Open Source Software Production: Climbing on the Shoulders of Giants

by
Margit Osterloh*
University of Zurich
Institute for Research in Business Administration
Plattenstrasse 14
CH-8032 Zurich
osterloh@ifbf.unizh.ch
Phone: +41 (0)1 6342840
Fax: +41 (0)1 6344942

Sandra Rota**
University of Zurich
Institute for Research in Business Administration
Plattenstrasse 14
CH-8032 Zurich
srota@ifbf.unizh.ch
+41 (0)1 6342939
Fax: +41 (0)1 6344942

Bernhard Kuster***
University of Zurich
Institute for Research in Business Administration
Plattenstrasse 14
CH-8032 Zurich
bernhard.kuster@ifbf.unizh.ch
+41 (0)1 6342926
Fax: +41 (0)1 6344942

---

* Margit Osterloh is Professor of Business Administration and Organization Theory at the Institute for Research in Business Administration, University of Zürich.
** Sandra Rota is a Research Assistant at the chair of Margit Osterloh at the Institute for Research in Business Administration, University of Zurich.
*** Bernhard Kuster is a Research Assistant at the chair of Margit Osterloh at the Institute for Research in Business Administration, University of Zurich.

The authors thank Bruno S. Frey, Margaret Levi, Susan Rose-Ackerman, Charles Baden Fuller, Georg von Krogh, Stefan Haefliger, Roger Luethi und Sebastian Spaeth for valuable contributions.

Open Source Software Production: Climbing on the Shoulders of Giants

Abstract:

Open source software production is a successful new production model in which a public good is voluntarily provided. We argue that by studying this new production model we gain valuable insight for organization theory beyond software production. Under specific conditions this model can be generalized, contingent on the interplay of motivational, situational, and institutional factors. It is argued that a production model building on the shoulders of predecessors and peers depends on a well balanced portfolio of intrinsic and extrinsic motivation, low costs for contributors and governance mechanisms that do not crowd out intrinsic motivation.

(95 words)

Key words: Networks, open source, intrinsic and extrinsic motivation, public goods, property rights

A dwarf standing on the shoulders of a giant may see farther than a giant himself. (Merton 1993, p. 4)

1. Introduction

For a long time, open source software was only known to insiders in the information technology industry. Recently, it has triggered attention from the public and from organization researchers. Open source software like Linux, Apache and Sendmail are examples of this highly successful production model where software is developed by a network of loosely connected programmers.

Why is open source software production appealing for organization theorists? We argue that by studying this phenomenon we can gain insights into four important questions (Kogut and Metiu 2001):

1) Why is private ownership of intellectual property rights sometimes inefficient?
2) Why do people contribute to a public good voluntarily?
3) How is distributed knowledge organized efficiently in "virtual communities of practice" which constitute new organizational forms beyond the market/hierarchy-dichotomy?
4) How can cooperation between commercial firms and non-commercial communities work efficiently?

Answering these questions sheds light on the conditions under which this new production model works beyond software development. Open source projects are the most successful representatives of this production model but by no means the only one (Barbera 1999, Benkler 2002). Other examples are the NASA Clickworkers (a project where volunteers mark and classify craters on maps of Mars), Slashdot (a site with "News for Nerds" where users

can post submissions, comment on their content and classify the comments themselves as to their helpfulness) and Project Gutenberg (peer-based distribution of books that includes volunteer scanning of hard copies and proofreading). The common characteristic of these projects is peer production which is comparable to academic work (Benkler 2002). Due to the large absence of private ownership of intellectual property rights, the contributors to these projects are able to "stand on the shoulder of giants" by reusing, revisiting and transferring the work of predecessors and peers.

Regarding the four questions mentioned above, our paper concentrates on questions one, two and four, while dealing less with the third one. We argue that these questions are closely connected with the dynamics of intrinsic and extrinsic motivation.

In the *second* section of this paper, a short overview of the characteristics of open source software is provided. We argue that this kind of production model challenges conventional economists' wisdom about the production of public goods. The *third* section distinguishes various types of actors in open source software production according to their motivation to contribute to this kind of software. The *fourth* section discusses how the interaction of these different types helps to overcome a first and second order social dilemma arising in situations of public good production. In section *five*, we analyse under what conditions the open source model can be successful in general. We isolate motivational, situational, and institutional factors. These factors make "virtual communities of practice" work without central authorities and privatisation of intellectual property rights, even when no clear group and resource boundaries exist. It is *concluded* that to understand the new production model, the dynamics between intrinsic and extrinsic motivation have to be taken into account.

The methodology approach is informed by "analytic narratives" (Bates et al. 1998) but in some regards differs. We share its basic idea that scholarly research can gain from the intensive dialogue between economic theory and case material. Our approach differs,

however, in that it is not solely committed to behavioural assumptions commonly made in economics. Rather than taking selfish motivations as exogenously given, we also take into account non-selfish preferences such as obligation-based and pro-social motivations and their dynamics.

2. What is open source software?

Linux, Apache and Sendmail are three of the most famous examples of this very successful production model. Linux as a server operating environment has gained a market share of 25% in 2001 (Deutsche Bank Research 2002). It is winning unexpected converts like government agencies (e.g. the German Parliament) and investment banks (e.g. Credit Suisse First Boston). In January 2003 the open source web server Apache was used by over 65% of active servers across all domains. It received many industry awards for excellence.[1] Sendmail routes at least 42% of mails in the Internet. The closest competitors are Windows Microsoft Exchange (18%) and Unix qmail (17%).[2] SourceForge.net, a repository of open source projects, lists more than 50.000 projects and more than 550.000 registered users.[3]

Open source software production challenges conventional economists' wisdom that innovations are better supported the more they are protected by private intellectual property rights (e.g. North 1981). These authors reason that privatization of property rights is needed to overcome what is called the "tragedy of the commons" (Hardin 1968), namely the overuse and undersupply of common resources. In open source software production, the problem of overuse does not occur since there is no rivalry in consumption. Additional users can even generate positive external network effects. Nevertheless, this cannot explain why the problem of underprovision seems to have been overcome in many cases.

[1] http://www.netcraft.co.uk/Survey/
[2] http://www.dwheeler.com/oss_fs_why.html
[3] http://sourceforge.net

In contrast to conventional economic argumentation, software production is an example of collective production in which private property rights might even cause a “tragedy of the anti-commons” (Heller 1998, Heller and Eisenberg 1998). It takes place if owners have the right to exclude others from a scarce resource while no one has an efficient right to use it, thus hindering possible contributors to “stand on the shoulders of giants”.

