The Meaning of Sharing in Free Software and Beyond

Tomáš Karger

Research Centre of Faculty of Humanities, Tomas Bata University, Zlín, Czech Republic

Nám. T. G. Masaryka 1279, 760 01 Zlín, Czech Republic

E-mail: karger@utb.cz

URL: https://www.researchgate.net/profile/Tomas_Karger

Tomáš Karger is a research fellow at Tomáš Baťa University in Zlín. He is a doctoral graduate of sociology at Palacký University in Olomouc with a dissertation thesis based on ethnographic research of knowledge networks in free and open source software development. In his work, he addresses the relationship between digital technologies and knowledge dynamics. The topic of knowledge production and transmission represents a general research interest which navigates his work into areas such as technology development or collective memory. He gained experience with qualitative research strategies while participating in several research projects funded by the Internal Grant Agency of Palacký University or by the Czech Science Foundation.

The Meaning of Sharing in Free Software and Beyond

Abstract: This study brings together findings about two contexts of sharing in order to explore the meaning of the word in the digital environment. First, this study is based on ethnographic research of free software projects and uses the resulting thick description to determine the meaning of sharing in this context. Second, the current literature on sharing usually takes user generated content platforms as its empirical reference, resulting in identifying a distinct meaning of sharing in this context. By combining the two sets of findings into a single narrative, this study makes three points: (1) the academic discourse on free software conceptualizing it as a form of gift-giving antithetical to the ways of capitalist production needs to be revised; (2) the use of sharing in the context of user generated content platforms relies heavily on references to the culture of free software; (3) although representatives from both contexts claim to be taking part in the same sharing practices, there are substantial differences in the type of information being shared, the explicitness of the sharing mechanisms and the organizational context of monetization of the shared objects.

Keywords: sharing, free software, open source software, user generated content, facebook

1. Introduction

In the online environment, sharing has become a buzzword that is associated with a wide variety of practices, some of them emerging spontaneously and spreading through social movements, some designed purposefully into the interfaces of commercial platforms. The word has become so ubiquitous that it’s meaning is now highly ambiguous, drawing together a wide variety of contexts and connotations. The therapeutic narrative associated with the term imbues it with positive connotations drawing upon the experience of caring and well-being intrinsic to intimate relationships (John, 2013, pp. 124-125). In the technological field, it initially designated pooling of scarce computing resources in an institutionalized context (Kennedy, 2016, p. 4). In the space of online interaction, its original archetype embodied in the intimate relationships within a family

(Belk, 2010, p. 717) has shifted to include communities of people who may have never met face-to-face but have a common cause and sharing the results of their work should help them achieve it. Today, various initiatives and organizations use the term to construct a favorable self-description even though their underlying logic may be based on self-interest and maximization of utility (Banning, 2016, p. 5). There seems to be a parallel with the history of the concept of gift (Mauss, 2002) as it was also originally intended to describe an alternative form of exchange, but ended up showing just how utilitarian gift-giving may often be (Belk, 2010, p. 716).

In this text, I do not aim to search for instances of pure and selfless sharing (Sützl, 2014), nor do I aim to criticize the existing forms of online sharing for not being selfless enough. Instead, I acknowledge the interests various parties have in the practice of sharing. This allows me to see online sharing not as a single and homogenous practice, but as a variety of practices which provide differing contexts and differing meanings to the word. To answer the call made by Jenny Kennedy (2016, p. 11) to explore the elements that constitute sharing in its various forms, I attempt to describe a connection between two prominent contexts using the term – the context of free software and the context of user generated content platforms. To demonstrate the context-dependency of the meaning of sharing, I compare the practices and design decision from those two contexts against a set of distinctions drawn from the existing literature.

The overall argument I am attempting to make consists of several parts. (1) I show that free software practices do not have to be seen only as instances of gift-giving but that they can also be conceptualized as instances of sharing. (2) I describe the relationship between free software projects and commercial entities to make clear how in this context the source code (as an object of sharing) is monetized. (3) I proceed to make explicit the meaning sharing has in the context of free software by pointing to rationalizations and design decisions which are aimed to avoid possible sources of friction perceived as unnecessary. (4) I describe the self-descriptions of Facebook as drawing heavily on the cultural repertoire of the free software movement, claiming its sharing practices to be essentially the same. When the findings from previous points are compared to what is known about the sharing practices of Facebook, it is clear that they differ at least in three regards: the type of information that is shared, the explicitness of the sharing mechanisms, and the organizational arrangements of monetization.

1.1 Free Software

Having roots in the sharing practices that emerged in the 1970s around the UNIX operating system, free software is often seen as inconsistent with several aspects of market capitalism. In particular, authors point to differing work ethics, differing values regarding private property and to existing subversive practices. According to Pekka Himanen (Himanen, Castells, & Torvalds, 2001, p. 6), the 'hacker' work ethics contrasts sharply with the duty oriented ethics of capitalism by placing emphasis on intrinsic motivation (Shah, 2006, p. 1010; Osterloh & Rota, 2004, pp. 291-292; Freeman, 2007, p. 73) and passion for exploration. Furthermore, Gabriella Coleman (2009; 2013, p. 147) identified the central value of the free software movement, which emphasizes that software source code should be treated as a result of free expression, not as a private property. Finally, Johan Söderberg (2008, p. 44; Dafermos & Söderberg, 2009) claims that the free software movement is a continuation of labor struggle revolting against commodification of labor. Software developers can engage in either building free alternatives to proprietary software, or even attempt to exploit the functions of existing software to their advantage. Both types of what is often called 'hacking' can be seen as direct venues of anti-capitalist practice.

