Diskussionsbeiträge
des Fachbereichs Wirtschaftswissenschaft
der Freien Universität Berlin

Nr. 2004/19

VOLKSWIRTSCHAFTLICHE REIHE

Intrinsic Motivation in Open Source Software Development

Jürgen Bitzer, Wolfram Schrettl und Philipp J. H. Schröder

ISBN 3-935058-89-6

Intrinsic Motivation in Open Source Software Development

Jürgen Bitzer
Free University Berlin

Wolfram Schrettl*
Free University Berlin

Philipp J.H. Schröder
Aarhus School of Business

September 2004

Abstract

This papers sheds light on the puzzling evidence that even though open source software (OSS) is a public good, it is developed for free by highly qualified, young and motivated individuals, and evolves at a rapid pace. We show that once OSS development is understood as the private provision of a public good, these features emerge quite naturally. We adapt a dynamic private-provision-of-public-goods model to reflect key aspects of the OSS phenomenon. In particular, instead of relying on extrinsic motives for programmers (e.g. signaling) the present model is driven by intrinsic motives of OSS programmers, such as user-programmers, play value or *homo ludens* payoff, and gift culture benefits. Such intrinsic motives feature extensively in the wider OSS literature and contribute new insights to the economic analysis.

Keywords: open source software, public goods, homo ludens, war of attrition.
JEL classification: L86, H41, L31

*Corresponding author: Wolfram Schrettl, Free University Berlin, Department of Economics, Garystr. 55, D-14195 Berlin, Germany, Tel.: +49-30-838 57037, Fax: +49-30-83852072, e-mail: schrettl@wiwiss.fu-berlin.de. The Authors thank the participants of the annual conference of the European Association for Research in Industrial Economics (EARIE) at Humboldt University Berlin for valuable comments. The usual disclaimer applies.

1 Introduction

With the success of famous open source software (OSS) products like the operating system 'Linux', the 'Apache' Web Server' or the web browser 'Mozilla', the open source movement has attracted the interest of economists; e.g. Johnson (2002), Lerner and Tirol (2002), Myatt and Wallace (2002), Mustonen (2003) and Von Krogh and von Hippel (2003). What often puzzles economists dealing with this topic is the fact that OSS – a public good[1] – is developed for free by highly educated volunteer programmers and at a speed of innovation that in some respects outpaces commercial software development. Yet, in general, economics would predict that privately provided public goods suffer from problems of under-provision, delays in supply, and inferior quality.

This puzzle appeared to have been solved – at least for economists – by the work of Lerner and Tirole (2002). In their paper, entitled 'Some Simple Economics of Open Source', the fundamental question of why someone would want to contribute, without pay, to the development of OSS is answered by appealing to the concept of signaling. They suggest that it is the signalling of a hacker's programming ability which serves as the key driving force behind the voluntary commitment of OSS programmers.

However, recent findings indicate that this answer might fall short of capturing all aspects of the OSS phenomenon. For example, empirical studies on

[1]OSS is a public good since it features non-rivalry in consumption (a characteristic embedded in most non-material goods) and non-excludability. The latter feature is ensured by distribution of OSS under the Free Software Foundation (FSF) GNU, GPL, or GNU LGPL licenses, which ensure that the source code of software can be copied, modified, distributed freely and that – most importantly – all derived work has to be under the same open source license (FSF (2004)). See separate appendix B (not intended for publication).

