



An Ethical Obligation to Copy – the Origins and Implications of Free Software

Karl Arthur Giverholt

Peer production has become an important way to manufacture digital products and information, for example software and encyclopedias. The products are generally made without payment to contributors, and may most often be used free of charge. They can also be modified, copied and shared by the users. This paper examines one of the first examples of a product built on this business model, the operating system GNU/Linux, and suggests that the prominence of digital tools and information, which may be copied and modified without detriment to the original, prompts us to rethink traditional conceptions of ownership, work and technology, based on ethical rather than economic considerations.

In its beginning as a publicly accessible communication channel, the World Wide Web was more or less one-way, with a clear demarcation between creators and consumers of content. With the emergence of what many have taken to be a second generation of network communication, this has changed, and users are now free to comment on, alter or personalize much of the content they encounter while surfing. The much discussed and visited websites Wikipedia and YouTube, and the blogosphere in general, are produced entirely by users with almost no editorial constraints. This development has caused excitement and uncertainty in the worlds of business, journalism, entertainment and technology development.

The questions and surprises arising are manifold: Will traditional journalism be able to maintain its influence in a world where anyone with access to an online computer can report on any story s/he sees fit? How will the business models of the entertainment industry keep up with the demands of a public that is not only getting used to having access to the latest pop cultural products instantaneously and free of charge, but is increasing its own role in producing music and video, and distributing its own artwork with very low production costs? Even the institution that is Encyclopedia Britannica is being seriously challenged by the efforts of thousands of volunteer contributors to Wikipedia, and actors in the corporate world are beginning to look at ways of outsourcing their research and development to the

general public. How will this alter the conception of accountability, reference material and intellectual property? To summarize: In what ways will peer production¹ shape the economy and society of tomorrow?

This is not a question that can be answered with any degree of certainty, but in this paper I will try show how the phenomenon of decentralized collaborative effort may have implications for our thinking about traditional questions in matters political, legal and technological, arguing that peer production should be seen to be based on ethical, rather than economical, incentives. My basis for examining these questions will be a collection of computer software known as GNU/Linux². This project was founded in 1984 by Richard Stallman, who was at the time working in the Artificial Intelligence Lab at MIT, and from the very beginning, the ethical and political aspects of the project were explicitly formulated by way of Stallman's definition of free software³.

Most of this paper will focus on GNU/Linux in general, but when specific references are needed, I will turn to the South African-based distribution⁴ Ubuntu, currently the most widespread. Starting out, I will look at the more technical aspects of GNU/Linux, explaining in what ways this operating system differs from the commercial alternatives MacOS and Windows. By taking the cue from Sherry Turkle's notion of text-based operating systems as modern and graphic user interfaces as post-modern, as she presents it in her book *Life on the Screen – Identity in the Age of the Internet*, I will argue that GNU/Linux is able to keep the best of the two worlds and that it constitutes something entirely different. I will go on to see how scholars working in the anthropological, legal and economic field have written about peer production, and finally discuss what I see to be the social implications of the phenomenon. In this discussion, I will employ the writings of Andrew Feenberg, and especially his treatment of the philosophical standpoint presented in Martin Heidegger's "The Question Concerning Technology". Before starting, however, I find it necessary to explain briefly why I have chosen these particular peer production phenomena, and why I find the question to be both compelling and important.

The potential of free software

GNU/Linux is a highly interesting collection of software for a variety of reasons, but I find it especially compelling because, in a world that seems to be increasingly preoccupied with financial gain, it is surprising that a large number of people should be willing to spend their free time not only developing software, but also in helping others learn how to use this software. The collaborative effort of volunteers has come up with a product that is able to compete, in quality if not in popularity, with the products of some of the biggest brands in corporate America. To me, this suggests that the future of technology is not necessarily in the hands of financial giants. With the impasse of peer production in more visible areas such as entertainment and journalism, mass collaboration has caught the eye of more

1 This term was coined by Yochai Benkler in "Coase's Penguin", and I will use it in this paper in the same way he does: as the decentralized collaboration by self-identifying actors on a product in which the actors have no individual economic interest (Benkler: 375 – 380)

2 In general, this operating system is referred to merely as Linux. However, Linux is only the kernel (the part of the operating system that lets the different pieces of hardware function together) while GNU is the collection of software originally used along with the kernel to let the user and machine interact. For an exhaustive explanation of why GNU/Linux is a more accurate term, see "What's in a name?" in Stallman, 2002.

