand how the cultural, economic and social practices of open source movements are currently being absorbed and adopted in a commercial context.

Keywords: Free and Open Source Software, Justification, Historical approach

1 Introduction

Over the last decade, free and open source software (FOSS) have transformed from being an ideologically driven movement, organizing resistance against proprietary software development, to a means for revitalizing the way firms produce software and make business. This change can be described as a development, where open source gradually have been recognized by different actors outside the open source communities and incorporated in corporate software development contexts. While this development has been described by different authors e.g. [4, 5, 8, 9, 10, 13, 24, 38], the conditions and circumstances enabling the transformation of FOSS has rarely been addressed.

The purpose of this paper is to articulate this transformation, by analyzing how the use of FOSS has been justified during different periods of time. These justifications are analyzed by using Boltanski and Thevévenot's [3] framework of "justificatory regimes" that enable actors in various settings to justify different means and initiatives. We claim that these historically formed arrangements condition the

adoption of FOSS software and methods and its cultural, economic and social practices in a commercial context.

2 Logics of Justification

FOSS has during its development been motivated or justified with different arguments ranging from a moral non-utilitarian stance to a pragmatic, utilitarian stance [36]. In this paper justification refers to how actors make use of different logics to embrace ideas of change or novelties. Logics of justification are ways by which actors make the changes legitimate through ongoing processes of valuation.

The analysis focus on how the importance of FOSS is recognized through combinations of logics of justification, reshaped over time. In order to distinguish more than e.g. the importance of a market value, we will then talk about different types of worth [39]. An important start point is the theoretical elaboration on the concept of justification done by Luc Boltanski together with Laurent Thevénot [3], in which different social and moral aspects are considered as important to how change is justified. More precise, they depict six different logics or "justificatory regimes" that enable actors in various settings to justify means and initiatives. Boltanski and Chiapello also used this framework to map the "spirit of capitalism", i.e. the justifications of people's commitment to capitalism in a certain era. Here we apply it to the development of FOSS and changes in the dominant value system. Thus, this is an analysis of "the spirit of open source" or the ideology that justifies people's commitment to FOSS in a certain time period. The six logics of justification suggested by Boltanski and Thevénot [3] are:

- An *inspirational logic* is founded on a principle of grace or artistry serving what is perceived as authentic qualities of life.
- A *domestic logic* can be traced when an established hierarchy made out of personal interdependencies, with a patriarch or guru on top, is justified as natural by referring to a stable social order or tradition. An example could be a conservative family organization, ruled by an authoritarian father or elder.
- In a *popular logic* justification is reached through importance of being renowned, i.e. by being granted credit and esteem in the opinion of others. The achieved worth becomes dependent on identification and fame.
- Within a *civic logic*, justification relies on being representative and on acting in accordance with a collective will. Worth is created through the capacity to mobilize collectives around common interests. In this process, moral claims, and definition of identity become important.
- Justification within the *market logic* depends on individuals and their ability to possess and compete. The worth would then be related to individuals' selling and buying goods and services. This can be perceived as an egoistic practice. However, the right to possess and seize market opportunities is related to a claim that, if done fairly common good will emerge out of market transactions.

- An *industrial logic* justifies actions and initiatives by referring to efficiency and the scale of abilities. Contrary to the market logic, the industrial logic focuses on whether functionality and productivity is organized in a reliable way.

To Boltanski and Thevénot [3] these logics describe how the worth of initiatives can be perceived differently, but also that any justification relies on claims that are based on socioeconomic conditions as well as some sort of moral order. Thus, the ability to make trustworthy references to both a general fairness and social order will be necessary for a justifying logic to emerge.

In our analysis, the various logics that Boltanski and Thevénot identify are shaping the justifying arrangements emerging within a certain time frame or a certain group of actors. The tensions between FOSS practices and proprietary practices, creates uncertainties about the future impact of adopting FOSS software. Thus, different actors struggle to justify the use of FOSS by referring to different potentialities. These justifying arrangements can work as integrative forces, but they are also associated with tensions between e.g. social order and moral claims. The analysis concerns how old perspectives on FOSS software development become active components in new circumstances, justified according to principles that involved actors in different organizational contexts can agree and act upon [3]. Hence, we will follow how new and old discourses overlap and form interpretative arrangements, guiding how means and measures are motivated and common principles articulated [2].