To explain the success of the new production model, one must take into consideration three interlinked characteristics that ensure an efficient concurrence of design and testing in open source software production. These characteristics, which make open source programs more innovative and robust than proprietary programs (Kogut and Metiu 2001), will be discussed in turn:

a) Open source software is produced under different licences that assure (partly) *public ownership* by allowing:

- to read and have access to the software’s source code as a necessary first step before one can change it,
- to make copies and to distribute those copies,
- to modify the program and distribute modified versions. In the special case of the GNU General Public License, the modified versions have to be published under the same terms as the original software (Stallman 1999).

The open source software code constitutes a public good in the classical sense. The various open source licences differ to the extent in which they allow public property to be mixed with private property rights. One of the most far reaching is the GNU General Public Licence (GPL). It forces every program that contains a free software component to be released in its entirety as free software. In contrast to the conventional copyright, this licence is called “copyleft”. It “infects” the open source software with a “virus” to enforce compliance to the

copyleft. Thus, it is ensured that any derived software will remain free software. Other licenses, like Apache's, allow programmers to make their modifications private and distribute them as proprietary products. This blending of open and proprietary source, however, is sometimes condemned as a threat to the ideals of the open source community (Stallman 1999).

b) User driven distributed knowledge production in rapid feedback cycles implements concurrence in design and testing of software modules and thus enables a very efficient new product development process. In traditional software production, the software is usually sold in a form giving no access to the source code. Customers therefore have only limited possibilities to detect mistakes ("debugging") and to improve the program. They can only give feedback to the seller about any malfunctions. In contrast, in open source software production, program innovations are disclosed to the users. A large audience tests the program, debugs it during use and gives immediate feedback. This is the reason why open source software is considered to have a lower defect density than proprietary software: "given enough eyeballs, all bugs are shallow" (Raymond 2001). The user driven rapid feedback cycles work not only with debugging but also with the production of whole modules. These are contributions to the source code which are published and reviewed by peers before they become part of the next release of the software. This new production model beyond traditional markets and hierarchies was characterized by Raymond (2001) as a "bazaar" in contrast to a "cathedral".

c) *Voluntary "virtual communities of practice"*[4] are characterized *firstly* by a strong common culture which ensures an intensive network embeddedness (Sydow 1997, Sydow and Windeler 1998). *Secondly* the members of these communities have a common situated practice which

[4] Communities of practice are groups of people informally bound together by shared expertise and interest (Brown and Duguid 1991, 1998). *Virtual* communities of practice are described by Faraj and Wasko (2001) and Tuomi (2000).

constitutes their knowing how (Orlikowski 2002) on an expert level without having face-to face-interaction. In contrast to proprietary software, the users of open source software are often more sophisticated. *Thirdly*, Benkler (2002) argues that in virtual communities of practice, transaction costs of matching talents to tasks can be reduced dramatically compared to market or hierarchical modes of organization. Individuals can judge for themselves in which tasks their talents might be put to most efficient use. This voluntary matching not only avoids information losses which are characteristic for market transactions and within firms, due to incomplete contracts. It also strengthens intrinsic motivation as a result of the autonomy of individuals. This links to the *fourth* characteristic. The community members are, to a certain extent, intrinsically motivated[5] because of their voluntary work without time pressure (Deci et al.1999). The motivation to volunteer in unpaid helping activities is higher when the external pressure is low (Stukas et al. 1999). The absence of time pressure favours creativity (Amabile et al. 2002). Not being subjected to delivery deadlines is an important characteristic of open source projects (Raymond 2001).

3. Multiple types of contributors to open source software

Why should thousands of programmers contribute freely to the provision of a public good? Two alternative explanations are discussed.

- Is it the result of the collaboration of self-interested individuals who invest in their reputation (e.g. Lerner and Tirole 2002a) or who calculate their investment in the open source community lower than their personal benefits (e.g. von Hippel 2001, von Hippel and von Krogh 2003)?

[5] See section 3 on intrinsic and extrinsic motivation.

- Is it fun, altruism or prosocial preferences which guide the contributors, as some of the leaders of the open source community claim (e.g. Kollock 1999, Raymond 2001, Stallman1999, Torvalds 1998)?

The two alternative explanations refer to a distinction between two kinds of motivation (Deci and Ryan 2000, Frey 1997, Osterloh and Frey 2000): *Extrinsic* motivation works through indirect satisfaction of needs, most importantly through monetary compensation. *Intrinsic* motivation works through immediate need satisfaction. An activity is valued for its own sake and appears to be self-sustained. The ideal incentive system for intrinsic motivation consists in the work contents itself.[6]

Intrinsic motivation has two dimensions. Following Lindenberg (2001), one can differentiate between enjoyment-based and obligation-based intrinsic motivation.

- *Enjoyment-based* intrinsic motivation is the incentive focused on by Deci and his group (Deci et al. 1999). It refers to a satisfying flow of activity (e.g. Csikszentmihalyi 1975) such as playing a game or reading a novel for pleasure.

- *Obligation-based* intrinsic motivation was introduced by Frey (1997) as a further important form of incentives. Empirical field evidence for the relevance of obligation-based rules are tax morale and environmental ethics, or organizational citizenship behavior (e.g. Organ 1988).

We will argue that in the open source community there exist a variety of ideal types of contributors with different extrinsic and intrinsic motives.[7] In this section, we will

[6] In economics, with the exception of Frey (1997) and more recently Benabou and Tirole (2002), only a few authors deal with intrinsic motivation. Examples are implicit contracts (Akerlof 1982), or norms (Kreps 1997). Milgrom and Roberts (1992: 42) admit that the assumption of solely extrinsically motivated people is an "extreme caricature", nevertheless institutions should be designed as if people were entirely selfish. But this has consequences for the crowding-out effect of intrinsic motivation, see section 5.

[7] An ideal type is according to Weber (1949) a construct which has been arrived at by the thinking accentuation of certain elements of reality. In its conceptual purity, this ideal type cannot be found empirically anywhere in reality.

distinguish five different types of contributors according to their motives. In reality, these types are overlapping (Hars and Ou 2002, Lakhani et al. 2002).

3.1. Commercial service providers

Commercial service providers make money with open source software in spite of the fact that open source software is a public good. There are different business models (for an overview see Berlecon Research 2002 or Markus et al. 2000). The most prominent example is Red Hat. This company does not actually sell the source code (which anybody can download from the internet for free). Instead, it sells support and services. In addition, it adds value by integrating autonomous open source components into a working and reliable operating system that can easily be installed by inexperienced users. Based on this business model, Red Hat succeeded to capture a market share of approximately 50% of the market for Linux (Deutsche Bank Research 2002). Other commercial firms like Hewlett Packard sell hardware, like printers, and contribute add-ons, like printer drivers, to make their products work with open source software. Finally, companies like IBM contribute to open source software by making their hardware compatible with it. These firms are absolutely vital for the widespread adoption of this kind of software, because the inexperienced consumer gets reliable services and add-ons, thus helping to drive these programs into the mainstream (Kogut and Metiu 2000).