The activities of the free software movement focus solely on the first type of hacking – building functional alternatives to proprietary software and licensing them so that their free distribution is granted. It is for the reason that free software developers are seen as giving away their product for free that their behavior is described in the academic discourse as an instance of gift-giving (Benkler, 2006, p. 109; Bergquist & Ljungberg, 2001, p. 308; Chełkowski, Gloor, & Jemielniak, 2016, p. 16). By conceptualizing free software development as a gift-based symbolic economy, the authors recognized characteristics such as implied reciprocity and status attainment through gift-giving. However, the authors also note that the form of gift-giving they observe in free software differs from the practices the concept originally referred to. Magnus Bergquist and Jan Ljungberg (2001, p. 310) noted already in 2001 that the gifts circulating in free software are specific in that the transaction costs of giving them are reduced to a bare minimum. This premise was later systematically explored by Yochai Benkler (2002, 2006) who saw the minimal transaction costs stemming from digital infrastructure and informal modes of organization as a defining characteristic of peer-production in general.

As a result, gift-giving in free software can be seen as routinized or even automated. There is a wide ranging infrastructure surrounded by specific practices (Karger, 2016) to facilitate gift-giving and reduce transaction costs. However, according to Russell Belk (2007, 2010), routinization is precisely the aspect differentiating gift-giving from sharing. Gift-giving requires some kind of ritualized ceremony during which gifts are presented. On the other hand, sharing is largely unnoticed because of its routinized nature (Belk, 2010, p. 717). This author also notes that sharing on the internet often involves an aspect of anonymity between the givers and the recipients of gifts (Belk, 2007, p. 132). Given the "long tail" distribution of free software contributors (Chełkowski, Gloor, & Jemielniak, 2016, p. 7; Kuk, 2006), this aspect further supports the conceptualization of free software practices as sharing rather than gift-giving.

1.2 User Generated Content

In the recent literature on sharing the empirical reference point is largely represented by social media and social networking sites which focus on end-users and on the data they provide (Van Dijck, 2013; John, 2012, 2013; John & Sützl, 2016; Sundararajan, 2016). In other words, the research on sharing focuses on empirical instances that fall under the rubric of user generated content (UGC) (van Dijck, 2009, p. 41). According to Nicolas John, the term has undergone a substantial transformation during the brief history of its use in this area. The objects of sharing became less and less concrete until they vanished completely, broadening indefinitely the set of objects to be shared (John, 2012, p. 8).

Facebook is a well-studied case of a business based on sharing. In her thorough analysis of the networking site, José van Dijck (2013) points out an ambiguity which is strategic for the platform owners. This ambiguity allows for conflation of two meanings: a) connectedness which designates information sharing among users; and b) connectivity which designates sharing of user data with third parties (van Dijck, 2013, p. 46).[1] Such a duality of meaning also results in a duality of practices related to information sharing. While the function and consequences of connectedness (interaction among users within the platform) are visible directly in the interface of the website, the operation of connectivity (transmitting user data outside of the platform)

[1] This conflation is also explicitly acknowledged by Arun Sundararajan by pointing out that what is in the context of sharing economy labeled as sharing is otherwise usually seen as a commercial exchange (Sundararajan, 2016).

remains invisible in the user interface. As a result, this conflation of sharing as a social interaction among users with sharing as an economic exchange among formal parties has been a common point of criticism with regard to Facebook (Bodle, 2011, p. 335; John, 2012, p. 12; van Dijck, 2013, p. 64).

Van Dijck illustrates the conflation on the mechanisms behind the like button, which, although being operated by Facebook, is present on websites external to it. While the visible part provides an interface for bringing objects to user's Facebook profiles and perform the user-to-user interaction of sharing, its underlying mechanisms allow for automated gathering of data about presence and identity of users who did not use it or who do not even have Facebook account (van Dijck, 2013, p. 49). In other words, Facebook creates a convenient interface which spares users of opening their account and copy-pasting URL addresses each time they want to share something with their Facebook friends. And with the same interface element, Facebook simultaneously gathers data in a way that sidesteps the boundary of the platform, is not visible to the users and does not require their conscious decision. In this way, Facebook avoids a possible source of friction in the practice of sharing, but at the same time does away with an important distinction without this being obvious to the user.

In its implications the distinction between connectedness and connectivity seems to be analogical to other distinctions that can be found in the recent literature on sharing. Andreas Wittel (2011, p. 6) distinguishes between sharing as distribution of data and sharing as a personal interaction. Nicolas John (2012, p. 12) distinguishes between sharing as an informal interaction and sharing as a commercial exchange. Russell Belk (2010, p. 725) distinguishes between sharing in and sharing out – a distinction based on identification of the boundary of 'extended self' (family, community, etc.) and whether the sharing practice in question surpasses its boundary. A systematic account of the distinctions was provided by Jenny Kennedy (2016) who distinguishes between sharing as economy, sharing as scaled distribution, and sharing as social intensity while pointing out that these clear-cut meanings are conflated within the networked culture (Kennedy, 2016, p. 5). However, how these distinctions are conflated in particular contexts is an empirical question that I will attempt to tackle in my analysis.