3 Method

In order to understand how perceptions of FOSS software have developed over time, we have traced justifying arrangements that historically have been used to define the value of FOSS. This has been done by going through canonical texts and previous research [3]; i.e. we look at research and publications considered to have had a major impact on the perception of FOSS. Typical for this archeology of knowledge [14] is to compare series of sources over time in order to capture changes in dominant modes of thinking, acting and organizing.

We have been looking at three time periods, were we claim that certain arrangements of justification logic is constituted. Certain events function as approximate starting points of these time periods. The formulation of the free software definition and constitution of the Free Software Foundation is the starting point for the first time period (early 1980s), constituting the first justificatory arrangement. Here analysis is based on texts that evolved around the Free Software Foundation (FSF) and the front figure Richard Stallman (e.g. [15, 17, 36]). The starting point of the second time period is the formulation of the open source definition and constitution of the Open Source Initiative (late 1990s). This is based on texts related to the Open Source Initiative (OSI) and the front figure Eric Raymond and his seminal and much referred texts that were later published as the book *The Cathedral and the Bazaar* [29]. The starting point for the third time period is the emergence of public sector policy documents regarding FOSS, created by policy

making bodies, advocacy groups and governments representing public sector interests (early 2000s). Some of these documents, e.g. national reports and policy documents, constitute the basis for the analysis of the justificatory arrangement that we call public commons e.g. [27, 28, 32, 33].

4 Arrangements of Ideological Justification in the History of FOSS

4.1 First Arrangement: Software Commune

When the free software movement started to mobilize during the eighties it was a reaction against the emerging software industry, and it was organized as an ideologically framed commune. Earlier no software industry or market for software did exist because software was developed directly for specific hardware [6]. Since intellectual property for software was a non-issue, the programmers were used to share solutions, knowledge and the source code itself. They took pride in being skilled programmers, and were eager to help fellow programmers. However, when the market for software took off, the programmers' old practices of sharing were abandoned, and the source code became a private company property to be carefully protected. This provoked some developers to take action in the shape of a politically driven movement. One of the key persons in this process was Richard Stallman (RMS) who still plays an important role in the movement. Stallman's work on the text editor Emacs is a good example of the spirit of the movement. Emacs was given away by Stallman on the condition that other programmers should "give back all extensions they made, so as to help Emacs improve. I called this arrangement the Emacs commune" [Stallman in 25, p. 416]. The emerging copyright protected software development practices faced Stallman with what he describes as a stark moral choice:

> "The easy choice was to join the proprietary software world, signing nondisclosure agreements and promising not to help my fellow hacker." [15, p. 17].

Stallman chose another route and facilitated a number of initiatives that institutionalized the resistance to proprietary software, such as the GNU project, the Free Software definition [17], the Free Software Foundation, and the GNU General Public License (GPL) [15, 36] that was designed to ensure that the rights of the free software definition were preserved (i.e. an inscription of the free software definition in copyright law). The "viral" character of GPL, i.e. that other software that is bundled with a GPL-licensed software must also be released under GPL, created tensions with proprietary software.

Here, justification was based on a *civic logic* based on principles and rules defining free software as a common good. Software code must be made available for anyone to use, alter and redistribute to secure future development of the ideas that the code entails. The proprietary development was a threat against the programmers' freedom: "The fundamental act of friendship among programmers is the sharing of programs;

marketing arrangements now typically used essentially forbid programmers to treat others as friends." [15]. It was also threatening a more general public interest in the freedom of information.

Besides the civic logic, an *inspirational logic* could be identified, emanating from the roots of the free software movement in the hacker culture of the early sixties. This was mainly formed around several MIT research groups who were experimenting with new technologies (e.g. TX-0 computer, MIT AI Lab and the Unix operating system). Levy [25] described this culture as:

> "a new way of life, with a philosophy, an ethic and a dream.[...] hackers that by devoting their technical abilities to computing with a devotion rarely seen outside monasteries they were the vanguard of a daring symbiosis between man and machine." [25, p. 39].

Here hacking and playing with technology were justified as the authentic values of life and the true motivational force for programmers' engagement. The activity of programming itself is often referred to as an art [11, 22], e.g. as Donald Knuth has formulated it "The chief goal of my work as educator and author is to help people learn how to write *beautiful programs*" [22, p. 6]. The word hack and hacking changed over time from "a spirit of harmless, creative fun" to "acquire a sharper, more rebellious edge" [36]. Still, the hacker concept is deeply linked to the ability to solve difficult problems for its own sake, as the definition Stallman gives to it: "Playfully doing something difficult, whether useful or not, that is hacking." [15].