3.2. Lead users

Contributors to open source can gain non-monetary benefits as lead users who tailor the software to their own needs (von Hippel 1988, von Hippel and von Krogh 2003). They follow the saying “if you want something done right, do it yourself” (von Hippel 2001). Lead users have sufficient incentives to contribute to an innovation when they expect the personal benefits to exceed their costs.

Why should the benefits of publishing one's improvements on the internet exceed the costs of revealing information? It is argued that, *firstly*, publication opens up the possibility that other users might work with the amendments of the code, maintain and develop them. That includes the elimination of possible errors (e.g. von Hippel 2001, Lerner and Tirole 2002a). *Secondly*, the internet makes it possible for a software developer to access a wide audience with very low costs. Because publication costs are small, publication on the internet can pay off even if the expectations for helpful comments from other users are relatively low. Besides, the gains the developer reaps from the newly developed functionalities are not diminished by additional users.

These arguments may possibly hold for bug fixing, but not for contributions to newly developed functionalities. Even though participants are free to contribute as long as they release the modified code, their proposal has to pass quality and compatibility controls by peers to be accepted (von Krogh et al. 2002, Markus et al. 2000). For sophisticated software components which imply a high effort, the risk of a low pay-off might be quite high. As a consequence, some contributors to open source software may act as extrinsically motivated lead users but by far not everyone.

3.3. Reputation investors

Contributors can make money *indirectly* by signalling their ability in the open source community which can then be turned into money through employment by a commercial software company or through easier access to venture capital. Employers or venture capitalists can take the reputation of a programmer as a signal for his/her abilities which would otherwise be hard to identify. It is argued that in open source projects reputation can be more easily made visible than in proprietary projects due to the system of files that list people who made contributions, and due to the public nature of mailing list archives (Lerner and Tirole 2002a, Moon and Sproull 2000). This system makes open source production

comparable to the production of research in an academic community where reputation is made visible through citations. As in the academic community, strong norms exist regarding the public validation of innovative results.

While reputation and signalling incentives for monetary benefits indeed might be a motivation for some programmers, empirical evidence suggests that such benefits are likely to be slim (e.g. Lakhani and von Hippel 2003). “Occasionally the reputation one gains in the hacker culture can spill over into the real world in economically significant ways. It can get you a better job offer, or a consulting contract, or a book deal. This kind of side effect, however, is at best rare and marginal for most hackers; far too much so to make it convincing as a sole explanation [..]” (Raymond 2001). *Firstly,* if all developers really competed for reputation, we would expect them to try to heighten their visibility by submitting numerous contributions. Kogut and Metiu (2001) analysed the ’Changes’ files to Apache between March 1995 and February 2000. These files list the new patches included in each new version of Apache together with their authors. This analysis shows that 82% of individuals active during this period made only one or two contributions. *Secondly*, the pioneers of open source software like Linus Torvalds could hardly have been motivated by reputation benefits because at the beginning of the open source movement nobody could guarantee a success which leads to a marketable reputation (Moon and Sproull 2000). *Thirdly*,, it must be considered that recently there was a worldwide shortage of information technology specialists. The engagement in open source projects generates high opportunity costs. Working on a proprietary software as an employee might have been more favourable than an investment in a reputation for monetary benefits.

3.4. Fun seekers

While commercial service providers, lead users and reputation investors are extrinsically motivated, much evidence exists that for a lot of programmers the work itself is intrinsically

rewarding. As Raymond (2001) puts it: „We’re proving not only that we can do better software, but that joy is an asset“ (see also Brooks 1995 and Torvalds 1998). Many actors of the open source community emphasize that the most important motives for participation are fun and the public display of one’s abilities. Writing or debugging software is perceived as a „flow experience“ (Csikszentmihalyi 1975). More than 70% of open source developers report that they lose track of time while programming (Lakhani et al. 2002). As Ullman (1997) shows, programmers often experience a strong personal satisfaction from creating “something that works”. In that case, contributions to the open source code are not a cost but a benefit, not investment but consumption.

3.5. Members of the tribe

The open source community is often described as a gift-culture instead of an exchange culture (e.g. Raymond 2001). A gift is characterized by giving no tangible rewards but psychological benefits such as the „warm glow“ of sympathy or the satisfaction of living up to a moral commitment (Rose-Ackerman 1998). Gift-giving reveals the motivations of altruism or generalized reciprocity. Open source contributors report that they like the sense of “helping others” or “giving something back” to like-minded others (Faraj and Wasko 2001). Norms of generalized reciprocity sustain kindness as a social institution and lead people to provide help (Constant et al. 1996). These motivations are apart from transactional exchange relationships, because the receiver is often unknown to the giver. Participants report that “the person I help may never be in the position to help me, but someone else might be” (Rheingold 1993). People seem to reply to the entire group when answering an individual question (Wellman and Gulia 1999). The good of the community enters into the preferences of the individual contributor.

The belief that it is the right thing to give software away as a common good leads to the corollary that private ownership of intellectual property can be damaging.[8] The open source movement seems to be fuelled to some extent by the aim to destroy Microsoft's monopoly (e.g. Markus et al. 2000, Raymond 2001). Members of the tribe thus produce a public good of two different orders. *Firstly*, they contribute to the functionality and quality of the programs (first order public good). *Secondly*, they are engaged in keeping the source code open (second order public good[9]). This includes a heated discussion between various fragments of the open source community on what kind of licence best supports these moral concerns. While some believe that only the GNU General Public Licence guarantees that source code remains open, others feel that the viral effect of this licence actually reduces freedom.

4. Coexistence of the multiple types

In the ongoing discussion about how open source software production works, a lot of authors try to prove that either extrinsic or intrinsic motivation is prevailing. In contrast, we argue that this discussion is not fruitful. Rather it is important to realize that there is a coexistence of different types of motivation which are overlapping to a high degree. It is crucial to analyse the interaction of these different types.