2. Method

This text draws upon ethnographic research (Brewer, 2000; Neyland, 2008) I conducted in a free software project. My fieldwork consisted mainly of participant observation which I carried out in 2014 when (for a period of six months) I assumed the role of software documentation writer in a project aimed at developing a video editing application called Pitivi. Subsequently, I kept monitoring the project's communication channels for another six months until the end of 2014. During my fieldwork, I assembled data from various sources: conversations with or among other project contributors, my own experience with participating on the work in the project, and relevant documents. The material collected in 2014 was supplemented by another wave of data gathering which took place in late 2016/early 2017, that focused on gathering documents in order to investigate further the working premises that resulted from analysis of the original data. The gathered data were analysed and interpreted in order to produce a thick description (Geertz, 1973, pp. 20–21) that weaves together empirical data and existing theoretical knowledge into a single narrative.

The networked nature of the field warranted the use of multi-sited ethnography (Marcus, 1995, p. 669). This research strategy led me to investigate not only the actual practices involved in use of the project's infrastructural elements but also their origins and the decisions behind their actual design. In this way, I took part in the process of constructing the space that represented the investigated ethnographic field (Cook, Laidlaw, & Mair, 2009, p. 63; Falzon, 2012, p. 4; Rybas & Gajjala, 2007, p. 12; Teli, Pisanu, & Hakken, 2007, p. 38). In the second wave of data gathering, I applied the same principle to an object of a more symbolic nature – the concept of sharing, which is implicitly or explicitly present in the infrastructure common to free software projects.

3. Results

3.1 Sharing in Free Software

Free software practices are conceptualized as sharing in the foundational documents of the movement itself. This applies to both of its branches – free software just as well as open source software. For example, the word ‘sharing’ is present in the preamble to the influential GNU General Public License since its first version published in 1989.[2] Overall, the license is designed to allow and encourage distribution, re-use and modification of software source code (while making sure that any software that re-uses the code has the same licensing conditions). In the more general preamble, sharing stands for the first two meanings specified further in the license – distribution and re-use. The situation is similar with regard to the Open Source Initiative which maintains a document called the Open Source Definition. Originally published in 1999, the document[3] provides criteria for evaluation of licenses concerning their consistency with the principles of open source software. Although the word ‘sharing’ is nowhere to be found in the

[2] GNU General Public License, version 1. https://www.gnu.org/licenses/old-licenses/gpl-1.0.html. Accessed: 2017-02-06.

[3] The Open Source Definition. http://www.oreilly.com/openbook/opensources/book/perens.html. Accessed: 2017-02-06.

definition itself, OSI uses it to summarize the definition on a more general level.[4] The three activities of use, modification and sharing are used to characterize open source licensing throughout the web of the Open Source Initiative[5] and so it seems that here sharing designates mainly distribution of software source code.

That sharing is a norm intrinsic to free software is also evident from some of the conversations I witnessed in chat channels during my fieldwork. On one instance, software developers made this clear when assessing another software project that claimed to be open source:

> [21:48] Ted: looking at the [video editor project] blog post... it baffles me that the fact Phillip has not released any source code since 2011 continues to go unchallenged
> [22:03] Jim: Ted: has he released binaries?
> [22:04] Ted: nope, haven't seen that either
> [22:05] Ted: it's been "trust us, we're working on it, it's all on two secure private servers! it will be open source in the end!"
> ...
> [22:06] Jim: well, heh
> [22:06] Jim: maybe they're honest
> [22:06] Jim: but it doesn't help getting contributions if they don't have some public repo[sitory] somewhere
> [22:11] Ted: that's the most puzzling part for me too
> [22:11] Ted: I don't get what they gain by keeping it all behind closed doors
> [22:12] Ted: except having absolute control and avoiding public scrutiny for code quality or licensing or whatever
> [23:01] Jim: it could just be a feeling that things are too 'sloshy' and the design isn't done
> [23:02] Jim: so he/they don't want contribution until things are up to a certain level of

[4] Licenses & Standards. http://opensource.org/licenses. Accessed: 2017-02-06.

[5] Open Source Initiative. http://opensource.org. Accessed:2017-02-06.

stability
[23:02] Jim: then again, the history is littered with vaporware editors ;)

In the conversation, developers Ted and Jim[6] consider what can be the reason for an open source project not to publish source code for several years. They agree that keeping the source code private has not only practical drawbacks, but that it raises doubts regarding trustworthiness of the project and its members. The strength of the norm to publish (*share*) the source code is most clearly articulated by Ted who finds the fact that the practice has not yet been publicly challenged 'baffling.'[7] In the free software context, sharing of source code is encouraged on the grounds of both ethics (emphasized by the FSF) and technical efficiency (emphasized by the OSI). This provides a range of motivations for free software developers to adopt and expect from each other. However, my field experience indicates that such differences are not subject to negotiation in day-to-day operation of the projects, unless explicitly challenged or inquired upon by an actor. Overall, it seems that most of the time, strategic ambiguity is facilitating cooperation among developers with differing worldviews.[8]

[6] The chat nicknames have been altered.