Also a *popular logic* was visible, since the reputation of being a skilled hacker is at the heart of the very concept. To become a hacker is not something that individuals decide by themselves, it is something they earn by getting respect from the community. Public opinion itself establishes the worth of FOSS initiatives and actors, in the sense that popular and famous projects or persons will attract many contributors. There is even a special word in the hacker dictionary for the most admired programmers with an exceptional reputation - demigood: "A hacker with years of experience, a world-wide reputation, and a major role in the development of at least one design, tool, or game used by or known to more than half of the hacker community." (Jargon-file 4.3.1).

Furthermore, the tight community with its' closed clan-like hierarchy of personal interdependencies and patriarchic governance, resembles a *domestic logic*. This is what Raymond in his book "the Cathedral and the Bazaar" criticized as the cathedral-building style of development, even though Raymond himself was a former believer:

> "I believed that the most important software (operating systems and really large tools like Emacs) needed to be built as cathedrals, carefully crafted by individual wizards or small bands of mages working in splendid isolation, with no beta released before its time." [29].

We have labeled the justifying arrangement of the first time period, *commune*, due to its nature of a tight community, kept together by strong common hacker values. As has been shown, this arrangement drew mainly on a technically driven civic *community logic* rooted in the hacker movement, demanding free access to

information and source code while fighting against proprietary commercial interests in software development. In addition, the arrangement also relied upon an *inspirational logic* stressing the importance of authentic grace or technical artistry. This encourages the developers to independently realize personal creativity, and thereby improve their status within the community. This leads also to a *popular logic* stressing the importance of reputation and fame. Also, a domestic logic follows from the nature of the movement as a closed tight community were highly respected developers took on roles representing hierarchical superiority typical for the domestic logic.

4.2 Second Arrangement: The Bazaar

During the mid nineties a new approach to justify FOSS can be identified. Eric Raymond, saw the earlier movement's hostile attitude to commercial software as a big problem:

> "It seemed clear to us in retrospect that the term 'free software' had done our movement tremendous damage over the years. Part of this stemmed from the well-known 'free speech/free-beer' ambiguity. Most of it came from something worse -- the strong association of the term 'free software' with hostility to intellectual property rights, communism, and other ideas hardly likely to endear themselves to an MIS manager." [29].

In order to avoid these connotations, the term open source was coined, indicating that open source is viewed as a means to an end of producing software of high quality. The Open Source definition is similar to the Free Software definition, but it explicitly states that an open source license must not contaminate other software (as the GPL-license), claiming that this would hamper commercial use of open source. A plethora of more permissive licenses [35], were used to make it easier for open source and proprietary software to coexist. The Open Source Initiative (OSI) was founded in 1998 to support the new focus on technology rather than ideology. This more pragmatic nature of the movement, downplayed some of the most ideological parts of the value system, but also contributed to a wider diffusion of free and open source software. The movement grew substantially, and included both large traditional software companies (e.g. IBM, HP) and small companies that were founded on open source business models (e.g. Red Hat, Mandrake).

In this time period the dominating technically driven civic imperative is replaced by clearly visible *market logic*:

> "RMS's best propaganda has always been his hacking. So it is for all of us; to the rest of the world outside our little tribe, the excellence of our software is a far more persuasive argument for openness and freedom than any amount of highfalutin appeal to abstract principles. So the next time RMS, or anybody else, urges you to "talk about freedom", I urge you to reply "Shut up and show them the code." [30]

It is not philosophical or political principles, but the excellence of the software that should convince. The excellence of the software also points to quality ideals that are often found in an *industrial logic*. This also stresses the importance of the code itself and its accessibility, as a key to the arrangement. As an alternative to the domestically oriented cathedral style, where wizards were leading a tight, closed tribe of skilled hackers, Raymond proposed the "Linus Torvalds's style or the bazaar style of development - release early and often, delegate everything you can, be open to the point of promiscuity" [29, p. 30].

However, the *inspirational* and *popular logic* are still visible in the bazaar. Inspirational worth of open source software would depend on spontaneous and passionate initiatives, like Linus Torvalds' initiative to write the Linux system in order to learn how operating systems work [34], and software development close to artistry were still appreciated. The *popular logic* was strengthened in the bazaar where skilled programmers could gain reputation and fame if they succeeded to pass the peer review system [1]. They could gain reputation among an even larger crowd of developers, due to the open character of the bazaar. Also a *domestic logic* could be detected, e.g. in the coordinating model often referred to as "benevolent dictatorship" with Linus Torvalds as the prime example [29], and the informal hierarchies resulting from differences in status and skill within FOSS communities.