For that purpose we use an idea discussed in the literature on social dilemmas, namely, that these dilemmas can be dissolved by the intrinsic motivation of a sufficient number of actors. Social dilemmas arise if the actions of self-interested individuals do not lead to socially desirable outcomes (Dawes 1980, Ostrom 1998). Their analysis has been strongly influenced

[8] Empirical evidence about the importance of this motive within the open source community is ambivalent,. According to Ghosh et al. 38% of open source developers report that their motivation to contribute to the community is their believing in that software should not be a proprietary good (Ghosh et al. 2002). In a different survey, Lakhani et al. (2002) find that 11% of open source developers are driven by the motivation to beat proprietary software. The difference might be explained by different samples and methodology.

[9] For first and second order public goods see the next section.

by Hardin's (1968) pessimistic view of the "tragedy of the commons". In contrast to this view, not all people are extrinsically motivated and selfish. Overwhelming empirical evidence exists that many people contribute voluntarily to public goods (see e.g. Frey and Meier 2002). If the joy of working and the good of the community enter into the preferences of the actors, the social dilemma is transformed into a coordination game where more than one equilibrium exists (Sen 1974).

The social dilemma is located on different levels. *On the first level* free riding can take place with respect to the production of software itself, because open source software constitutes – as mentioned – a public good. Since nobody can be excluded from open source software, there is a problem of undersupply.[10]

On the second level, the rules of the game have to be observed and sanctioned. The worst kinds of free riding on this level are not honouring the terms of the licence, using open source components in proprietary commercial products without giving anything back to the community or not citing or removing the credits of a contributor. As Raymond (2001) points out "surreptitiously filing someone's name off a project is, in cultural context, one of the ultimate crimes".

Reprimanding rule breakers in order to enforce the code of ethics is itself a public good and thus constitutes a social dilemma of a higher order: "Punishment almost invariably is costly to the punisher, while the benefits from punishment are diffusely distributed over all members. It is, in fact, a public good" (Elster 1989, p. 41). Second order social dilemmas can be solved without a central authority if a sufficient number of obligation-based intrinsically motivated people exist which are prepared to punish rule-breakers even if such punishment is

[10] In newsgroups this kind of free riding is known as lurking. This means reading ongoing discussions without contributing. Lurking is usually not really a problem as long as enough individuals are willing to contribute, because there is no rivalry in consumption. But there can exist a rivalry in attention, due to excessive crossposting and trolling. Given the huge amount of information that is transferred, it is critical that contributors respect the focus of the problem that is dealt with and therefore avoid crossposting (Kollock and Smith 1996). Trolling refers to deliberately posting messages with no other aim than to provoke other users.

costly to them. Laboratory empirical evidence for the existence of such people can be found in one-shot public good games (Camerer and Fehr 2003, Ledyard 1995). In the open source community, these sanctions take place by violently blaming individuals on the internet, called "flaming". Flaming is not simply a way of punishing rule-breakers, but also has an expressive function in assuring users that others are doing their part in using the public good wisely (Kollock and Smith 1996). Other sanctions are the public announcement of 'kill-filing' (stating that one doesn't want to receive mails from a specific person) or shunning (deliberately refusing to respond).

Monitoring the behavior of participants is often easy in the open source community because the internet gives full transparency[11]. Sanctioning, however, is more of a challenge. *Firstly*, many sanctions (like flaming) are informal in nature. *Secondly*, the community members are often anonymous and no clear group and resource boundaries exist. Insofar the conditions in open source communities are different from the communities Ostrom (1990) has analysed. She argues that only if clearly defined group and resource boundaries exist, self governance of the commons can be successful. Nevertheless in the open source community self governance works. It is reported that sanctions have a significant effect on behaviour (Kollock and Smith 1996) though these sanctions often do no actual harm but only induce shame. In these cases, one has to assume that not only the sanctioner, but also the sanctioned person must be intrinsically committed to obligation-based rules. Purely extrinsically motivated egoists would not feel any shame (Elster 1999, Orr 2001). Therefore it can be concluded that to solve the first and second order social dilemma in open source software production intrinsically motivated contributors are needed.

[11] With the exception of illegally including open source code in proprietary programs.

If the existence (though to a different extent) of enjoyment based and obligation based intrinsic motivation is a precondition for the complementary interaction of the different types of contributors to open source software, the question arises, under which conditions the required amount of intrinsic motivation can exist and be maintained.

5 . Complementarity of multiple types of contributors

Open source projects are not the only virtual communities which produce knowledge intensive services with partly public ownership of intellectual property rights and without being governed by central authorities (Benkler 2002). But despite the success of this new model, the majority of innovation projects still work in the traditional way.

In the following section, we will analyse under which conditions the model of virtual communities of practice works and why the “tragedy of the commons” is not salient. We look at three kinds of factors, namely the motivational, situational and institutional factor.

5.1. The motivational factor: Portfolio of intrinsic and extrinsic motivations

In the open source community different types of motivation among the contributors exist. It is argued that these different types do not only coexist but are complementary to each other.

Without intrinsically motivated “fun seekers” and “members of the tribe” open source projects would not gain enough momentum to attract extrinsically motivated contributors (Bessen 2002, Franck and Jungwirth 2002).

- Commercial service providers would lack the basis of their business. They make money on support only if the open source software is successful.
- Lead users would have to make higher set up investments so that the costs are likely to exceed the benefits.

- At the beginning of open source software production reputation investors are not able to produce marketable signals. Employers and venture capitalists are only attracted by successful projects that have already produced a critical mass of source code.

On the other hand the success of open source software is dependent on extrinsically motivated contributors. Commercial players, lead users or reputation investors trigger a leverage effect:

- If the open source movement were solely based on intrinsic motivation, the products would not be linked the way they are to the needs of the users. A disadvantage over commercial development might result.
- Inexperienced consumers could not use this kind of software which was originally designed by and for experts.

So far, it is not known which proportion of intrinsically and extrinsically motivated people exist. But there are some preliminary empirical findings about what presently drives open source programmers and participants of newsgroups. In an empirical study with participants of a user-to-user Apache field support system Lakhani and von Hippel (2003) report generalized reciprocity as the most agreed-with statement ("I have been helped before, so I reciprocate", "I help now so I will be helped in the future"), followed by identification with the community ("I answer to promote open source software"). But these self-reports might emphasize "socially correct" answers. The same empirical study found that in one of the Usenet newsgroups 57% were free riders (asking questions only), 21% were reciprocators (asking questions and giving answers), and 22% could be classified as altruists (providing answers only). Taking into account other empirical work one may conclude that extrinsic and intrinsic motivation exert about the same influence (Ghosh et al. 2002, Hars and Ou 2002, Lakhani 2002).