the motivation of OSS programmers, e.g. Hertel et al. (2003) and Lakhani and Wolf (2003), do not support the signaling hypothesis. Instead, both Hertel et al. (2003) and Lakhani and Wolf (2003) find that intrinsic motives are the most important reason for programmer's enthusiastic commitment to OSS projects.[2] Even more problematic for the signaling hypothesis is that previous empirical and theoretical research focuses exclusively on those OSS projects which are prestigious, visible and/or have survived for some time. Inevitably, such projects are prone to foster extrinsic motives (such as signaling value); furthermore, following Frey (1997, 2002), monetary incentives (i.e. extrinsic motivation) will crowd out intrinsic motivation.[3] In contrast, the majority of OSS projects have emerged from un-sensational, un-prestigious and humble software problems[4], while only a few have become widely used and thus well known. In such environments signaling can hardly be a key driving force – all the more so as there exist literally tens of thousands of unknown and therefore un-prestigious OSS projects that are developed and maintained by an army of volunteers with the same vigour and intensity as their famous counterparts. Why would an individual ever consider contributing to such a project? To illustrate, the SourceForge.net repository of OSS projects, on its own, hosts 86,873 OSS projects with 910,899 registered contributors.[5] The majority of these OSS projects are, of course, utterly unknown to the wider public. Taking into account the small expected value

[2]Note that intrinsic motives may also be an important parameter in commercial software development, see Trittmann et al. (2002).

[3]For further reading on the relation of intrinsic and extrinsic motives see the recent book by Frey and Osterloh (2002).

[4]A rare exception to this rule is software released as OSS after initially being developed as commercial software, e.g. StarOffice (OSS version: OpenOffice) and Netscape (OSS version: Mozilla).

[5]Data from September 2004. In comparison, the FSF hosts some 2,600 OSS projects.

of benefits from extrinsic motives that such projects can command, it becomes questionable if the phenomenon OSS, or at least the phenomenon of programmers starting up OSS projects, can at all be rationalised by extrinsic motives. Thus, the economics of OSS may not be that simple after all.

The present paper investigates this issue. The motives of participants of OSS start-up projects or projects with low visibility, i.e. motives that cannot be reduced to simple signaling, have to the best of our knowledge not yet been dealt with in economic analysis of the OSS phenomenon. To understand the roots of the OSS phenomenon, the motives and characteristics of *initiators* of OSS projects have to be analysed and understood. Reviewing the rapidly emerging literature on OSS, we identify three crucial themes that that regularly appear when analysing the motivation of OSS programmers and in particular initiators: (a) the need for a particular software solution, i.e. the phenomenon of user-programmers, (b) the fun to play, i.e. some form of *homo ludens* payoff, and (c) the desire to give a gift to the programmer community, i.e a gift benefit. In particular the latter two motives are greatly underrated in economics – with some notable exceptions (Frey, 1997, 2002, Lindenberg, 2001) – yet are frequently discussed in other branches of the social sciences. Once incorporated into a traditional private-provision-of-public-goods economic framework, these motives yield important new insights.[6]

The private-provision-of-public-goods model we propose for the study of the OSS phenomenon follows the tradition of Bliss and Nalebuff (1984), Hendricks et al. (1988) and Bilodeau and Slivinski (1996). The private provision

[6] Other papers that model OSS in private-provision-of-public-goods frameworks, but do rely on more traditional economic rewards, are e.g. Johnson (2002) and Myatt and Wallace (2002). Motives such as gift benefit and *homo ludens* payoff are included in the OSS accounts of e.g. Zeitlyn (2003), Torvalds and Diamond (2001), Raymond (2000a, 2000b). See also section 2.

of OSS is modelled as a continuous time n player *war of attrition*. By including the motives of user-programmers, a *homo ludens* payoff, and a gift benefit, this model allows us to determine the characteristics of the provider of the OSS. It is found that software is provided sooner rather than later. Thus, in contrast to other applications of the war of attrition (e.g. Alesina and Drazen, 1991), the present model features *no* rational delay. More importantly, as to the contributing individual: *ceteris paribus* an OSS programming individual is characterised by a higher gain from using the OSS software solution, a larger value adjoint to the gift benefit, a longer time horizon (i.e. a younger individual), more patience, higher efficiency (lower development cost), and a high *homo ludens* payoff, i.e. the value derived from playing with the software and mastering the challenge. These model results correspond well to the picture of OSS developers that has emerged from empirical studies.