We have labeled the justifying arrangement of this time period, the *bazaar*, in accordance with Raymond's metaphor. Here, FOSS and proprietary software will coexist, and anyone is free to choose what is considered the best solution (*market logic*). FOSS is viewed as a better, more efficient method for developing software of good quality (*industrial logic*). Hacker values are still emphasized claiming that free access to code would improve developers' opportunities to do innovative and artistic programming (*inspirational logic*). The spread of the movement make opportunities to get reputation and fame among peers for making good contributions even more attractive (*popular logic*). However, despite its strong market component, open source software was still associated with a *civic logic* where freedom of information became important.

4.3 Third Arrangement: The Public Commons

In the beginning of 2000, a pragmatic version of FOSS started to be appropriated by large user groups outside FOSS communities. Especially governments and public sector organizations found an interesting potential in keeping computing costs down by using FOSS software. The domination of FOSS applications in the horizontal domain of infrastructural software (e.g. operating systems, web servers, and data

bases) was complemented with an increasing use of vertical software as desktop and enterprise systems [13]. With a growing number of FOSS users that were not producers or experts but "general end-users", FOSS moved out of the pure hacker domain.

The growing use of FOSS in this context revitalized the former *civic logic*. Manifested in a number of policy texts, it was reinterpreted by public sector organizations and advocacy groups representing public sector interests (e.g. governments, municipalities, FSF and OSI representatives). The incorporation of FOSS in public sector was seen as an appraisal of values associated with democracy, citizenship and the relationship between citizens and public sector. On the one hand FOSS became attractive to public authorities due to new demands and needs dictated by changes in their own organizations and in society. Economically it was a way to cut costs in the public sector, and get value for taxpayer money. Ideologically FOSS was seen as an expression of the principle of the commons, a way to promote ideas associated with public sector organizations' role in a democracy. It was presented as a radical alternative that could liberate the public sector by getting rid of bureaucratic and expensive non-democratic historical burdens; the public sector would serve the people while standing free from partial interests on the market. On the other hand the FOSS movement took the opportunity to influence public authorities by lobbying activities in order to gain a widespread impact of their goals. FOSS provided the public sector with the ideology and examples it needed to make its point.

A set of policy documents formulated around year 2000 [20, 21, 26, 27, 28, 32, 33, see also 18] give a more detailed view of basic arguments about the use of FOSS in public sector. One argued advantage was cost reasons. FOSS was often made available at a low acquisition cost, and without licensing costs. By adopting FOSS solutions it was also argued that different public actors could develop shareable solutions that would decrease development costs. The cost argument is an efficiency argument, relating to an *industrial logic*, but also to a public governance version of the *civic logic* demanding transparency in how taxpayer money is used. Other arguments aligned to an industrial logic are related to the supposed quality and reliability of FOSS. The same holds for common arguments related to security, transparency and privacy. The free availability of the source code supposedly offers better protection against malware, meaning better protection for the citizens' integrity. These arguments also relates to how an *industrial logic* of efficiency is combined with a public version of the *civic logic*, i.e. how the civic mission best could be implemented in an efficient way.

Other arguments relates to market *logic*. FOSS devotion to promoting open standards and interoperability secures that systems ensure access to government data without possible barriers posed by proprietary software and data formats. This would lead to a situation where lock-in effects of proprietary companies' software could be avoided. By promoting open standards and interoperability in its own systems, public sector contributes actively to well functioning software markets, minimizing monopolies and lock-in effects. Another argument related to a *market logic* is that regional software industry was supposed to prosper as a consequence of public sector interventions in FOSS. Local programmers were to be engaged in flexible adaptation

to specific needs that were not supported by global commercial actors, and create new niche markets to be exploited by local entrepreneurs rather than by global software firms.

Finally, a set of arguments more directly related to a *civic logic* could be found. Arguments related to political reasons, claimed the advantages of national-wide FOSS based IT-infrastructure in developing countries. Here post-colonial arguments were raised, highlighting the possible independence from Western software houses controlling the IT-development: "If South Africa chooses the open route […] South Africa can break dependence on foreign companies, and potentially become a player in the world of software development and software services markets" [32]. Other arguments directly linked to a civic logic stress *freedom* and *democracy* as basic values inherent in FOSS.