3 Free software denotes software that is unrestricted, not necessarily free of charge, as opposed to proprietary software. The use of proprietary software often requires the user to sign so called End User License Agreements, which limits users' right to copy and modify the software, effectively making it the property of the seller, while the user merely buys the right to use it. For more, see "Free Software Definition" and "The Gun Manifesto" in Stallman, 2002

4 GNU/Linux is available in roughly one hundred versions, known as distributions. For more information, rankings, reviews etc., see <http://www.distrowatch.com>

than a dedicated fan-base, making it all the more prudent to examine the implications of this emerging scene.

The exponential growth of venues such as the blogosphere, YouTube, MySpace and Facebook has been explained in part by the increased desire among teens and young adults to present and promote themselves to a larger audience than their immediate family and friends⁵. The attraction of anonymous contributions, however, cannot be of the same kind. In the case of Wikipedia, one could argue that most people will never be able to partake in the writing of a traditional encyclopedia, and so doing it for free is the only way to see one's knowledge being utilized in this way. But in the case of GNU/Linux contributors, this argument does not hold, as there is an ever-growing use for skilled programmers in high paid positions (in fact, many programmers involved with GNU/Linux already hold such positions), and the willingness of businesses and private users to pay for technical support is significant. I will examine the question of motivations in a later section of this paper.

As to the choice of Ubuntu as my example, I believe the reason Ubuntu has become the most popular distribution so quickly is the commitment made by the developers to user-friendliness and openness to beginners, to which the activity on the Ubuntu forums⁶ testifies clearly. There are technical aspects that lay the basis for the community effort, so I will in the following section examine the technicalities that separate GNU/Linux from proprietary operating systems.

Technical aspects

The most accurate metaphor of operating systems is that of language. Microsoft Windows speaks one language, MacOS another. These two operating systems are quite similar on the surface, in that they are both object-oriented languages, based on the use of the mouse to manipulate visible, spatial symbols, collectively known as “graphical user interfaces” (GUIs). A few idiosyncrasies aside, such as whether to use the ctrl-key or the apple key for shortcuts, translation between the two is relatively uncomplicated.⁷ As with all languages, users learning a foreign one will find similarities and differences in comparison with their mother tongue, but will also find that some structures are universal; for example, some signs signify actions, while other signs point to objects. Knowledge about some basic signs will normally ensure communicability within the new language, so Windows users can often find their way around a machine running on MacOS, because the two languages share large portions of their vocabulary (much like the way a French-speaking person may talk to a Spanish native with only a limited amount of practice).

Before Microsoft launched Windows, however, the languages mediating between user and machine were as different as a spoken language and the visual language used by people with hearing impairments. Microsoft DOS employed a text-based “command line interface” (CLI), using written commands to perform tasks, as opposed to the manipulation of symbolic objects. This was seen as a more direct way of communicating with the machine⁸, because it employed language the machine understood, not visual simulations of these commands more intuitive for users, and it was the only

5 In regards to my understanding of the social, cultural and psychological aspects of a networked society, I am deeply indebted to the insights offered by Thomas de Zengotita in his recent book *Mediated – How the Media Shapes Your World and the Way You Live In It*.

6 <http://www.ubuntuforums.org>

7 This is, however, only the surface; when it comes to the way the OS interprets the applications, the two systems are as different as French and Chinese. This is why Microsoft software does not work on an Apple machine unless designed specifically for it and vice versa.

8 This is a very problematic notion, because there are always deeper levels of interaction going on in a computer: If you look below the commands you find the programming language or source code, below that the binary code, and below that the way the binary code makes the hardware behave in certain physical ways. Both the limits of this paper, and, more importantly, my lack of technological knowledge, prevent me from going into this question.

possible way of working on a GNU/Linux computer. Sherry Turkle describes this phenomenon in the chapter “A Tale of Two Aesthetics” of her book *Life on the Screen*, using the differences to illustrate a much larger cultural shift:

Postmodern theorists have suggested that the search for depth and mechanism is futile, and that it is more realistic to explore the world of shifting surfaces than to embark on a search for origins and structure. Culturally, the Macintosh has served as a carrier object for such ideas. The modern and postmodern aesthetics were locked in competition for the second half of the 1980s with the IBM personal computer becoming the standard bearer on the modernist side. The myth of the Macintosh was that it was like a friend you could talk to; the myth of the IBM [...] was that the computer was like a car you can control (Turkle: 36).

Staying within this language, it seems like modern computing definitely yielded to postmodern with the introduction of Windows XP, Microsoft's first OS without the text-based MS-DOS underneath.