[7] It needs to be noted that in this case, the assessed project went through a successful crowdfunding campaign roughly a year before this conversation took place. In this light, the assessment could be interpreted as criticism of a project that does not fulfil the obligations of its own campaign. However, the fact that Ted refers to a time period before the campaign and that the project frames itself continuously as open source point to a more general problem.

[8] The fact that developers with differing values participate on common free software practices was noted already by Kelty (2008, p. 14), while the role of ambiguity as a facilitator of cooperation was described in other contexts of sharing such as Wikipedia or Quantified Self (Barta & Neff, 2016; Matei & Dobrescu, 2011).

When we compare thenorm of sharing with the business logic which sees the source code as a private property resulting from the production process, we can see why free software has been considered inconsistent with the traditional forms of capitalist production. In the preceding sections, I suggested that this tendency is to be revised as the movement cannot be considered to be wholly anti-capitalist.The vantage point for this claim is that the movement bifurcated into two branches: free software which considers abiding to the principles outlined above to be a moral obligation, and open source software which considers it to be a practical advantage enhancing cooperation in problem solving. Even such brief characterization indicates that the second branch would be more accommodating when it comes to applying free software principles in the business environment. In fact, the split (which took place in 1998 by establishing the Open Source Initiative next to the already existing Free Software Foundation) was motivated precisely by the effort to do away with the heavy moral load associated with free software and offer it as a business friendly software development method.[9]

In the project I was studying this connection was indirect, but was indicated when I traced some of the infrastructural elements outside of it. For its user interface, Pitivi used a framework that belonged to a much larger overarching project – the GNOME Desktop Environment. Although GNOME itself is formally an independent project backed by the non-profit GNOME Foundation, its source code repositories are hosted by Red Hat,[10] a company that uses the GNOME desktop environment in its product, an operating system called Red Hat Enterprise Linux. It is also no secret that Red Hat employs some of the software developers working on the

[9] History of the OSI. http://opensource.org/history Accessed: 2017-01-21.

[10] GNOME git repository. http://git.gnome.org/. Accessed: 2017-02-07.

desktop environment's framework that Pitivi uses. Their individual contributions can even be pinpointed in the publicly available source code repositories. In this way, some of the contributors to the free software project hold a status of respected community members while at the same time being employed to work on free software.

This relationship is reflected also in the organizational structure of the GNOME Foundation as Red Hat has its firm place both in the Board of Directors and in the Advisory Board. However close this relationship may seem, Red Hat does not control the GNOME project. First, the Board of Directors is appointed according to the results of elections in which each member of the GNOME Foundation (the last published report states the foundation has 255 members)[11] has a vote. Furthermore, the GNOME Foundation's charter makes it clear that regardless of election results, 'no single organization or company will be allowed to control more than 40% of the board seats.'[12] Second, the Advisory Board, although it represents a means of exerting influence over the GNOME project, has no formal decision-making ability.[13] This body is intended for representatives of other organizations with a membership fee intended for supporting the foundation.

Taking into account this organizational structure, I gathered from reports and other documents that the relationship between Red Hat and GNOME is articulated in that two of the five current members of Board of Directors are Red Hat employees and that Red Hat is one of the

[11] GNOME Foundation Fiscal Year 2015 Annual Report. http://www.gnome.org/foundation/reports/. Accessed: 2017-02-07.

[12] GNOME Foundation Charter. http://wiki.gnome.org/FoundationBoard/Resources/Charter. Accessed: 2017-02-07.

[13] GNOME Foundation Charter. http://wiki.gnome.org/FoundationBoard/Resources/Charter. Accessed: 2017-02-07.

ten organizations holding membership in the Advisory Board. A similar relationship, although less spelled out formally, can be found with regard to the GStreamer project (whose framework Pitivi uses for its video-editing functionality) and companies such as Collabora and Fluendo. While free software embodies principles that seem to be inconsistent with capitalist ways of production, some of its parts became integrated into the system through indirect associations with private businesses.

When observing the extensive free software infrastructure, it becomes clear that it has one general purpose, that is, to facilitate modification and transferring of digital text – the software source code. A quick inspection of widely used tools in free software development will validate this claim. Text editors such as Vim or Emacs allow for reading and modifying digital text, source code management systems such as Git allow for tracking, comparing and merging changes made to source code, source code repositories allow for publication and distribution of source code, compilers and debuggers translate human-readable source code into machine-readable binary files and (to a more limited extent) vice versa. Furthermore, free software licenses such as the GNU General Public License are designed to facilitate a free flow of source code from one developer to another, from one project to another. This purpose is made explicit for example in the licensing recommendations of the GStreamer project which is tightly associated with the Pitivi project:

> Our suggestion among these choices is to use the LGPL license, as it is what resembles the GPL most and it makes it a good licensing fit with the major GNU/Linux desktop projects like GNOME and KDE. It also allows you to share code more openly with projects that have compatible licenses. As you might deduce, pure GPL licensed code without the above-mentioned clause is not re-usable in your application under a GPL plus exception clause unless you get the author of the pure GPL code to allow a relicensing to GPL plus exception