The paper is structured as follows. Section 2 reviews the OSS phenomenon, provides an introduction to the literature on OSS, and relates our paper to this context. Section 3 introduces the formal model. In Section 4 we derive the equilibria and characterise the agents who develop OSS. Section 5 concludes the paper.

2 OSS: Features and Motives

There have been a number of papers dealing with the motivation of programmers and examining why they are willing to program OSS. Raymond (2000a, 2000b), Lerner and Tirole (2002), Torvalds and Diamond (2001) carried out case-study-based analysis, focussing on famous OSS projects like the Apache Web Server, Perl, and sendmail. Empirical studies based on the analysis of web archives are those of Hertel et al. (2003), who dealt with Linux, and

Lakhani and Wolf (2003), Krishnamuturthy (2002), and Hars and Ou (2002) who were concerned with a broad range of different OSS projects.

In the studies mentioned, two general groups of motives are identified: intrinsic and extrinsic motives. While intrinsic motivation describes the circumstance that somebody is doing something because it is inherently interesting, enjoyable or challenging, in the case of extrinsic motivation, someone expects a separable outcome (Deci and Ryan, 1985, Ryan and Deci 2000).

Furthermore, the empirical studies show that the 'average' OSS contributor is about 30 years old and well-educated. Hars and Ou (2002), for example, find 54 percent of the contributors in their sample to be less than 29 years of age and 72 percent have a bachelor's, master's or Ph.D. degree. Similar results are found by Hertel et al. (2003), Lakhani and Wolf (2003), Krishnamuturthy (2002).

A central aim of the present paper is to understand the economic effect of the motives of those agents who are willing to provide the *initial* public good OSS, and to characterise these agents within a formal economic framework. As argued in the introduction, the present paper goes beyond the simple economic motives associated with signaling. In particular, since programming software is associated with the risk of failure (e.g. the software is not technologically successful or the project does not become famous) extrinsic motives (signalling) are unable to explain the OSS phenomenon in its full, and can rarely be linked to the motives of initiators of OSS projects.

Hence, apart from extrinsic motives associated with signalling value, there remain three main motives that are frequently discussed as explanations for why an initiator starts an OSS project:

- need for a particular software solution
- fun/play, i.e. a *homo ludens* payoff

- gift culture, social standing

The motivation of a single programmer for starting an OSS project is of course a mixture of these reasons.

Founders usually start a project because they are not satisfied with existing software or simply because the required software does not exist. For example Linus Torvalds needed a Unix for his PC, resulting in Linux (Torvalds and Diamond, 2001), Eric Allman needed a more efficient email server resulting in 'Sendmail', Larry Wall needed a tool to automatically generate web pages resulting in 'Perl', and finally Don Knuth needed a convenient tool for type-setting documents resulting in 'TEX' (Knuth, 1979).

Thus, the programmer benefits directly from developing the software, further famous examples in that category being KDE, XFree86, The Gimp, Emacs. The need for a particular software solution can be seen as a necessary but not sufficient condition for the start of an OSS project – in particular it cannot explain the decision to make one's programming results available publicly and for free. This is where other intrinsic motives play a crucial role for an initiator starting up an OSS project. Thus, besides mentioning the need for a new software, important contributors to the development of OSS claim that they are doing the programming 'just for fun' (Torvalds and Diamond (2001)). Programming is a leisure-time activity, i.e. playing around with the possibilities of software or mastering the challenge as a pastime. This idea – that the fun of play is an important motivation for humans – is not new and can be traced back to Plato; the *locus classicus* is Johan Huizinga (1938). Huizinga's *homo ludens*, the playful human, means in our setting that the programmer receives some form of benefit simply from carrying out the programming or rather from mastering a software problem. Raymond even goes a step further and argues that the intrinsic interest in the software leads to better quality programming. He states:

> “It may well turn out that one of the most important effects of open source's success will be to teach us that play is the most economically efficient mode of creative work.”
>
> (Raymond, 2000b, chapter 11)

The fun of play thus appears to be another necessary condition for starting an OSS project, but like the motive 'need for a new software', it is not sufficient. There are thousands of programmers who program new software because they need it and have fun programming it, but these programmers decide to earn money with the final product or are not willing to publish the source codes, but rather keep them as private software solutions. Thus, what is the motive to turn one's efforts into a public good, namely publish the source code of one's programm and have it licensed under the GPL?