The difficulties involved in learning a new language is one of the main obstacles for computer users wanting to change OS, and the threshold for moving from a GUI to a CLI are particularly high. In addition to the acknowledgment of the many advantages GUIs can offer, the wish to attract new users prompted GNU/Linux developers to design distributions with the *possibility* of daily use by virtual object manipulation, the most widely known GUIs being Gnome and KDE⁹. I stress possibility, however, because the idea of replacing the CLI altogether would seem ludicrous to most GNU/Linux users, and certainly all developers. The consensus seems to be that the GUI is easier to use, while the CLI offers more control, more possibilities and is more efficient, so that users will employ the latter more as they grow in experience and/or their needs become more advanced¹⁰. In other words, the two languages or aesthetics are set to work together rather than one being favored.

This does not, however, justify the position I took in the introduction about GNU/Linux being different altogether. A second distinction needs to be made. Maintaining the metaphor of the OS as a language, imagine that you speak a language fluently, that you can communicate without misunderstandings, but when you decide to begin studying linguistics, you find that the books you need are locked up in a vault, controlled by the owners of that language. Similarly, buyers of proprietary software may use the program, but are not allowed to know *how* it functions¹¹. This is not the case with free software, because users of free software have access to the source code¹² of the programs they are using. However, mere access to the meta-language is not my central point. The point is that if you know the meta-language, and are not completely satisfied with the language, you can change it. Because of the history and philosophy behind free software, you are encouraged not to remain unhappy with a program that does not fulfill your needs; you are told you should change it into something better, and to share your efforts of improvement with the community, by making your derived product publicly available. Let us take a closer look at this history and philosophy.

Origins of free software

Paradoxically, the history of free software begins with the implementation of restrictions on the software developed and used by the programmers working at the AI-lab at MIT in the 1970s and early

9 Both of these GUIs are highly flexible interfaces, with individual strengths and weaknesses. They can be made to look and function largely in the same way as Windows, or quite differently, according to taste and utility. For more information, see <http://www.gnome.org> and <http://www.kde.org>

10 For discussions about the need for a CLI, and the differences between CLIs and GUIs, see for instance http://linux.about.com/cs/softofficeutility/a/gui_cli.htm or <http://www.computerhope.com/issues/ch000619.htm>

11 Ref. Footnote 3, above.

12 Ref. footnote 8, above.

1980s (Stallmann: 15). Before this time, according to Stallman, software was distributed freely among the users, and no one would consider denying a colleague access to the code that made the software work, so that it could be modified according to individual needs. When the corporations producing hardware introduced non-disclosure agreements, however, this environment changed radically. As Stallman puts it: “This meant that the first step in using a computer was to promise not to help your neighbor. A cooperating community was forbidden. The rule made by the owners of the proprietary software was, 'If you share with your neighbor, you are a pirate. If you want any changes, beg us to make them'” (16).

Seeing the emerging scene as intolerable, Stallman decided to launch the GNU project, and he described his goals, and the means for reaching them, in the GNU Manifesto:

GNU, which stands for Gnu's Not Unix¹³, is the name for the complete Unix-compatible software system which I am writing so that I can give it away free to everyone who can use it. Several other volunteers are helping me. Contributions of time, money, programs and equipment are greatly needed. [...] In the long run, making programs free is a step towards the post-scarcity world, where nobody will have to work very hard just to make a living. People will be free to devote themselves to activities that are fun, such as programming, after spending the necessary ten hours a week on required tasks such as legislation, family counseling, robot repair and asteroid prospecting. (31, 39)

Writing a functioning operating system from scratch necessarily takes time, and it was not until Linus Torvalds launched the kernel Linux in 1991, and the GNU system and Linux kernel were set to work together in 1992, that GNU/Linux would be able to function as the sole operating system on any given machine¹⁴. Since then, development has continued at a steady pace, with some major landmarks being the first release of a graphic desktop environment (KDE) in 1996, the first free, graphic web browser (later to become Firefox) in 2000, and the first free office package (later to become OpenOffice.org) in 2002.

In “The GNU Manifesto,” Stallman refers to Kantian ethics and the Golden Rule as reasons to why users should have the right to modify and copy their software (36). But in the essays collected in *Free Software, Free Society*, he also presents original legal and ethical reflections based on the American constitution and freedom as an ultimate value. His views on copyright are especially provocative, and I will spend some time describing and commenting on these ideas.