The justifying arrangement of the third time period is labeled *public commons*. This is the first justifying arrangement that takes shape outside the movement. This becomes evident while looking at how the *civic logic* is reinterpreted from a public sector perspective; open source is seen as a mean for enhancing democracy and making the public sector free in relation to commercial interest. The civic logic promotes the public sector as a service provider for citizens, which calls for certain moral claims regarding loyalty to the public who elects officials to represent them. This justifies claims on honest and transparent development of software made for the citizens by using FOSS. The public sector is given a mission based on the *market logic* and the *industrial logic*, directing attention towards issues of cost efficiency, reliability and quality. Furthermore, the promotion of open source in public sector is supposed to contribute to a well functioning software market.

5 Emerging Justificatory Logics of Contemporary FOSS

The historically based arrangements presented above show how FOSS has gone from being justified as a community driven software development endeavor with the developer at the center of attention, to become more motivated by external interests. As shown, these are emerging arrangements in which the content of identified justifying logics continuously have changed. The transformation of FOSS into a commercially viable form has been described by different authors as OSS 2.0, progressive open source, corporate code, professional open source etc. [12, 13, 16]. Here, we view the characteristics of all these phenomena as parts of an emerging arrangement, partly overlapping in time with the mentioned public commons arrangement. The arrangements presented also reveal how e.g. initial civic and inspirational logics are reinterpreted over time. We will now discuss how these logics play out in the formation of a new emerging justifying arrangement, and what this may mean for the adoption of open source in a corporate context.

According to Boltanski and Thevénot [3] justification through the *industrial logic* is achieved by making claims on efficiency, expertise and the scale of abilities. Focus also lays upon whether technological innovations and functionality is organized in a scientifically controlled and predictable way. In accordance, many descriptions of

FOSS have always focused on what can be described as an industrial logic [12, 13, 16], and today is further emphasized in descriptions of how e.g. the voluntary nature of FOSS is substituted by strategic planning, bulletin board like product support becomes professional, the open access to source code is challenged by giving controlled access only to specific business partners, or only internally behind corporate fire walls [12].

The basic idea within market logic is that justification is based on individuals' ability to possess and compete [3]. The worth of such a justification is created when as many as possible are able to sell and buy goods and services. One of the main arguments for FOSS in public commons was to maximize the positive effects of taxpayers' money by making them operate on a more open and transparent market. Hereby the market could be used for reaching a higher cause and thus legitimize public sector civic claims. This way of reasoning, supporting the civic emphasis on honest markets, is also found in firms built around FOSS today. However, these firms also struggle to find ways of combining FOSS with proprietary code [19, 31].

The *civic justification logic* has undergone an interesting development. In the commune and the bazaar, civic justification was the nexus of the FOSS movement's ideological agenda. The aim was to strengthen democracy and free access to information and source code by the help of software communities supporting a common right to independently control software. In the public commons arrangement FOSS was then justified as a mean helping the public sector to become independent from private companies' proprietary standards and lock-in strategies. This also paved the way for a customer and user perspective, supporting a market logic that emphasizes common good rather than proprietary strategies. Open source software becomes a way to improve honest, flexible and efficient relations with customers and end-users. Now, similar claims on challenging the idea of possessing software is found in contemporary pure play firms built around FOSS business models [31], where competition with free and open standards is said to be a more honest approach to customers and users.

Boltanski and Thevénot [3] describe *inspirational logic* as a type of justification that refers to principles such as grace and artistry serving authentic qualities of life. These principles lies at the heart of the initial hacker culture and were prominent both in the commune and bazaar arrangements, but could not be identified in the public commons arrangement. The inspirational logic traced in firms today seems to be reinterpreted from a business perspective, by being less associated with contributions to a higher cause and more associated with being engaged in work. Movement driven inspiration is replaced by professional inspiration associated with a hobby or a scientific quiz triggering lust for work by making professional developers free to access and manipulate the code [13, 31]. In addition, the inspirational and industrial logic do then also support each other. This type of inspirational logic resembles FOSS research on intrinsic motivation, stressing that open source may connect the professional world of software development with the exercise of a hobby [4].