There are a host of candidate motives, and the literature is far from settled on the issue. Although the various explanations differ, all of them carry the common theme of 'giving one's program as a gift to the community'. We summarise such motives under the term 'gift culture' (Schmied, 1996; Berking, 1996; Zeitlyn, 2003).

One of the motives falling into this category is the desire to gain a reputation within the hacker community. The names of the 'patrons' are distributed with the source code of their piece of software, which includes a list (e.g. the update log) of all contributors to the project. In this way, the individual's acceptance within the community is boosted, and with it her reputation and social status. Thus, the 'social status' of a programmer in this community is determined by the 'gift' which she has given to it (Raymond 2000a, chapters 6-8). Another motive is what Hars and Ou (2002) call 'community identification'. By this they mean that programmers see the open source community as a family which works towards a convergence between their individual goals and the goals of the community. Members see the community as kin and are therefore willing to do something which might be beneficial to the community even if it is not for themselves. Of course, 'membership' in the OSS community entails some kind of obligation for the individual programmer to follow the rules of the community, i.e. to publish the source code of his software. Although this is not obvious, Lindenberg (2001) shows that obligations can be considered as intrinsic motives, arguing that if people act based on a principle, they do not pursue external rewards. Another branch of the obligation hypothesis can be regarded as 'reciprocal altruism' in the sense that the volunteers who invest their efforts carry a belief that other programmers investing efforts into related problems will also make the resulting solution publicly available. Finally, pure altruism must also be mentioned as an important motive to publish the source code of one's software. For example, Richard Stallman (1999) sees OSS as a social movement promoting computer users' right to use, study, copy, modify, and redistribute computer programs as part of fundamental democratic principles.[7] Publishing the source code of one's software is often based on the wish to support this movement.

It is noteworthy that such explanations of gift culture are only feasible due to the freely available information within the community of developers. This close-to-costless information is a crucial characteristic of the open source development process. Information about new or ongoing projects are compiled on websites and in news groups.

[7] Cf. www.fsf.org for more information.

3 The Model

Consider a population N of individuals j. Each individual has the ability to develop one discrete unit of OSS, which is identical for all individuals, and consumption of which is characterised by non-rivalry and non-excludability.[8] Thus, the software is a public good and, once developed, it exists forever. Time is continuous and individuals discount the future at rate r_j. Utility flows are as follows. Without the OSS, individuals have to live with a commercial proprietary software alternative and receive the utility flow v_j. From the time of introduction of the OSS, all individuals obtain the flow utility $u_j = v_j + z_j$, where $z_j \geq 0$ for all $j = 1, ..., N$ is the flow value of the OSS becoming available. Individuals can produce the software at net cost C_j (to be specified below), which is the discounted present value of the net costs to individual j performing the development, i.e. it is the actual development cost minus any gains from the gift given to the community and the play value of performing the programming (*homo ludens* payoff). Given these specifications we can state:

Lemma 1. *No individual of the group m, defined by $C_j \geq \frac{z_j}{r_j}$ for all $j = 1, ..., m$, would ever develop the OSS. The community of potential developers consists of $n = N - m$ individuals.*

According to lemma 1 the community of potential developers is characterised by $C_i < \frac{z_i}{r_i}$ for all $i = 1, ..., n$. Stated differently, lemma 1 says that those with low costs of development and those with much to gain from the OSS – or alternatively, those with a great deal of disutility from having to

[8] Even though one may think of a complete piece of software, the actual OSS programming process consists rather of contributing to the pool of OSS by programming an enhancement or a new module to an ongoing OSS project.