The law granting authors copyright over their work was implemented in the constitution, Stallman insists, for the benefits of the general public, not the authors (77). It was believed that the possibility for authors to copyright their work for a limited amount of time, and thereby make money on it, would encourage authors to write, hence increasing the public's access to valuable information and thoughts. The writing was seen as being provided for the sake of the public, but the accessibility of the works had to be restricted to some degree to ensure the production of such works. However, “the copyright system [that provides] privileges for publishers and thus benefits to publishers and authors, [...] does not do this for their sake. It is like a government purchase of a highway or an airplane using taxpayers' money, except that the government spends our freedom instead of our money.” (78) Stallman argues that the discussion of copyright questions today is often based on the wrongful assumption that it is possible for the sake of authors (or, in effect, the publishers), or that it is intended only to “strike a balance” between the publishers and the public (78). The constitution clearly states,

13 UNIX was the preferred operating system for networked computers in university settings at the time. Recursive acronyms such as GNU are popular amongst hackers in naming their products.

14 For an extensive description of the events leading up to this collaboration, see the Wikipedia entry on GNU/Linux: <http://en.wikipedia.org/wiki/GNU/Linux>

however, that the interest of the public takes priority in these issues¹⁵. Using the field of science as an example, Stallman claims that today copyright may in fact decrease the public's access to information: "Many journal publishers appear to believe that the purpose of scientific literature is to enable them to publish journals so as to collect subscriptions from scientists and students. Such thinking is known as 'confusion of the means with the ends'" (87).

Certainly, copyright had its place in a time when it was mainly concerned with the written word, and the means of mass copying these words was out of the reach of the general public. There can be little doubt that if any publishing house could buy a single copy of a book in a bookstore, and begin printing and selling it on its own, the financial risks of publishing would make books far less accessible. Today, however, the situation has changed: "Computers and digital information technology are bringing us back to a world more like the ancient world, where anyone who can read and use the information can also copy it and make copies about as easily as anyone else. [...] So the centralization and economy of scale introduced by the printing press and similar technologies are going away" (136). As an alternative to the copyright system as it functions today, Stallman has introduced the legal device "copyleft,"¹⁶ an agreement under which software and other information-products are free to be modified and copied (or in the case of non-technical products such as text, music, etc., free for verbatim reproduction) as long as the derived products are issued under the same terms.

Stallman seems to think that the most important reason copyright on software exists today is that people make nonsensical comparisons between software, which can be shared without diminishing the value of the original, and material objects¹⁷:

People ask silly questions like, 'Well, should hardware be free?' 'Should this microphone be free?'¹⁸ Well, what does that mean? Should you be free to copy it and change it? Well, as for changing it, if you buy the microphone, nobody is going to stop you from changing it. As for copying, nobody has a microphone copier yet. [...] Maybe some day there'll be nanotechnological analyzers and assemblers, and it really will be possible to copy a physical object, and then these issues of whether you're free to do that will start being really important. We'll see agribusiness companies trying to stop people from copying food, and that will become a major political issue, if that technological capability will ever exist (134).

So this is the potential importance of software freedom in an imagined future, questions that, if they ever arise, will undoubtedly cause a much more extensive discussion – if matters like these are still being discussed at this hypothetical time – than free software has been able to. However, there are, as Stallman sees it, more contemporary reasons for the importance of free software:

[I]t's obvious that professional programmers can make use of this freedom very effectively, but not just them. Anybody of reasonable intelligence can learn a little programming. [...] Now, if you don't have this freedom, it causes practical, material harm to society. It makes you a prisoner

15 Under section 8 in the U.S. Constitution, listing the powers of Congress, it is stated: "To promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries." In other words, the purpose is to further science and useful arts, and the means is copyright. (http://en.wikisource.org/wiki/Constitution_of_the_United_States_of_America)

16 For more information about copyleft and the legal documents describing this device, see <http://www.fsf.org/licensing/essays/copyleft.html>

17 It is important to understand that computer software, even if it comes in the form of a physical CD, is nothing more than information, that is, information about how to manipulate the components in a computer in order to get it to perform a specific task.

18 The essay "Copyright and Globalization in the Age of Computer Networks" was originally given as a lecture at MIT in April 2001.

of your software. [...] But it also affects people's morale. If the computer is constantly frustrating to use, and people are using it, their lives are going to be frustrating, and if they're using it in their jobs, their jobs are going to be frustrating [...] And you know, people protect themselves from frustration by deciding not to care. [...] And when that happens, it's bad for those people, and it's bad for society as a whole. [...] [F]or beings that can think and learn, sharing useful knowledge [i.e. software] is a fundamental act of friendship. [...] And, in fact, this spirit of goodwill [...] is society's most important resource. [...] Nowadays, according to the U.S. Government, teachers are supposed to do the exact opposite [of telling kids to share the candy they have brought to school]. 'Oh, Johnny, you brought software to school. Well, don't share it. Oh no. Sharing is wrong.' [...] I think this is the most important reason why software should be free: we can't afford to pollute society's most important resource. [...] It's a psycho-social resource [...] and it makes a tremendous difference to our lives (164-165).