5.2. The situational factor: Low cost situation

Why does production flourish in open source projects based on the intrinsic motivation of many people? Why do other fields like the pharmaceutical or bio-medical industry not apply this model? The simple answer is that this production model only works in situations in which the benefits exceed the costs. This is also true for intrinsic benefits. Even among intrinsically motivated donators, martyrs and saints are in short supply. Donators are more willing to contribute if the private opportunity costs are not too high (Rose-Ackerman 2001, p. 553). According to North (1990, p. 43) there is a downward sloping demand curve for moral concerns. The more costly it gets, the less people contribute. On the other hand, if there exists a low-cost-situation, many people contribute small bits to the public good so that the total amount of contributions rises considerably (Kirchgässner 1992, Kliemt 1986).

We distinguish *two* different aspects: The costs and benefits of actually producing the code and the costs and benefits of revealing it to the community. Even though these aspects are intermingled, we will now consider them in turn.

1. For the actual *production of the source code*, it is often argued that the potential of software for modularisation is the main reason that the kind of distributed innovation we see in open source projects can work. Modularisation makes parallel work of many contributors easy while at the same time keeping integration costs low. This is true under two conditions, namely, sequential and complementary innovation processes (Bessen and Maskin 2000) which are often found in the system product industry (Somaya 2003). These two conditions also make clear that dwarfs can see farther then the giant himself only if the costs to climb on the giants` shoulders are not too high.

   - *Sequential* means that each successive invention builds on the preceding one, in particular that there are incremental, not radical steps of innovation. This allows users

and contributors to amortize their initial investments in project specific human capital over several rounds of innovation.[12]

- Innovation is *complementary* if several inventors, by following somewhat different research lines, enhance the overall probability that a certain problem is solved or, more generally, an innovation arises. This can be explained by internal dynamic economies of scales alongside trajectories, which help to concentrate successful search to a narrow field. Thus, if the efforts of an actor further the chances of discovery sufficiently, this actor has an incentive to contribute to the process of discovery even if, in principle, she could wait until someone else makes the invention.[13] The impact of such trajectories is of greater weight during the period of exploration than during exploitation (March 1991), because in that period uncertainty is more important.

2. Software is an area where the monetary *costs of revealing innovations* compared to the benefits are often quite low. There are *two* kinds of monetary costs to be considered. *Firstly*, costs of diffusion are low. Participants simply post their contributions on the appropriate internet site. *Secondly*, the losses stemming from sharing intellectual property rights by using an open source license are often low compared to the gains from the expected feedback by other participants.

The situational factor of low costs does not mean that intrinsic motivation is dispensable. At least in the beginning of projects it holds that free riding is advantageous because one gets the public good but does not carry the cost. Thus, if nobody is intrinsically motivated, open

[12] Von Krogh et al. (2002) studied the process of joining an open source project empirically, using the example of Freenet. They show that this joining process takes quite some time.

[13] Bessen and Maskin (2000) show that, for the reasons outlined above, under conditions of sequential and complementary innovation patent protection may even reduce incentives of commercial firms to invest in R&D. Their model was empirically tested in a natural experiment. Before 1980, patent protection for software was very limited in the United States, as it still is in the EU today. A series of court decisions in the early 1980´s extended patent protection considerably. Consequently the number of issued patents increased. However, in contrast to what was expected, R&D expenditures relative to sales in relevant samples of the software and software related industries dropped significantly (Bessen and Maskin 2000).

source software production would not gain momentum. In addition, the matching of talents to the different tasks, while avoiding information losses, would be less efficient. It follows that the issue of how to enable and maintain intrinsic motivation is crucial.

5.3. The institutional factor: Initial intrinsic motivation must not be crowded out

It is hard to analyze the reasons why people develop an initial sense of fun for or a commitment to certain projects. But we know the institutional conditions under which initial intrinsic motivation is crowded out (undermined) or crowded in (strengthened) by external interventions (Frey and Osterloh 2002). Two conditions are relevant for the required institutional governance mechanisms to foster the new production model.

- External interventions crowd out intrinsic motivation if the individuals affected perceive them to be controlling. In that case self-determination and self-esteem suffer and the individuals react by shifting their "locus of causality" from inside to outside. In contrast, intrinsic motivation is crowded in, if a person's feelings of self-determination is enhanced. (Deci and Ryan 2000; for a comprehensive overview over the empirical evidence see Frey and Jegen 2001).

  In open source projects self determination is enhanced for *two* reasons. *Firstly*, contributors choose for themselves where and what they wish to contribute (Benkler 2002; for empirical evidence see von Krogh et al. 2002). *Secondly*, a variety of self governance mechanisms give contributors large possibilities to participate in collective decision making in a transparent way. Extensive experimental and field research show that civic virtues are strengthened by procedural utility (Benz and Frey 2002, Frey and Stutzer 2002, Osterloh et al. 2002) and that "organizational citizenship behavior" is strengthened by participation and procedural fairness (e.g. Organ and Ryan 1995). Though governance rules in open source projects differ to a great extent (for an

overview see Markus et al. 2000), open source contributors submit themselves voluntarily to these rules without any contract.

- Empirical evidence shows that many individuals contribute voluntarily to public goods in social dilemmas as long as some other individuals contribute also. They are conditional cooperators (Fischbacher et al. 2001, Levi 1988, Ostrom 2000). KDE (K Desktop Environment) gives an impressive example of how conditional cooperation can be undermined. KDE is a Windows-like desktop for Linux and other open source operating systems developed by the open source community. It is based on a graphical interface toolkit called Qt. Qt was developed by Trolltech, a commercial software firm. It did not comply with the requirements of open source (Stallman 1999). Even though the Linux community agreed that KDE was a technically excellent product, many members refused to endorse it because they didn't agree with the terms of the Qt licence. These members started a parallel project called GNOME that is distributed under copyleft. Finally Trolltech reluctantly relicensed their product. By now Qt is available under a copyleft license.

Two consequences follow. *Firstly,* intrinsic motivation is crowded out by free riders. Therefore institutional governance mechanisms must be set in place which hinder exploitation of voluntary donors. Open source licences, in particular "copyleft", are such institutional mechanisms (Franck and Jungwirth 2002). They impose to contributors as well as to free riders what Hansmann (1980) calls the nondistribution constraint which is characteristic for non profit organizations (Rose-Ackerman 1996): Voluntary contributions cannot be redistributed among those who have a main impact on the organization. The nondistribution constraint is a major institutional precondition for voluntary donations to organizations. It is the reason why institutions like the Red Cross or most universities are governed as non profit organizations. Also for commercial providers who are dependent on

the goodwill of the developers, it is crucial to commit credibly to the nondistribution constraint and not to appropriate the joint project in an unfair manner (Franck and Jungwirth 2002). Otherwise conditional cooperation breaks down and their business model will fail (Benkler 2002).