live with the commercial software alternative – are members of the community, i.e. potential software developers. This fits well with the observations in Section 2. Assume that within this group n (the community) all costs and benefits are common knowledge. If we model a simultaneous one-time choice from the strategy set $\{develop, do\ not\ develop\}$, then this game becomes a static game of chicken, where the winning agents free-ride and have payoff $\frac{u_i}{r_i}$ and the losing agent develops the software and has payoff $\frac{u_i}{r_i} - C_i$. If no one develops, the payoff for everyone is $\frac{v_i}{r_i}$. As usual, this kind of game features a host of pure and mixed-strategy Nash equilibria in which anyone might be the current developer of the software.[9]

By allowing individuals to postpone their decision for some time, for example to wait and see if someone else is developing the software, we can introduce important dynamics into the game. Obviously the length of time a member of the community n is willing to wait depends on her benefit from the existence of the OSS, the cost of developing the software herself, and her time preference. In the normal-form version of this game, a pure strategy is a time $t_i \in [0, \infty)$ where i will develop if no one else already has done so.

We can state the following payoffs. If the OSS is developed by individual $j \neq i$ at time t, i's payoff is:

$$F_i(t) = \frac{v_i}{r_i}\left(1 - e^{-r_i t}\right) + \frac{u_i}{r_i} e^{-r_i t} \tag{1}$$

If individual i is the actual developer of the software, she suffers a one-time development cost c_i, but receives a one-time play value p_i and a net utility flow g_i incurred for Δ periods. The term g_i denotes the gift benefit. The total

[9] If all except for individual i choose *do not develop*, then it is optimal for i to choose *develop*, since $\frac{u_i}{r_i} - C_i > \frac{v_i}{r_i}$ (lemma 1). And, if i chooses *develop* then it is optimal for the $n-1$ other agents to choose *do not develop*. This reasoning holds for any agent of the community n.

net cost for agent i of voluntarily developing the software at time t can thus be written as: $C_i(t) = c_i e^{-r_i t} - p_i e^{-r_i t} - \frac{g_i}{r_i}\left(e^{-r_i t} - e^{-r_i(t+\Delta)}\right)$. Assuming that Δ extends to infinity, then via lemma 1 we know that $u_i + g_i + r_i(p_i - c_i) > v_i$. Thus every individual in the community n would rather develop in period 0 than live without the OSS forever. Now, if individual i develops the software at time t her payoff is

$$D_i(t) = F_i(t) + \frac{g_i}{r_i}\left(e^{-r_i t} - e^{-r_i(t+\Delta)}\right) + (p_i - c_i)e^{-r_i t} \quad (2)$$

Finally, if no one ever develops the software, individual i has payoff $R_i = \frac{v_i}{r_i} = lim_{t\to\infty} F_i(t) = lim_{t\to\infty} D_i(t)$.

Any individual i such that $c_i < p_i + \frac{g_i}{r_i}$ will develop voluntarily and immediately at time $t = 0$. The condition $p_i + \frac{g_i}{r_i} > c_i$ implies $D_i(t) > F_i(t)$ for all t. Since $D_i(t)$ is monotone and falling in t, $D_i(0)$ maximises utility. Put differently, an individual with a high *homo ludens* payoff and/or a high gift benefit simply develops the software, rather than waiting for someone else to provide it.[10] If such an individual exists, the game ends at time $t = 0$, and the OSS is developed immediately, i.e. at maximum speed. The more complex game emerges under the assumption that $c_i \geq p_i + \frac{g_i}{r_i}$ for all $i = 1, ..., n$. Given this assumption and lemma 1 we have $F_i(t) > D_i(t) > R_i$ for all t and the game becomes an n player war of attrition.

[10]Implicitly we assume that several agents can develop the software, but only one unit of software is created. However, all developing agents get their *homo ludens* and gift benefit payoff.

4 Results

In the remainder of this paper we employ the assumption that $c_i \geq p_i + \frac{g_i}{r_i}$ for all $i = 1, ..., n$. One can characterise the following equilibria for this type of game.