Admittedly, Stallman is not an expert on legal matters, but according to Lawrence Lessig, a professor of law at Stanford University who wrote the introduction to *Free Software, Free Society*, Stallman's reasoning and argument on copyright and its need to undergo change is sound¹⁹. When it comes to the ethical questions, some reservations should be made. First, Stallman sees freedom as a fundamental human value, and any restriction on this value may only be justified if the trade-off yields clearly positive results. I agree with his view on freedom, but this is not a given, and differing opinions may find his ethical reasoning flawed or even invalid. Second, while he constantly claims ethical purposes, it could be argued that his stance is more political than philosophical, and there has been no lack in accusations of him being little more than a communist. (Stallman himself denies this, pointing to what he sees as the complete lack of freedom to copy information in Soviet Russia (138).

It is certainly not hard to defend his argument as long as it revolves around the responsibility to share and help others, but the weight he places on the psycho-social impacts of computer software may seem somewhat exaggerated, leaving open the question of whether this might not be a far more important issue for programmers than society at large. In my experience, people who spend their time working with or through computers, but not directly on them, may complain and become frustrated with their tools, but this does not cause them to stop caring about the work itself. Looking at the grander scheme of things, it certainly does not seem to be a bad idea to root the idea of freedom and community in the equipment many of us use every day, but it remains an open question of how much effect this will have as long as most people probably would not know much about the foundations of their software, free or not.

The most compelling aspect of Stallman's work is the economic questions his work implicitly raises. Free software does not mean that it *must* be distributed free of charge, but since it must be freely copyable, it must in theory be possible to acquire any program without paying for it. (In Stallman's vision, the way of reaching a post-scarcity society would be to “get rid of the corporate control over the economy and the laws” (149).) I think it may be of importance whether our tools are shaped by corporations or communities, and this question will be the basis of the next section.

Motivation and the economy

The name of the Ubuntu distribution of GNU/Linux comes from an African term meaning “humanity to others.” The community aspect of the software is central to this project, and the developers are in close contact with the users through, for instance, the IdeaPool²⁰, a wiki-page where

19 Lessig's introduction, as well as all of Stallman's essays published in this book, are available free of charge from the GNU website: <http://www.gnu.org/doc/book13.html>. For more of Lessig's theoretical work, see his own website:

<http://www.lessig.org>

20 <https://wiki.ubuntu.com/IdeaPool>

anyone may post suggestions for improvements or features they would like implemented in upcoming releases, marketing ideas, ideological points, etc. My impression from talking to some of the contributors to The Ubuntu Beginners Team (a group of users dedicated to the systematic improvement of online user manuals for beginners and general lowering of the threshold for entering the world of GNU/Linux), suggests that this community commitment is in itself an important motivational factor in offering one's time and resources to a product in which one has no financial interest. Several of the users said they had met helpful, friendly people when they first began learning their way around Ubuntu, and they wanted to become active in order to give something in return²¹. Many merely stated that working on Ubuntu is fun, that it is a good way to learn, or that they like the product and want to contribute to it being as good as possible.

The commitment to the quality of the product is important to elaborate on, because it is not altogether clear exactly what this product is, since it consists of a multitude of parts, some unrelated, some in direct competition with each other. In his essay "Don't fear the penguins," Matt Ratto suggests that the successful establishment of a mascot common to the community, the penguin known as Tux, has been an important factor in holding this community together (Ratto: 828). The glue, he claims, is primarily emotional:

The development of Linux as both a technical object and a social movement has been built on contentious debates about both how it should develop (i.e., what technical features should be included) and what it might mean (i.e., its social ramifications). People care deeply about Linux, a point that goes some way toward understanding how the collective involved in maintaining and developing Linux manages to stay coordinated (830).

The many alterations on the originally slightly obese and peaceful penguin serve to unite the many differing goals and opinions held by participants in the GNU/Linux community, and may be seen as the symbol of a "shared material object [that] can serve as an alternative location for social organization" (832).