According to Boltanski and Thevénot [3] justification through the *popular logic* is reached through reputation; i.e. being granted credit and esteem in the opinion of others is a goal in itself. The popular logic identified in the commune and in the

bazaar arrangement, related to the reputation that was gained when programmers succeeded to get their contributions of code accepted by peer-reviewers and introduced into the code base [4, 1]. Highest reputation was attached to the visionaries and ideological leaders, gaining reputation through developing widely used FOSS programs. Companies that struggle to attract the best FOSS-programmers, indicates that this logic is re-articulated in firms today. The majority of contributions appear to be rejected, and accepted contributions are still manifestations of programming skills and status. However, focus rather lay on the status of the project than the individual programmer; i.e. the success of attracting contributors to a company owned FOSS project is main issue [31].

In the *domestic logic* Boltanski and Thevénot [3] describe how justification is reached through the stability of conventions or traditions, revolving around a family like organization and its ruler. In the commune, with its tight tribe of developers and highly reputed ideological leaders, a domestic logic could be sensed. The bazaar arrangement then challenged this closed commune, and in the public commons arrangement the domestic logic was not visible at all. The domestic logic is hard to trace in contemporary FOSS in corporate settings. However, it is potentially inherent in many FOSS practices, where developers are part of an informal hierarchy due to skill and reputation, and were the most respected developers is dedicated to roles that clearly points out a hierarchical superiority. Also, FOSS still appears as heavily male dominated, indicating a stable gender structure that could be investigated further in terms of a domestic logic [7, 23].

6 Conclusions

By focusing on the value accredited to FOSS by different groups, and how the justifications of FOSS have been formed in different time periods, we may move beyond the established distinction between the initial movement driven approach and the current business driven OSS 2.0. It becomes possible to describe how different justifying logics are re-articulated in the intersection of FOSS movement and corporations. While the industrial and the market logics emerge as major justificatory means in contemporary commercial FOSS, it is important to notice that core driving forces as the inspirational logic and the civic logic still could be considered as important parts of the FOSS bandwagon. Even in a commercial context marked by industrial and markets logics, these logics still makes it possible to justify the use of FOSS by referring to potentialities that could inspire and empower programmers developing open source, as well as to contribute to the society as a whole.

References

1. Bergquist, M., Ljungberg, J.: The Power of Gifts: Organizing Social Relationships in Open Source Communities. Information Systems Journal, 11, 305-320 (2001)
2. Boltanski, L., Chiapello, E.: The New Spirit of Capitalism. London. New York: Verso (2005)

3. Boltanski, L., Thévenot, L.: On Justification: Economies of Worth. Princeton: Princeton University Press (2006)
4. Bonaccorsi, A., Rossi, C.: Comparing Motivations of Individual Programmers and Firms to Take Part in the Open Source Movement: From Community to Business. Knowledge, Technology and Policy 18 (4): 40-64 (2006)
5. Bonaccorsi, A., Rossi, C.: Why Open Source Software can Succeed. Research Policy 32 (7), 1243-1258 (2003)
6. Campbell-Kelly, M .: From Airline Reservations to Sonic the Hedgehog: A History of the Software Industry. Cambridge, MA, USA: MIT Press (2003)
7. Cuckier, W.: Constructing the IT Skills Shortage in Canada: The Implications of Institutional Discouse and Practices for the Participation of Women. SIGMIS Conference '03, April 10-12 Philadelphia, Pennsylvania, Copyright 2003 ACM (2003)
8. Dahlander, L., Magnusson, M.G.: Relationships between open source software companies and communities: Observations from Nordic firms. Research Policy 34 (4) (2005)
9. Dahlander, L., Magnusson, M.G.: How do Firms Make Use of Open Source Communities? Long Range Planning 41 (6) (2008)
10. Demil, B., Lecocq, X.: Neither Market, nor Hierarchy or Network: The Emergence of Bazaar Governance. Organization Studies 27 (10): 1447-1466 (2006)
11. Dijkstra, E. W.: EWD316: A Short Introduction to the Art of Programming. T. H. Eindhoven, The Netherlands, Aug. (1971)
12. Dinkelacker, J., Garg, P.K., Miller, R., Nelson, D.: Progressive Open Source. In Proceedings of ICSE'02, May 19-25, 2002, Orlando (2002)
13. Fitzgerald, B.: The Transformation of Open Source Software. MIS Quarterly 30 (3) 587-598 (2006)
14. Focault, M.: The Archaeology of Knowledge. London: Routledge Classics 2002.
15. Gay, J., editor (2002) Free Software, Free Society: Selected Essays of Richard M. Stallman. Boston, Massachusetts: GNU Press (1969)
16. Gurbani, V.K. , Garvert, A. , Herbsleb J.D.: A case study of open source tools and practices in a commercial setting, In Proceedings of the 3rd IFIP Working Group 2.13 International Conference on Open Source Software (OSS2007, June 11th–14th, Limerick, Ireland, IFIP Advances in Information and Communication Technology, vol. 234/2007, Springer (2007)
17. GNU's Bulletin 1 (1): 8. Available at http://www.gnu.org/bulletins/bull1, accessed 5 March 2010.
18. Hahn. R.W,. ed.: Government Policy toward Open Source Software. AEI-Brookings Joint Center for Regulatory Studies: Washington DC (2002)
19. Höst, M., Oruc˘evic´-Alagic´, A.: A systematic review of research on open source software in commercial software product development, Inform. Softw. Technol. (2011), doi:10.1016/j.infsof.2010.12.009
20. IDA Study: Study into the use of Open Source Software in the Public Sector Part 1. Available Online: http://europa.eu.int/ISPO/ida/export/ files/en/840.pdf ; Part 3 Available online: http://europa.eu.int/ISPO/ida/ export/files/en/835.pdf , (2001)
21. International Institute of Infonomics “Free/Libre and Open Source Software: Survey and Study” Available Online: http://www.infonomics.nl/FLOSS/report/ [040401] (2002)
22. Knuth, D.: Computer Programming as an Art. Communications of the ACM, December (1974)
23. Lakhani, K. R., Wolf, R.G.: Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. In: Feller, J., Fitzgerald, B., Hissam, S. and Lakhani, K. (eds) Perspectives on Free and Open Source Software. MIT Press, Cambridge, Mass (2005)