Red Hat has submitted itself voluntarily to constraints which maintain conditional cooperation: As mentioned, Red Hat does not sell the Linux code. Instead, it sells support, services and value added by assembling and testing a running operating system that is compatible with other operating systems carrying the same brand. After a short while, other CD-ROM distributors were advertising the same CD-ROM for a considerably lower price than Red Hat charged for its product. Even if Red Hat does not own intellectual property rights on the entire source code distributed on their CD-ROM, they do have the copyright on parts of the CD. How did Red Hat react? The somewhat astonishing answer is: not at all. The managers of Red Hat argued that the norms of the open source community precluded any claim on property rights on their product. Since Red Hat is dependent on the goodwill of the open source community, it adheres to the rules which maintain conditional cooperation.

Secondly, intrinsic motivation is crowded out if the existing rules of cooperation in a public good situation, e.g. the rules of nondistribution, are disregarded. People decrease their contribution to public goods the more others decrease it. Therefore rule breaking must be hindered. Rule breaking can be made more difficult if costs for monitoring and sanctioning are low. As mentioned, monitoring in open source software production is easy due to the publicity of the internet. Also, sanctioning by flaming or kill-filing is a low cost activity. As the case of KDE shows, this kind of pressure can be very effective. But it seems reasonable to assume that in particular anonymous defectors are not vulnerable by sanctions. In these cases, sanctioning is only insofar effective as defectors in the open source community still feel a minimum of intrinsically motivated shame. Because informal graduated sanctions have a

strong expressive function, they are very suitable not to crowd out shame by alienating people from the community (Kollock and Smith 1996, Orr 2001, Ostrom 1990).

6. Concluding remarks

Open source software production is a highly successful production model. It has gained a lot of attention from organization and network theory. But it is, by far, not a singular case but rather one example of "virtual communities of practice".

The purpose of this paper is to inquire under which conditions the new production model might work in general. We started by asking four questions.

The *first* question (why is private ownership sometimes inefficient?) refers to what we have identified as the situational factor. The private ownership of intellectual property among independent suppliers is less efficient in low-cost situations, (a) characterized by incremental and complementary innovations or (b) whenever concurrence of design and testing is crucial. In these cases, owners have a right to exclude others from a scarce resource while no one has an efficient right to use it. This condition not only holds in software production but also in other peer productions of intellectual goods (Benkler 2002). It does not hold in situations (e.g. in the pharmaceutical industry) where testing and market launching demand high investments. Further empirical research is needed to shed light on the quantitative dimension of this situational factor.

The *second* question (why do people contribute to a public good voluntarily?) refers to the motivational factor. We argue that intrinsically and extrinsically motivated contributors are complementary to each other. *Intrinsically* motivated people contribute to the first and the second order public good. The first order public good is open source software of high quality. The second order public good is necessary to maintain the conditional cooperation of the

benevolent contributors. Intrinsic motivations are twofold, enjoyment-based (fun, public display of ones abilities) and obligation based (following norms of generalized reciprocity, commitment to the idea that private property of intellectual rights is an evil). We presume that in the case of obligation-based motivation, low-cost-situations will cause many participants to contribute to the public good. Thus, situational and motivational factors are highly interlinked. *Extrinsically* motivated participants contribute to the public good in an instrumental way. Their aim is to tailor the products to their own needs, to invest in their reputation for monetary purposes or to enlarge the user base for their complementary products. Both kinds of motivation – intrinsic and extrinsic – are vital for the success of the products. Further research is needed to find out which combination of intrinsically and extrinsically motivated contributors supports the new production model best. It should be analysed whether the existing balance will be destroyed by increased control of commercial providers over the products in question.

The *third* question deals with the organization forms of distributed knowledge work. Though we do not concentrate on that question we looked upon some characteristics of the internal structures of open source projects. We found that volunteering, self-governance, participation and transparency of decision making are important in two ways. *Firstly*, they enhance self-determination and thus strengthen intrinsic motivation. *Secondly*, they enable a very efficient matching of talents to tasks due to the voluntary choice of projects one wants to join. Hence, information losses, characteristic for non-voluntary work, are avoided (Benkler 2002). Further research is needed first to compare different organization forms with respect to their productivity; second, to study whether and under what conditions proprietary developers are able to duplicate these organization forms in order to unleash the dynamics and creativity of virtual communities of practice.

The *fourth* question asked how cooperation between commercial firms and non-commercial communities can work efficiently. The answer to this question is linked to the institutional factor. Licences like „copyleft“ seem to be a good institutional solution to foster the conditional cooperation of intrinsically motivated contributors on the one hand and serve the interests of extrinsically motivated investors on the other hand (Franck and Jungwirth 2002). Nevertheless, it seems to be important that the commercial providers commit themselves to the norms of the open source community beyond the obligations of the open source licence. As a consequence, companies like Red Hat submit themselves voluntarily to constraints beyond copyleft to maintain conditional cooperation in the community. Further research is needed in two directions. *Firstly*, which different kinds of licences are favourable for different projects (e.g. Lerner Tirole 2002b)? *Secondly*, will the increased participation of commercial players in the open source community cause a change in the voluntary commitment to nondistribution constraints?

To conclude, it is fruitful to analyse the dynamic relationship between intrinsically and extrinsically motivated contributors to the public goods produced in virtual communities of practice, rather than to argue that either intrinsic or extrinsic motivation is prevalent. We are far from fully understanding the interplay between motivational, situational, and institutional factors which could explain the new production model. However, it became clear that this model is based on the endeavour of thousands of dwarfs which, on the one hand, are motivated and, on the other hand, are allowed to climb on the shoulders of their peers and predecessors.

(6889 words)

7. References

Akerlof, G. A. 1982. Labor Contracts as Partial Gift Exchange. *Quart. J. of Econom.* **97**(4) 543-569.

Amabile, T. M, C. N. Hadley, S. J. Kramer. 2002. Creativity under the gun. *Harvard Bus. Rev.* **80**(8) 52-61.

Barbera, F. R. 1999. Berkman Center rebuffed. *Harvard Law Record.* **109**(9) 1.

Bates, R. H., A. Greif, M. Levi, J.-L. Rosenthal, B. Weingast.1998. *Analytic Narratives.* Princeton University Press, Princeton, NJ.

Benabou, R., J. Tirole. 2002. *Intrinsic and Extrinsic Motivation.* Princeton University Working paper, Princeton, NJ.

Benkler, Y. 2002. Coase's Penguin, or, Linux and the Nature of the Firm. *The Yale Law J*. **112**(3).