According to Yochai Benkler, however, there are more fundamental issues than the possibility of a shared space at stake in the question of how peer production functions. Psychosocial motivational factors remain important to him, and he indeed treats them exhaustingly in the article "Coase's penguin, or Linux and 'The Nature of the Firm'," which contains an extensive analysis of variable modes of rewards. Benkler claims that it "becomes relatively straightforward to see that there will be conditions under which a project that can organize itself to offer social-psychological rewards removed from monetary rewards will attract certain people," and that it is important to "understand that when a project of any size is broken up into little pieces, each of which can be performed by an individual in a short amount of time, the motivation to get any given individual to contribute need only be very small" (Benkler: 378). But in his view, the phenomenon of peer production must be seen as the expression of a much deeper economical-cultural shift: "[T]his emerging third model [the first and second being the market and the firm] is (1) distinct from the other two and (2) has certain systematic advantages over the other two in identifying and allocating human capital/creativity" (381). The peer production model is, according to Benkler, simply more cost-efficient in a networked information-society, because "[t]he widely distributed model of information production will better identify who is the best person to produce a specific component of a project, all abilities and availability to work on the specific module within a specific time frame considered" (414).

Does this mean that the firm no longer has a future? Will the economy of tomorrow be a social

21 I conducted informal interviews with Ubuntu users on IRC, but many have posted their experiences, positive and negative, on the Ubuntu forum: <http://www.ubuntuforums.org/forumdisplay.php?f=103>

space organized around continuously evolving end products designed and produced from the bottom up, or will the firms and corporations find ways to reorganize according to cultural changes, and maintain their hegemony over the economy? It seems that any attempt to answer this would be an effort in more or less educated guesswork, and one of the more heavily researched attempts to predict the future of the economy is the book *Wikinomics* by Don Tapscott and Anthony D. Williams. These authors indeed see vast changes about to happen: “Twenty years from now we will look back at this period of the early twenty-first century as a critical turning point in economic and social history. We will understand that we entered a new age, one based on new principles, worldviews, and business models where the nature of the game was changed” (Tapscott: 19)²².

There is no lack of claims with huge implications in *Wikinomics*; for instance: “The ability to pool the knowledge of millions (if not billions) of users in a self-organizing fashion demonstrates how mass collaboration turns the Internet into something not completely unlike a global brain” (41). However, the main purpose of the research seems to be to inform current business managers as to how they may stay on top of the game, and keep their positions as prominent producers of wealth. Pointing to how Procter & Gamble opened up their research and development and how IBM at an early stage joined the collaboration on GNU/Linux, the reason for change is seen to be economic, not ethical: “Smart firms today understand that sharing is more than playground etiquette. It's about lowering costs, building community, accelerating discovery and lifting all boats in the sea” (281). This way of thinking is not easily transferable to the case of Ubuntu, because in this instance, economic growth has never been a goal. It is in fact the first Ubuntu promise, made on the front page of the official website²³ that “Ubuntu will always be free of charge,” and it is a prominent feature of GNU/Linux in general that it runs on less-than-state-of-the-art hardware, allowing schools and developing world users to use donated or old computers without being shut out from the development race. It is the very nature of peer production, as founded by Richard Stallman, that it does not yield financial gains for owners at the top of a hierarchy, because without the collaborate effort from the bottom, these gains would be impossible. The fact that this is not discussed in *Wikinomics*, in my opinion, renders it a half-told tale.

In many ways, it seems like the phenomena collected under the term peer production are at a crossroads. As corporate interests being to open their eyes and wallets to the input coming from all over the planet, there is no way of saying whether the democratic and ethical foundations will hold or be swallowed in the need for a constantly growing economy. In February 2007, major hardware vendor Dell announced that it would begin selling laptops with Ubuntu pre-installed, as a direct result of consumer demand²⁴; so far, however, not one of the dozens of e-mails I have received as a previous consumer has informed me of this possibility, nor does it figure prominently on their website²⁵. Only time will tell whether the cooperation of a non-commercial actor and a multi-billion-dollar enterprise will leave them both strengthened, and how society at large will respond to changes in business models.

So far, I have attempted to show various ways in which GNU/Linux may point to or partake in future societal changes, small and large, by way of posing alternative views on production and ownership of information. It remains to be asked, however, whether the question of computer software really is all that important, and if so, how we should go about answering it?

Social implications

22 This book is based on several multi-million-dollar research programs, and is, of course, copyrighted, as opposed to Benkler's recent book, *The Wealth of Networks* (Yale University Press, 2006), which is available as a downloadable PDF file and in Wiki-version.