24. Lerner, J. and Tirole, J.; Some Simple Economics of Open Source. Journal of Industrial Economics 50, 2:197-234 (2002)
25. Levy, S.: Hackers: Heroes of the Computer Revolution. New York: Anchor Press/Doubleday (1984)
26. LinuxToday "LinuxPR: Munich Goes with Open Source Software". 28 May, (2003) Available online: http://linuxtoday.com/infrastructure/2003052802126NWDTPB.
27. MIMOS Berhad; Worldwide Open Source Policy: National Summaries. Available online: http://community.asiaosc.org/~iwsmith/policy/ [040401]. (2003)
28. MITRE Corporation "Use of Free and Open-Source Software (FOSS) in the U.S. Department of Defense" Available online: http://www.egovos.org/rawmedia_repository/588347ad_c97c_48b9_a63d_821cb0e8422d?/document.pdf (2002)
29. Raymond, E.S.: The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary. Sebastopol, CA: O'Reilly and Associates (1999)
30. Raymond, E.S.: Shut Up and Show them the Code. Linux Today. Jun 28, (1999b)
31. Rolandsson, B., Bergquist, M. Ljungberg, J.: Open source in the firm: Opening up professional practices of software development. Research Policy, 40 (3), 576–587 (2011)
32. South African National Advisory Council on Innovation (2002) "Open Software & Open Standards in South Africa: A Critical Issue for Addressing the Digital Divide" Available online: http://www.naci.org.za/pdfs/opensource.pdf [040401].
33. Statskontoret: Free and Open Source Software – a feasibility study, Appendix 1: Extensive survey. Available Online: http://www.statskontoret.se/pdf/200308eng.pdf [040401] (2003)
34. Linus, T., Diamond, D.: Just For Fun: The Story of an Accidental Revolutionary. New York, New York, United States: HarperCollins (2001).
35. Välimäki, M.: The Rise of Open Source Licensing: A Challenge to the Use of Intellectual Property in the Software Industry. Helsinki, Turre Publishing. versity Press (2005)
36. Williams, S.: Free as in Freedom: Richard Stallman's Crusade for Free Software. O'Reilly Media (2002)
37. Zittrain, J.: Normative Principles for Evaluating Free and Proprietary Software. University of Chicago Law Review, 71 (1). (2004)
38. Ågerfalk, P.J., Fitzgerald, B.:. Outsorcing to an unknown workforce: exploring opensourcing as a global sourcing strategy. MIS Quarterly (32: 2), pp. 385-409 (2008)
39. Stark, D.: The Sense of Dissonance. Accounts of Worth in Economic Life. Princeton: Princeton University Press (2009)