> clause. By choosing the LGPL, there is no need for an exception clause and thus code can be shared freely between your application and other LGPL using projects.[14]

The GStreamer project focuses on developing free software multimedia framework, but due to some of the widespread multimedia technologies being patented or proprietary, it does not use the GNU GPL and instead uses the permissive GNU Lesser General Public License (LGPL), which, unlike the original GPL license, does not require associated or derivative code to be licensed under the same conditions.[15] Stating technical compatibility as the main reason for using the LGPL license, the GStreamer developers suggest to others who would develop plug-ins for their framework to license their work under LGPL as well. The reason for this is that a standard license such as the LGPL avoids the need to negotiate exception clauses among developers of various software projects. As a result, the source code can be shared more ‘openly’ and ‘freely’ meaning that there has to be no interaction among the projects besides automated data transfer of the source code itself. In other words, negotiation represents a point which could introduce friction to sharing source code and therefore the preferred way to share it should avoid it. The Pitivi project, being a graphical interface for some of the GStreamer functionality, also follows the licensing suggestions of the GStreamer developers.

[14] Licensing your applications and plugins for use with GStreamer. GStreamer documentation page. http://gstreamer.freedesktop.org/documentation/licensing.html. Accessed: 2014-07-23.

[15] The absence of the „viral“ aspect in popular non-copyleft licenses such as MIT, BSD or the LGPL is seen by some as acceptable (notably the OSI’s Open Source Definition) or even abolishing an unnecessary restriction on code distribution, while others (including the FSF) encourage the use of copyleft as a means of leveraging reciprocity and avoiding the “tragedy of the commons” (Chopra & Dexter, 2007, p. 33).

An analogical case could be observed when Linus Torvalds – the originator of the Linux kernel – gave a public talk (referenced at the Pitivi wiki) about his design decisions with regard to the Git source code management system, now the de-facto standard tool for source code management in free software projects.[16] Torvalds explicitly states, that his motivation behind designing the tool was to avoid what he calls "politics" – the often lengthy negotiation of write access to central source code repositories. Eventually, Torvalds came up with a design that implemented a decentralized model where developers have their own repositories among which the source code is periodically transferred. In this sense, Git is designed intentionally to avoid this kind of negotiation – which is also seen as a source of friction – while sharing software source code.

Such arrangements can be seen as similar to the Facebook's like button in that their aim is to avoid perceived sources of friction, but there are also substantial differences. From my observation I know that however routine sharing in free software projects may be, it is still done through conscious decisions of adopting a specific licensing for one's work and publishing the work through online source code repositories. When the infrastructure is in place, publishing is a matter of typing a command into terminal (for example, *git push*). The effects and implications of this command are well documented in the Git user manual[17] and the command line interface makes explicit to the user what is going on – see the image below for the exact representation of the command being run.

[16] Tech Talk: Linus Torvalds on Git. http://www.youtube.com/watch?v=4XpnKHJAok8#t=18m05s. Accessed: 2014-10-06.

[17] Git push. http://git-scm.com/docs/git-push. Accessed: 2017-02-07.

Figure 1: The interface of the *git push* command

To summarize, both branches of the movement use sharing to characterize their activities even though they use it in a more generalized fashion. Furthermore, the sharing practices which take place within free software projects often have a commercial relevance through the relationships the projects maintain with business companies. However complicated the relationships may initially seem to be, it is necessary to acknowledge that the monetization of source code takes place in a different organizational setting and is thus clearly separated from the practices of free software projects themselves. Finally, I attempted to demonstrate that while free software projects have similar aims to UGC platforms with regard to avoiding sources of friction in sharing, they still design solutions that require conscious decisions and make their functionality explicit.

3.2 Sharing on UGC Platforms

Given the findings above, a claim that Facebook employs the same practice of sharing as free software may seem surprising. However, as I will attempt to show in the following paragraphs, this claim is made by representatives of both sides. Particularly revealing in this regard is a letter to potential investors which was published in 2012 when Facebook was about to enter the stock market.[18] Mark Zuckerberg used the letter to present the dual mission of the company: to open up a space for sharing of information among various parties (individuals, companies or even

[18] Mark Zuckerberg's Letter to Investors: "The Hacker Way". http://www.wired.com/2012/02/zuck-letter/. Accessed: 2017-02-07.

governments) and to create a profitable source of revenue while doing this. At the end of the letter, Zuckerberg devotes several paragraphs to describe how the goals he laid out are to be achieved. This section of the letter is titled ‘The Hacker Way’ and it summarizes how the inner organizational structure of the company has been influenced by elements of what is commonly known as the ‘hacker culture.’

In the introductory paragraphs describing Facebook’s ‘hacker way’, Zuckerberg does away with the negative connotations of the word hacking by making an argument I have seen numerous times employed by free software hackers to differentiate themselves from whom they call ‘crackers’ or ‘black hats’. Here, Zuckerberg claims that hacking is essentially about testing the boundaries of what can be achieved and that although the results of hacking can be used for either good or bad, most hackers strive for a positive impact on the world.