23 <http://www.ubuntu.com>

24 <http://direct2dell.com/one2one/archive/2007/05/01/13147.aspx>

25 <http://www.dell.com>

In “The Question Concerning Technology,” Martin Heidegger attempts to reveal the essence of technology by questioning it. This questioning is of utmost importance to him, because “[e]verywhere we remain unfree and chained to technology, whether we passionately affirm or deny it. But we are delivered over to it in the worst possible way when we regard it as something neutral” (Heidegger: 311-312). Technology, for Heidegger, is not to be understood in our average, everyday use of the word; even more rigorous contemporary definitions are rejected as merely partial, although “correct”: “The current conception of technology, according to which it is a means and a human activity, can therefore be called the instrumental and anthropological definition of technology” (312). Taking the long route through a questioning of the phenomenon of causality, as Aristotle defines it, he ends up with the preliminary notion of technology as “a mode of revealing” (319). Technology distorts man’s relationship to nature by subjecting it to the “unreasonable demand that it supply energy which can be stored” (320), reducing even the mighty Rhine river to a waterpower supplier (321). This ordering of nature is not, however, man’s own doing or decision: “[W]hen man, investigating, observing, pursues nature as an area of his own conceiving, he has already been claimed by a way of revealing that challenges him to approach nature as an object of research. [...] Modern technology, as a revealing that orders, is thus no mere human doing.” (324).

In other words, technology itself imposes on man the technological way of approaching nature; it has (ontologically, not chronologically) *become* autonomous as the Gestell or enframing (324) of the world in which man exists, and this is its essence. Oblivious to the fact that his worldview is enframed by technology, man loses sight of himself in his essence, making revealing in truth impossible (332). And the most astounding problems arise when it is revealed that the essence of technology is not really essential, that the very essence of essence is challenged by the demands of technology: “The essential unfolding of technology threatens revealing, threatens it with the possibility that all revealing will be consumed in ordering (335).” The essay fades out with the loose hope that art might provide a departure from the demands of technology, but it does not seem to be an optimistic Heidegger who writes this. And if Heidegger is right, this would render the whole phenomenon of GNU/Linux irrelevant in the bigger picture, as it would merely be a cog in the machinery that is the destructiveness of technology.

The major problem with this essay is that it amounts to little more than demonstrating the inevitable plight of modern technological man. If technology is in fact autonomous, and to be understood as the very approach to the world that has been the prevailing one in the west for centuries, how can anything short of a complete, world-wide rejection of this paradigm rescue mankind from the danger Heidegger claims that technology inherently is? Certainly, we cannot demand of Heidegger that he should suggest “solutions” where he does not see any, but I find it problematic to accept an argument that seems to point towards the willed rejection of the human impulse to improve tools. It is at this point Andrew Feenberg makes his departure from Heidegger²⁶.

In his essay “From Essentialism to Constructivism,” Feenberg argues that Heidegger's way of seeing technology in terms of its determining of the future of objects does not leave room for the social aspects of it (Feenberg 2007: 2) In very condensed form one could say that while Heidegger sees technology as one necessary, all-encompassing phenomenon, Feenberg insists that it could (and can) potentially be, and in fact has been, radically different while still being technology. These (possible and actual) differences must be accounted for in social terms, he claims. An example of this is the way in which computer networks, originally intended for distribution of data, became a powerful tool for

26 I do not imagine that Heidegger would be in any way satisfied by this departure, and I doubt that it is ontologically a true departure, in that I do not think it fully appreciates the importance and magnitude of revealing in Heidegger's philosophy. What it does, however, which Heidegger cannot, is point towards a future that does not require a complete paradigm upheaval.

communication between human beings, simply because users choose to employ it for means unintended by its designers (9)²⁷. Thus, he concludes, “technology may enframe and colonize, but it may also liberate repressed potentialities of the lifeworld that would otherwise have remained submerged. It is thus essentially ambivalent”(24)²⁸.

The ambivalent essence of technology implies the possibility to choose, and the chapter “Postindustrial Discourses” in *Transforming Technology*²⁹ is dedicated to the explication of how this possibility has emerged and plays out in the world of computers. The need for continuous development in computer technology, he claims, leaves (at least) two paths open: “Automation increases management's autonomy only at the expense of creating new problems that justify workers' demands for an enlarged margin of maneuver. That margin may be opened to improve the quality of self-directed activity, or it may remain closed to optimize control” (Feenberg 2002: 96). Furthermore, this is seen as a sign that the very meaning of work is changing, from production to “a process of communicating and learning organized around the 'reflexivity' of computer technology” (99). According to Feenberg, the unpredictability of computers as they are used resembles the way we perceive the world, in that it is “not given to us as a collection of well-defined objects and problems but as an infinitely rich context of actions” (104), and this calls for “an alternate rationality, a rationality of implementation rather than of planning and control, based on self-referential processes of communicating and learning in the course of using and modifying tools” (106). As we saw in the technical section above, in the field of computer software, this alternate rationality is GNU/Linux.