Benz, M., B. S. Frey. 2002. *Not only what, but also how. A survey on procedural utility.* University of Zürich Working Paper, Zurich, Switzerland.

Berlecon Research 2002. *Floss final report – Part 3: Basics of open source software markets and business models.* http://www.berlecon.de/studien/floss/FLOSS_Grundlagen.pdf.

Bessen, J. 2002. What good is free software? R. W. Hahn, ed. *Government policy toward open source software*. Brookings Institution Press, Washington, DC, 12-33.

Bessen, J., E. Maskin. 2000. *Sequential innovation, patents and imitation*. MIT Economics Department Working Paper, Cambridge, MA.

Brooks, F. P. 1995. *The mythical man-month: Essays on software engineering*. Addison-Wesley, Reading, MA.

Brown, J. S., P. Duguid. 1991. Organizational Learning and Communities-of-Practice: Toward a Unified View of Working, Learning, and Innovation. *Organ. Sci.* **2**(1) 40-57.

Brown, J. S., P. Duguid. 1998. Organizing knowledge. *California Management Rev.* **40**(3) 90-111.

Brown, J. S., P. Duguid. 2001. Knowledge and organization: A social-practice perspective. *Organ. Sci*. **12**(2) 198-213.

Camerer, C. F., E. Fehr. 2003. Measuring social norms and preferences using experimental games: A guide for social scientists. J. Henrich, R. Boyd, S. Bowles, C. Camerer E. Fehr, H. Gintis, eds. *Cooperation, self interest and punishment: Experimental and ethnographic evidence from small-scale societies.* Oxford University Press, Oxford, NY, forthcoming.

Constant, D., L. Sproull, S. Kiesler. 1996. The kindness of strangers: The usefulness of electronic weak ties for technical advice. *Organ. Sci*. **7**(2) 119-135.

Csikszentmihalyi, M. 1975. *Beyond boredom and anxiety.* Jossey-Bass, San Francisco, CA.

Dawes, R. M. 1980. Social dilemmas. *Annual Rev. of Psych.* **31** 169-193.

Deci, E. L., R. M. Ryan. 2000. The "What" and "Why" of Goal Pursuits: Human Needs and the Self-Determination of Behavior. *Psych. Inq*. **11**(4) 227-268.

Deci, E. L., R. Koestner, R. M. Ryan. 1999. Meta-Analytic Review of Experiments: Examining the Effects of Extrinsic Rewards on Intrinsic Motivation. *Psych. Bull.* **125**(3) 627-668.

Dempsey, B. J., D. Weiss, P. Jones, J. Greenberg. 2002. Who is an open source software developer*? Comm. of the ACM*. **45**(2) 67-72.

Deutsche Bank Research. 2002. Free software, big business? Open source software tightening their grip on industry and the public sector. *E-conomics*. **32**.

Elster, J. 1989. *The cement of society: A study of social order*. Cambridge University Press, New York, NY.

Elster, J. 1999. *Alchemies of the mind: Rationality and the emotions.* Cambridge University Press, New York, NY.

Faraj, S., M. M. Wasko. 2001. *The web of knowledge: An investigation of knowledge exchange in networks of practice.* Florida State University Working Paper, Tallahassee, FL.

Fischbacher, U., S. Gächter, E. Fehr. 2001. Are people conditionally cooperative? Evidence from public good experiments. *Econom. Lett*. **71**(3) 397-404.

Franck, E., C. Jungwirth. 2002. *Reconciling investors and donators – The governance structure of open source.* University of Zurich Working Paper, Zurich, Switzerland.

Frey, B. S. 1997. *Not just for the money: An economic theory of personal motivation*. Edward Elgar, Cheltenham, UK.

Frey, B. S., R. Jegen. 2001. Motivation crowding theory: A survey of empirical evidence. *J. of Econom. Surv.* **15**(5) 589-611.

Frey, B. S., S. Meier. 2002. *Pro-social behavior, reciprocity or both?* University of Zurich Working paper, Zurich, Switzerland.

Frey, B. S., M. Osterloh. 2002. *Successful management by motivation. Balancing intrinsic and extrinsic incentives.* Springer, Berlin, Germany.

Frey, B. S., A. Stutzer. 2002. *Beyond outcomes: Measuring procedural utility.* University of Zurich Working Paper, Zurich, Switzerland.

Ghosh, R. A., R. Glott, B. Krieger, B. Robles. 2002. *Floss final report – Part 4: Survey of developers.* http://www.infonomics.nl/FLOSS/report/FLOSS_Final4.pdf

Hansmann, H. B. 1980. The role of nonprofit enterprise. *Yale Law J.* **89** 835-901.

Hardin, G. 1968. The tragedy of the commons. *Science*. **162**(3859) 1243-1248.

Hars, A., S. Ou. 2002. Working for free ? Motivations for participating in open source projects. *Internat. J. of Electron. Comm*. **6**(3) 25-39.

Heller, M. A. 1998. The tragedy of the anticommons: Property in the transition from Marx to markets. *Harvard Law Rev*. **111**(3) 622-688.

Heller, M. A., R. S. Eisenberg, 1998. Can patents deter innovation? The anticommons in biomedical research. *Science*. **280**(5364) 698-701.

Kirchgässner, G. 1992. Toward a theory of low-cost-decision. *Eur. J. of Polit. Econom.* **8**(2) 305-320.

Kliemt, H. 1986. The veil of insignificance. *Eur. J. of Polit. Econom.* **2**(3) 333-344.

Kogut, B., A. Metiu. 2000. *The emergence of E-Innovation: Insights from open source software development.* Reginald H. Jones Center Working Paper, Philadelphia, PA.

Kogut, B., A. Metiu. 2001. Open source software development and distributed innovation. *Oxford Rev. of Econom. Policy*. **17**(2) 248-264.

Kollock, P. 1999. The economics of online cooperation: Gifts and public goods in cyberspace. M. A. Smith, P. Kollock, eds., *Communities in cyberspace*. Routledge, London, UK, 220-242.

Kollock, P., M. Smith. 1996. Managing the virtual commons: Cooperation and conflict in computer communities. S. Herring, ed., *Computer-mediated communication: Linguistic, social, and cross-cultural perspectives*. John Benjamins, Amsterdam, Netherlands, 109-128.

Kreps, D. M. 1997. Intrinsic Motivation and Extrinsic Incentives. *Amer. Econom. Rev.* **87**(2) 359-364.

Lakhani, K., E. von Hippel. 2003. How open source software works: "Free" user-to-user assistance. *Res. Policy.* Forthcoming.