Then the author goes on to make what could be seen as a paraphrase of one of the most famous free software principles (‘Release early, release often’) from the essay *The Cathedral and the Bazaar* by Eric Raymond (1999, p. 28) which represents one of the constitutive texts of the free software movement (or more precisely, its open source branch). This paraphrase is further extended by the statement: ‘We have the words “Done is better than perfect” painted on our walls to remind ourselves to always keep shipping.’[19] I met this exact motto in August 2013 when I visited GNOME’s user and developer conference which represents one of the major annual events in the free software context. Although it is not clear in which direction this motto traveled, it points to a discursive interconnection of the two contexts of sharing.

[19] Mark Zuckerberg’s Letter to Investors: “The Hacker Way”. http://www.wired.com/2012/02/zuck-letter/. Accessed: 2017-02-07.

In the subsequent paragraph, Zuckerberg claims the presence of another ‘hacker mantra’ in the company’s offices. This time, it is the motto ‘Code wins arguments’ which, by now is hard to associate with any particular author, but which also exhibits a very close resemblance to a well-known quote from Linus Torvalds (‘Talk is cheap. Show me the code.’) that originated in 2000 from a message in the Linux kernel mailing list.[20] Finally, by the end of the letter, Zuckerberg also advocates the values of openness and meritocracy – values I found to be praised across the free software movement, even directly in the free software project I did my fieldwork in.[21]

Given this striking resemblance of Zuckerberg’s description of Facebook’s ‘hacker way’ to the cultural elements prevalent in free software, it is hard not to see what type of hacking he refers to. Moreover, these references have been validated form the side of free software by a representative of the Linux Foundation. Jim Zemlin, the CEO of the foundation referred to the letter at length in his TEDx talk to provide evidence that the principles of free software (Linux is licensed under the GNU General Public License mentioned earlier) are consistent with successful business practices.[22] But this part of the letter has generated even further responses. A more elaborate commentary to Zuckerberg’s letter was published by Steven Levy – the author of *Hackers: The Heroes of the Computer Revolution* – in the Wired magazine.[23] In it, Levy draws a

[20] Linux Kernel Mailing List. http://lkml.org/lkml/2000/8/25/132. Accessed: 2017-02-07.

[21] Git http://wiki.pitivi.org/wiki/Git. Accessed: 2017-02-07.

[22] What the Tech Industry Has Learned from Linus Torvalds: Jim Zemlin at TEDxConcordiaUPortland. http://www.youtube.com/watch?v=7XTHdcmjenI. Accessed: 2017-02-07.

[23] Mark Zuckerberg, the Hacker Way and the Art of the Founder’s Letter. http://www.wired.com/2012/02/zuckerberg-hacker/. Accessed: 2017-02-07.

connection between the world-views of the ‘original hackers’ and the mission articulated in the letter:

> Indeed, many of the original hackers at MIT eschewed the traditional view of privacy—they hated passwords, for instance—on the basis that making information accessible promotes more general understanding of systems, the better to hack and improve those systems. Not exactly the same thing as letting everyone see your wedding pictures, or telling them that you’re listening to the new tUnE-YarDs record. But there is a genuine connection between the world-view of original hackers and the vision of Facebook. Zuckerberg, as a hacker, believes that sharing information increases its value.[24]

Levy’s (2010) book provides materials from several ‘noteworthy hackers’, including Bill Gates, Steve Wozniak, Mark Zuckerberg and also Richard Stallman, the author of the original GNU General Public License. Casting Stallman, the privacy concerned free software puritan, into the same crowd as the founder of a service which has been repeatedly criticized for withholding information and breaching privacy may seem to be an odd choice. Yet as the paragraph above illustrates, Levy sees the similarity in the belief that ‘sharing information increases its value’. However, he also acknowledges that the type of information involved in sharing varies significantly – in this case, there has been a shift between sharing consciously produced technical information and routinely shared personal data.

This shift is a result of tendencies which were accounted for already in 2005 in a document published Tim O’Reilly – a promoter of the principles of open source software influential in the commercial environment and the founder of a successful publishing house O’Reilly Media. In it he attempted to summarize the prospects of web 2.0 in terms of business

[24] Mark Zuckerberg, the Hacker Way and the Art of the Founder’s Letter. http://www.wired.com/2012/02/zuckerberg-hacker/. Accessed: 2017-02-07.

opportunities and software design. The document contains a recommendation of 8 specific design patterns, three of which are of particular interest to us here:

> **Data is the Next Intel Inside** Applications are increasingly data-driven. Therefore: For competitive advantage, seek to own a unique, hard-to-recreate source of data.
> **Users Add Value** The key to competitive advantage in internet applications is the extent to which users add their own data to that which you provide. Therefore: Don't restrict your "architecture of participation" to software development. Involve your users both implicitly and explicitly in adding value to your application.
> **Network Effects by Default** Only a small percentage of users will go to the trouble of adding value to your application. Therefore: Set inclusive defaults for aggregating user data as a side-effect of their use of the application.[25]

The three patterns constitute a logical succession. In the first, the author claims that one of the biggest opportunities lies in data aggregation and that competitive advantage will stem from holding unique sources of data. In the remaining two, he provides a way to gain the advantage. O'Reilly notes that valuable data come from application users. However, only a minority of users will exert additional effort to contribute data voluntarily and so the default settings for data sharing should encourage it implicitly or explicitly, that is, with or without being apparent to the user. In other words, the default settings should allow users to *share* their data without restrictions, they should be 'inclusive.' These design patterns correspond directly to the way the Facebook like button is built in the sense that it introduces implicit functionality in order to make sharing of data more widespread. They may seem analogical to the design decisions avoiding various sources of friction in free software, but the implicitness of sharing is present only in the

[25] What is web 2.0: Design patterns and business models for the next generation of software. http://www.oreilly.com/lpt/a/1. Accessed: 2017-02-07.

context of UGC platforms.