Lemma 2. *For every individual i there is a subgame perfect equilibrium outcome in which only i will develop immediately.*

Proof (Sketch). If no one else but i develops, then i's best strategy is to develop immediately, and if i develops immediately, everyone else's best strategy is to wait.

Thus the game permits – as is usual for this type of game – many subgame perfect equilibria in which anyone might volunteer. However, the set of subgame perfect equilibria can be radically reduced once time is finite.[11] With this assumption we are able to fully characterise the individual that will actually provide the public good.

Let T_i denote individual i's finite time horizon, e.g. marking the fact that i is a finitely lived agent, or stating a point in time at which i changes into a different job (where she is unable to expend effort on open source programming), or a point in time when i's human capital is outdated. Then the altered payoffs become:

$$F_i(t) = \frac{v_i}{r_i}\left(1 - e^{-r_i t}\right) + \frac{u_i}{r_i}\left(e^{-r_i t} - e^{-r_i T_i}\right) \quad (3)$$

$$D_i(t) = F_i(t) + \frac{g_i}{r_i}\left(e^{-r_i t} - e^{-r_i T_i}\right) + (p_i - c_i)e^{-r_i t} \quad (4)$$

$$R_i = \frac{v_i}{r_i}\left(1 - e^{-r_i T_i}\right) \quad (5)$$

The effect of a finite time horizon is that the game becomes non-stationary. Thus from an agent's perspective, there is a point in time $\bar{t}$

[11] See also the results in Hendricks et al. (1988) and Bilodeau and Slivinski (1996).

where she will no longer choose to enter as a developer of OSS. Beyond that point in time, even when the software is not provided, a dominant strategy is to *not develop* at all, i.e. $D_i(t) < R_i \; \forall \; t > \bar{t}_i$. Solving for $D_i(\bar{t}) = R_i$ defines this point in time.

Lemma 3. *Individual i will not develop the OSS after time*

$$\bar{t}_i = T_i - \frac{1}{r_i} \ln \left(\frac{z_i + g_i}{z_i + g_i + r_i(p_i - c_i)} \right). \tag{6}$$

Notice that $\lim_{p_i \to c_i} \bar{t}_i = T_i$, which means that when the play value approaches the cost of development, then an individual will always want to develop the software, even in the period in which she is leaving the game.[12] Also, if $z_i + g_i < r_i(c_i - p_i)$, then time $\bar{t}$ is not defined, which is in fact the condition of lemma 1, i.e. individuals that are not members of the community n. Using lemma 3 it is possible to state:

Proposition 1. *Given a finite time horizon for every individual in the community n and assuming that for all $j = 1, ..., n : \bar{t}_j \neq \bar{t}_i \; \forall \; i = 1...n; i \neq j$, there exists a unique subgame perfect equilibrium in which the individual with the highest $\bar{t}_i$ volunteers at time $t = 0$.*

Proof. Relabelling individuals, the different $\bar{t}_i$'s can be ordered $\bar{t}_n > \bar{t}_{n-1} > ... > \bar{t}_1$. When no one has provided the software until time $t \in (\bar{t}_{n-1}, \bar{t}_n]$, agent n knows that no one else will ever develop the OSS. Since $D_n(t) > R_n(T_n) \; \forall \; t \in (0, \bar{t}_n)$ and hence also for all $t \in (\bar{t}_{n-1}, \bar{t}_n)$, agent n's subgame perfect strategy is to develop the OSS if any time $t \in (\bar{t}_{n-1}, \bar{t}_n)$ is reached. Similarly at any time $t \in (\bar{t}_{n-2}, \bar{t}_{n-1}]$ agents n and $n-1$ are the last potential

[12]Notice also that when assumption $c_i \geq p_i + \frac{g_i}{r_i} \; \forall \; i = 1, ..., n$, i.e. the war of attrition assumption, is violated, then $\frac{z_i + g_i}{z_i + g_i + r_i(p_i - c_i)} < 1$ and hence $\bar{t}_i > T_i$, i.e. these individuals gain no utility from waiting.