Conclusions

There are many ways of looking at computer software. We can take Andrew Feenberg's point of view, or adopt Sherry Turkle's language, in which the computer becomes a tool in the shaping of the self. Regardless of this, if we look at our own lives as we live them, we realize that a growing portion of our work and our interaction with others is being conducted through computers, and that the importance of software is increasing. The choice for individual software users then becomes whether they want their tools of communication and production of information to be designed and controlled by for instance the Microsoft Corporation, or designed by the GNU/Linux community, and controlled by the users themselves.

On the other hand, one could argue that the importance lies in *the fact that we communicate* with others through our computers, not the technicalities of programming making this possible. Whether I use four different instant messaging programs from commercial actors, each linked to a specific network, or one free, connecting to all, does not matter nearly as much as the fact that my friends on the other end remain the same, and much more accessible over vast distances than they were a mere decade ago. I will not argue against this objection here, as I think it might very well be a valid claim, but merely point out that even with the metapsychological aspects bracketed, the legal and economic implications remains.

In this paper I hope to have shown that the phenomenon of collaborative effort in computer programming has sprung out of a deep ethical and political commitment to the responsibility to others

27 In this argument, Feenberg relies heavily on the work done in the field of science studies, mainly by Trevor Pinch, Wiebe Nijker, and Bruno Latour. It would reach far beyond the scope of this paper to treat these sources extensively, but the impact is substantial, and deserves to be mentioned.

28 Whether this is in a rejection or confirmation of Heidegger's idea of technology's essence as unessential is unclear to me.

29 The overarching objective in this work is to show how a socialistic way of organizing society has not become philosophically impossible with the growth of technological influence. Large parts of it, as Feenberg admits, are utopian speculations on how society might be structured around socialistic values, if the necessary technological choices are made. I will not take a standpoint regarding these aspects of the work, but focus on the descriptions of computers as they function in society today.

in the form of sharing, helping and cooperation towards common goals. Clearly, that this is the foundation of free software in no way implies that this is the basis for all forms of peer production, or that it will remain this way in the foreseeable future. However, it is my opinion that we can see the selflessness and community commitment exhibited on forums for GNU/Linux in general, and on the Ubuntu forums in particular, as a phenomenon genuinely based on generosity in a world where such signs have become precious. I believe that the exposure to such phenomena may have a positive effect on the worldview of those experiencing the hospitality and friendliness that meet new forum users struggling with software difficulties, and that receiving help in solving these problems might indeed induce feelings of responsibility to others on its own. It is therefore my opinion that any effort to increase the market share of GNU/Linux could be a small step in the direction of a more ethical society, without posing a threat to the possibility of profit and production of wealth.

Bibliography

Benkler, Yochai: "Coase's penguin, or Linux and 'The Nature of the Firm'" in
The Yale Law Journal company, Inc 2002: The Yale Law Journal Vol. 112, No. 3 (Dec. 2002)

Feenberg, Andrew: "From Essentialism to Constructivism: Philosophy of Technology at the
Crossroads" from <http://www-rohan.sdsu.edu/faculty/feenberg/talk4.html> (Printed April 21, 2007)

Feenberg, Andrew: "Postindustrial Discourses" in
Feenberg, Andrew: *Transforming Technology*: New York 2002: Oxford University Press [pp 91 –
114]

Gay, Joshua (ed): *Free Software, Free Society: Selected Essays of Richard M. Stallman* Boston,
Massachusetts 2002: GNU Press

Heidegger, Martin: "The Question Concerning Technology" in
Krell, David Farrell (ed.): *Basic Writings*: San Francisco 1993, Harper Collins Publishers

Ratto, Matt: "Don't fear the penguins: negotiating the trans-local space of Linux development" in
University of Chicago Press 2005: *Current Anthropology* Dec 2005 v46 i5

Tapscott, Dan, and Williams, Anthony D: *Wikinomics – How Mass Collaboration Changes Everything*
New York 2007: Penguin Books Ltd.

Turkle, Sherry: *Life on the Screen – Identity in the Age of the Internet* New York 1995: Simon &
Schuster Paperbacks

de Zengotita, Thomas: *Mediated – How the Media Shapes Your World and the Way You Live In It* New
York 2005: Bloomsbury