To wrap my argument around, I hope to have provided evidence that there is a discursive connection embedded in the concept of sharing between the subversive hacker culture embodied in the free software movement and the business friendly goals of various web services. Christopher Kelty (2008) already claimed that the cultural significance of free software stems from the fact that its principles have spread to other areas and inspired independent initiatives within them. In this sense, UGC platforms such as Facebook can be seen as ‘modulations’ (Kelty, 2008, p. 16) of some of the free software principles. This term is usually reserved for less mainstream and business oriented initiatives such as Creative Commons or Wikipedia but that only means that in this case, the value of sharing travelled a longer distance and underwent a more substantial shift in its meaning.

Overall, the contexts seem to differ in three important aspects I summarized in the table below. When we compare the practice of sharing in the context of free software projects and in the context of UGC sites, we can see that the type of information that is being shared changed from technical information (software source code) to personal information. Furthermore, implicit (unconscious) sharing was introduced to UGC sites to supplement the explicit (conscious) variant employed in free software. Finally, monetization, which in the context of free software is taking place in separate organizational contexts of companies supporting independent projects has been merged to a single organizational context provided by a company operating the given UGC platform. I will discuss the implications of these shifts in the next section.

Table 1: Differences in sharing between free software and UGC platforms

4. Discussion

In the preceding section, I attempted to provide evidence for my earlier claims about the use of sharing in the contexts of free software projects and UGC platforms while drawing an explicit connection between them. Much of the design decisions related to sharing have to do with securing resources for the operation of their parent organizations. As a result, the implications of these design decisions can best be seen when related to the ways connecting the sharing practices to market economy.

Free software, through the influence of its open source branch, has become one of the standard approaches to software development associated with private companies operating on the market. This finding warrants a revision of the relationship between free software – or what Yochai Benkler (2002, 2006) calls peer-production in general – and capitalist forms of production described in the existing literature (Coleman, 2009; Himanen et al., 2001; Söderberg, 2008). Furthermore, UGC sites such as Facebook are attempting to broaden the bounds of sharing to involve contractual exchange from the very beginning of their history. But the fact that these two contexts of sharing both have some connection to market does not mean they have the same implications for the parties involved. On the contrary, it seems that the decisive factor of such implications is not whether or not a practice of sharing is connected to the market. The most important matter in this regard seems to be *how* (under what conditions) the produced content enters the market economy.

The way sharing is used in the differing contexts could be described with the help of analytical distinctions elaborated earlier (Kennedy, 2016). However, these distinctions seem to be conflated by free software projects and UGC sites alike. When free software developers share their source code, they do not distinguish whether they do so with their most immediate peers, other occasional contributors, anonymous users, or private companies – all of which are commonly included in the term ‘community.’ On the other hand, Facebook too is conflating significant others with contractual parties in their involvement in sharing. And this conflation too brings the platform significant financial resources for its operation. However, nor the transfer of the shared production, neither the arrangements between the parties are channelled publicly in a way that would warrant scrutiny by other involved parties, especially users as the originators of the shared data.

If a more general implication is to be drawn from the findings above, it is that although the set of distinctions drawn from the literature allows us to navigate the diverse contexts of sharing in general terms, it does not provide clear-cut criteria for distinguishing between the contexts. Perhaps this warrants an introduction of a new set of criteria based on the dividing lines identified in the empirical analysis above and summarized in Table 1. These criteria could help us to categorize the particular contexts and meanings of sharing in the digital environment in a more unequivocal way.

5. Conclusion

It was the main goal of this text to demonstrate that the meaning of sharing is highly context dependent and that although parties from various contexts claim to be doing the same thing, the actual practices associated with sharing differ substantially. The importance of these differences lies in the fact that they result in differing levels of control that producers (end-users or software developers) exert over their products (source code, user content, or user data) and in the level of awareness about the arrangements of such production.

As it is often the case with qualitative research, the limitations of this study lie in the uncertainty of the extent to which the described phenomenon is representative of the overall empirical field. In this sense, the reliability of the findings presented above is dependent upon whether the cultural elements of free software presented above provide and authentic picture of the movement as a whole and whether Facebook is representative of the sharing practices employed by the UGC platforms in general. However, the second limiting point could very well be also seen as a vantage point for further research which would investigate, if the new set of distinctions also applies to other platforms and initiatives which claim to be practicing sharing.

References

Banning, M. E. (2016). Shared entanglements – Web 2.0, info-liberalism & digital sharing. *Information, Communication & Society*, 19(4), 1–15.

Barta, K., & Neff, G. (2016). Technologies for sharing: Lessons from quantified self about the political economy of platforms. *Information, Communication & Society*, 19(4), 1–14.

Belk, R. (2007). Why not share rather than own? *The Annals of the American Academy of Political and Social Science, 611*(1), 126–140.