candidates to provide the software. But there is a time $\tilde{t} < \bar{t}_{n-1}$ and sufficiently close to $\bar{t}_{n-1}$, such that $F_{n-1}(\bar{t}_{n-1}) > D_{n-1}(\tilde{t})$, therefore $n-1$, and everyone else, will prefer to wait for n to volunteer at time $\bar{t}_{n-1}$. Hence, in any subgame perfect equilibrium, n will volunteer at some time $t \in (\tilde{t}, \bar{t}_n]$. By backwards induction, the unique subgame perfect equilibrium has n developing the OSS at $t = 0$. □

The intuition for this proposition is straightforward. If you know that you are the one with the highest benefit/cost ratio of developing the OSS, and if you know that everyone else knows this as well, then you might as well give in right away. Thus, even though we allow individuals to wait, the war of attrition with full information features *no* rational delay. Software is developed sooner rather than later. The individual actually developing the software is characterised by the highest $\bar{t}$. Proposition 1 opens the possibility of obtaining a complete characterisation of the OSS programming individual that can be matched and compared to the accounts and stylized facts of the OSS community presented in Section 2. Formally,

Proposition 2. *Ceteris paribus an individual with a*

i) higher gain from the software, z_i

ii) larger gift benefit, g_i

iii) longer time horizon, T_i *(younger)*

iv) lower discount rate, r_i *(more patient)*

v) lower cost of software development, c_i

or

vi) higher value of play, p_i

is more likely to provide the OSS.

Proposition 2 follows from proposition 1, lemma 3 and the derivatives of

$\bar{t}_i$.[13] Comparing proposition 2 to the results of Section 2, we find that the characterisation of OSS programmers derived from a mildly adapted private-provision-of-a-public-good model matches surprisingly well with the popular accounts of the typical OSS programmer. By simply introducing the three central motives (need for a particular software solution, gift culture, *homo ludens*) into a private-provision-of-public-goods framework, we found that – fully in line with the evidence available – provision of OSS will be swift, and programmers will be young and efficient (talented/well-educated/low cost), have a high play value of performing the programming (*homo ludens* payoff), will benefit from the software they are producing, and value the gift culture surrounding OSS.

One of the puzzling aspects of OSS is that the privately provided public good OSS – which in principle should suffer from under-provision, delay or low quality – does exist and even pose a credible threat to commercial software producers. This can be explained in economic terms without simply appealing to differing monetary rewards as required by explanations relying on signaling, if one is prepared to open up for features which appear to play a role in the economic decisions of real people or at least of OSS programmers, but are commonly ignored in economic thinking. These features include the impact of fun and play in economic activity and gift societies. In other words one has to allow *homo ludens* to enter the playing field of *homo oeconomicus*.

[13]The derivatives of $\bar{t}_i$ are provided in appendix A (not intended for publication) at the end of the manuscript.

5 Conclusion

This paper tries to shed some light on the puzzling evidence that even though OSS is a privately provided public good, and accordingly should suffer from under-provision or low quality, it evolves quite to the contrary at a rapid pace, developed for free by highly qualified, young and motivated individuals and in fact poses a viable alternative to commercial software products. Based on a review of the phenomenon of, and the motives behind, OSS, we adapt a private-provision-of-public-goods model in the tradition of Bliss and Nalebuff (1984), Hendricks et al. (1988) and Bilodeau and Slivinski (1996) to address this puzzle and to characterise those agents who find it worthwhile to develop OSS.