Lakhani, K., B. Wolf, J. Bates, C. DiBona. 2002. *The Boston Consulting Group Hacker Survey*, http://www.osdn.com/bcg/BCGHACKERSURVEY.pdf and http://www.osdn.com/bcg/BCGHACKERSURVEY-0.73.pdf.

Ledyard, J. 1995. Public Goods: A Survey of Experimental Research. A. Roth, J. Kagel, eds., *Handbook of Experimental Economics.* Princeton University Press, Princeton, NJ.

Lerner, J., J. Tirole. 2002a. Some simple economics of open source. *J. of Indust. Econom.* **50**(2) 197-234.

Lerner, J., J. Tirole. 2002b. *The Scope of Open Source Licensing*. Harvard Business School Working Paper, Boston, MA.

Levi, M. 1988. *Of rule and revenue.* University of California Press, Berkeley, CA.

Lindenberg, S. 2001. Intrinsic Motivation in a New Light. *Kyklos* **54**(2/3) 317-343.

March, J. G. 1991. Exploration and exploitation in Organizational Learning. *Organ. Sci.* **2**(1) 71-87.

Markus, M. L., B. Manville, C. E. Agres. 2000. What makes a virtual organization work? *Sloan Management Rev*. **42**(1) 13-26.

Merton, R. K. 1993. On the Shoulders of Giants. University of Chicago Press, Chicago, IL.

Milgrom, P. R., J. Roberts. 1992. *Economics, Organization and Management.* Prentice-Hall, Englewood-Cliffs, NJ.

Moon, J. Y., L. Sproull. 2000. Essence of distributed work: The case of the Linux kernel. *Firstmonday*, **5**(11).

North, D. C. 1981. *Structure and Change in Economic History.* Norton, New York, NY.

North, D. C. 1990. *Institutions, institutional change, and economic performance*. Cambridge University Press, New York, NY.

Organ, D. W. 1988. *Organizational Citizenship Behavior: The Good Soldier Syndrome*. Lexington Books, Lexington, MA.

Organ, D. W., K. Ryan. 1995. A meta-analytic review of attitudinal and dispositional predictors of organizational citizenship behavior. *Pers. Psych*. **48**(4) 776-801.

Orlikowski, W. J. 2002. Knowing in practice: Enacting a collective capability in distributed organizing. *Organ. Sci*. **13**(3) 249-273.

Orr, S. W. 2001. The economics of shame in work groups: How mutual monitoring can decrease cooperation in teams. *Kyklos*. **54**(1) 49-66.

Osterloh, M., B. S. Frey. 2000. Motivation, knowledge transfer, and organizational firms. *Organ. Sci*. **11**(5) 538-550.

Osterloh, M., J. Frost, A. Weibel. 2002. *Solving Social Dilemmas: The Dynamics of Motivation in the Theory of the Firm*. University of Zurich Working paper, Zurich, Switzerland.

Ostrom, E. 1990. *Governing the commons: The evolution of institutions for collective action.* Cambridge University Press, New York, NY.

Ostrom, E. 1998. A behavioral approach to the rational-choice theory of collective action. *Amer. Polit. Sci. Rev*. **92**(1) 1-22.

Ostrom, E. 2000. Crowding out citizenship. *Scand. Polit. Stud*. **23**(1) 3-16.

Raymond, E. S. 2001. *The cathedral and the bazaar*. O'Reilly, Sebastopol, CA.

Rheingold, H. 1993. *The virtual community: Homesteading on the electronic frontier*. Addison-Wesley, New York, NY.

Rose-Ackerman, S. 1996. Altruism, nonprofits, and economic theory. *J. of Econom. Lit*. **34**(2) 701-728.

Rose-Ackerman, S. 1998. Bribes and gifts. A. Ben-Ner, L. Putterman, eds., *Economics, Values, and Organization.* Cambridge University Press, New York, NY, 296-328.

Rose-Ackerman, S. 2001. Trust, honesty and corruption: Reflection on the state-building process. *Eur. J. of Sociology.* **42**(3) 27-71.

Sen, A. K. 1974. Choice, orderings and morality. S. Körner, ed., *Practical reason: Papers and discussions*. Blackwell, Oxford, UK.

Somaya, D. 2003. Strategic determinants of decisions not to settle patent litigation. *Strategic Management J.* **24**(1) 17-38.

Stallman, R. 1999. The GNU operating system and the Free Software Movement. C. Dibona, S. Ockman, M. Stone, eds., *Open sources: voices from the open source revolution*. O'Reilly, Sebastopol, CA, 53-70.

Stukas, A. A., M. Snyder, E. G. Clary. 1999. The effects of "mandatory volunteerism" on intentions to volunteer. *Psych. Sci*. **10**(1) 59-64.

Sydow, J. 1997. Inter-organizational relations. A. Sorge, M. Warner, eds., *The handbook of Organizational Behavior*. Thomson, London, UK, 211-225.

Sydow, J., A. Windeler. 1998. Organizing and Evaluating Interfirm Networks: A Structurationist Perspective on Network Processes and Effectiveness. *Organ. Sci*. **9**(3) 265-282.

Torvalds, L. 1998. FM interview with Linus Torvalds: What motivates free software developers. *Firstmonday*. **3**(3).

Tuomi, I. 2000. Internet, innovation, and open source: Actors in the network. *Firstmonday*. **6**(1).

Ullman, E. 1997. *Close to the machine: Technophilia and its discontents*. City Lights, New York, NY.

Varian, H. R., C. Shapiro. 1999. *Information rules – a strategic guide to the network economy*. Harvard Business School Press, Boston, MA.

Von Hippel, E. 1988. *The sources of innovation.* Oxford University Press, New York, NY.

Von Hippel, E. 2001. Innovation by user communities: Learning from open source software. *Sloan Management Rev*. **42**(4) 82-86.

Von Hippel, E., G. von Krogh. 2003. Exploring the Open Source Software Phenomenon: Issues for Organization Science. *Organ. Sci*. Forthcoming.

Von Krogh, G., S. Spaeth, K. Lakhani. 2002. *Community, Joining, and Specialization in Open Source Software Innovation: The Case of Freenet*. University of St. Gallen Working paper, St. Gallen, Switzerland.

Weber, M. 1949. Objectivity in Social Science and Social Policy. Finch H. A., E. A. Shils, eds., *The Methodology of the Social Science*. Free Press, New York, NY, 49-112.

Wellman, B., M. Gulia. 1999. Virtual communities as communities. M. A. Smith, P. Kollock, eds., *Communities in cyberspace*. Routledge, New York, NY, 167-194.