Belk, R. (2010). Sharing. *Journal of Consumer Research*, *36*(1), 715–734.

Benkler, Y. (2002). Coase's penguin, or, Linux and "the nature of the firm". *Yale Law Journal, 112*(3), 369–446.

Benkler, Y. (2006). *The wealth of networks: How social production transforms markets and freedom*. New Haven: Yale University Press.

Bergquist, M., & Ljungberg, J. (2001). The power of gifts: Organizing social relationships in open source communities. *Information Systems Journal*, *11*(4), 305–320.

Bodle, R. (2011). Regimes of sharing: Open APIs, interoperability, and Facebook. *Information, Communication, & Society*, 14(3), 320–337.

Brewer, J. (2000). *Ethnography*. Buckingham: Open University Press.

Chełkowski, T., Gloor, P., & Jemielniak, D. (2016). Inequalities in open source software development: Analysis of contributor's commits in Apache software foundation projects. *PLoS ONE, 11*(4), 1–19.

Chopra, S., & Dexter, S. (2007). *Decoding liberation: The promise of free and open source software*. New York: Routledge.

Coleman, G. (2009). Code is speech: Legal tinkering, expertise, and protest among free and open source software developers. *Cultural Anthropology, 24*(3), 420–454.

Coleman, G. (2013). *Coding freedom: The ethics and aesthetics of hacking*. Princeton: Princeton University Press.

Cook, J., Laidlaw, J., & Mair, J. (2009). What if there is no elephant? Towards a conception of an un-sited field. In M. A. Falzon (Ed.), *Multi-sited ethnography: Theory, praxis and locality in contemporary research* (pp. 47–72). Farnham: Ashgate Publishing, Ltd.

Dafermos, G., & Söderberg, J. (2009). The hacker movement as a continuation of labour struggle. *Capital & Class, 33*(1), 53–73.

Falzon, M. A. (2012). *Multi-sited ethnography: Theory, praxis and locality in contemporary research*. Farnham: Ashgate Publishing, Ltd.

Freeman, S. (2007). The material and social dynamics of motivation. *Science Studies, 20*(2), 55–77.

Geertz, C. (1973). *The interpretation of cultures: Selected essays*. New York: Basic Books.

Himanen, P., Castells, M., & Torvalds, L. (2001). *The hacker ethic and the spirit of the information age*. New York: Random House.

John, N. A. (2012). Sharing and web 2.0: The emergence of a keyword. *New Media & Society, 15*(2), 1–16.

John, N. A. (2013). The social logics of sharing. *The Communication Review, 16*(3), 113–131.

John, N. A., & Sützl, W. (2016). The rise of "sharing" in communication and media studies. *Information, Communication & Society, 19*(4), 437–441.

Karger, T. (2016). *The weight of the intangible: Knowledge networks in free and open source software development*. Olomouc: Palacký University.

Kelty, Ch. (2008). *Two bits: The cultural significance of free software*. Durham: Duke University Press.

Kennedy, J. (2016). Conceptual boundaries of sharing. *Information, Communication & Society*, 19(4), 1–14.

Kuk, G. (2006). Strategic interaction and knowledge sharing in the KDE developer mailing list. *Management Science*, *52*(7), 1031–1042.

Levy, S. (2010). *Hackers: Heroes of the computer revolution*. Sebastopol, CA: O'Reilly Media.

Marcus, G. E. (1995). Ethnography in/of the world system: The emergence of multi-sited ethnography. *Annual Review of Anthropology*, *24*(1), 95–117.

Matei, S. A., & Dobrescu, C. (2011). Wikipedia's "Neutral Point of View": Settling conflict through ambiguity. *The Information Society*, 27(1), 40–51.

Mauss, M. (2002). *The gift: The form and reason for exchange in archaic societies*. London: Routledge.

Neyland, D. (2008). *Organizational ethnography*. Thousand Oaks, CA: Sage.

Osterloh, M., & Rota, S. (2004). Trust and community in open source software production. *Analyse & Kritik*, *26*(1), 279–301.

Raymond, E. (1999). The cathedral and the bazaar. *Knowledge, Technology & Policy*, *12*(3), 23–49.

Rybas, N., & Gajjala, R. (2007). Developing cyberethnographic research methods for understanding digitally mediated identities. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, *8*(3).

Shah, S. K. (2006). Motivation, governance, and the viability of hybrid forms in open source software development. *Management Science*, *52*(7), 1000–1014.

Söderberg, J. (2008). *Hacking capitalism*. New York: Routledge.

Sundararajan, A. (2016). *The sharing economy: The end of employment and the rise of crowd-based capitalism*. Cambridge, MA: The MIT Press.

Sützl, W. (2014). The anti-economy of sharing. *Else Art Journal*, 1(0). 122–141.

Teli, M., Pisanu, F., & Hakken, D. (2007). The internet as a library-of-people: For a cyberethnography of online groups. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, *8*(3).

van Dijck, J. (2009). Users like you? Theorizing agency in user-generated content. *Media, Culture & Society*, *31*(1), 41–58.

van Dijck, J. (2013). *The culture of connectivity: A critical history of social media*. New York: Oxford University Press.

Wittel, A. (2011). Qualities of sharing and their transformations in the digital age. *International Review of Information Ethics*, *15*, 3–8.