The paper departs from existing economic accounts of the OSS phenomena by arguing that traditional signaling payoffs cannot satisfactorily explain the involvement of hundreds of thousands of volunteer programmers in a veritable flood of humble and utterly invisible OSS projects and activities. In particular we argue that signaling – although it can have a role in explaining the involvement of programmers in mature and famous OSS projects – rarely features among the motives of those who start up OSS projects. Instead we rely on a set of predominantly intrinsic motives that have been discussed in the wider OSS literature: (a) user programmers that actually need a particular software solution, (b) the fun of play or mastering the challenge of a given software problem, i.e. *homo ludens* payoff, and (c) the desire of belonging to the gift society of active OSS programmers. In particular the latter two motives, though widely acknowledged in social the sciences in general, are often ignored in economics, yet carry important insights for the case at hand.

Our paper incorporates these three motives into a simple dynamic private-provision-of-public-goods model. Given this set-up, the privately provided public good OSS becomes less of a puzzle. We are able to characterise the contributing individual and to determine the time of provision, generating results that compare well with empirical accounts of the OSS phenomenon. In contrast to the standard models of the private provision of public goods (e.g. Bliss and Nalebuff (1984) or Alesina and Drazen (1991)), but in line with results of Hendricks et al. (1988) and Bilodeau and Slivinski (1996), this model features *no* delay. Open source software is provided at 'maximum' speed. The individual who will actually provide the OSS is characterised as follows. *Ceteris paribus* the provider extracts a higher gain from using the software, obtains a larger gift benefit, has a longer time horizon (i.e. is a younger individual), has lower costs of development, and is equipped with a high value from play.

References

Alesina A. and A. Drazen (1991), Why Are Stabilisations Delayed?, *American Economic Review*, Vol. 81(5), pp. 1170-1188.

Berking, Helmuth (1996): *Schenken: zur Anthropologie des Gebens*, Campus: Frankfurt/Main, New York.

Bilodeau M. and A. Slivinski (1996), Toilet cleaning and department chairing: Volunteering a public service, *Journal of Public Economics*, Vol. 59, pp. 299-308.

Bliss, C. and B. Nalebuff (1984), Dragon-slaying and ballroom dancing: The private supply of a public good, *Journal of Public Economics*, Vol. 25, pp. 1-12.

Deci, E. L. and R. M. Ryan (1985): *Intrinsic Motivation and Self-Determination in Human Behavior*, Plenum Press: New York, London.

Frey, Bruno S. (1997): *Not Just for the Money: An Economic Theory of Personal Motivation*, Edward Elgar: Cheltenham, UK and Brookfield, US.

Frey, Bruno S. (2002): How Does Pay Influence Motivation?, in: Frey, B. S. and M. Osterloh (eds.) (2002): *Successful Management by Motivation: Balancing Intrinsic and Extrinsic Incentives*, Springer: Berlin, pp. 56-88.

Frey, Bruno S. and Margit Osterloh (eds.) (2002): *Successful Management by Motivation : Balancing Intrinsic and Extrinsic Incentives*, Springer: Berlin.

FSF (2004), Online Information Free Software Foundation [www.fsf.org].

Hars, A. and S. Ou (2002), Working for Free? Motivations for Participating in

Open-Source Projects, *International Journal of Electronic Commerce*, Vol. 6 (3), pp. 25-39.

Hendricks, K., A. Weiss and C. Wilson (1988), The war of attrition in continuous time with complete information, *International Economic Review*, Vol. 29, pp. 663-680.

Hertel, Guido, Sven Nieder and Stefanie Herrmann (2003): Motivation of Software Developers in Open Source Projects: An Internet-based Survey of Contributors to the Linux Kernel, in: *Research Policy*, Vol. 32(7), Special Issue: Open Source Software Development, pp. 1159-1177.

Huizinga, Johan (1938), *Homo Ludens*, Reprint 1956, Rowohlt: Hamburg.

Johnson, Justin Pappas (2002), Open Source Software: Private Provision of a Public Good, *Journal of Economics and Management Strategy*, Vol. 11(4), pp. 637-662.

Knuth, Donald E. (1979): Mathematical Typography, in *Bulletin (New Series) of the American Mathematical Society*, Vol. 1(2), pp. 337-372.

Krishnamurthy, Sandeep (2002): Cave or Community? An Emprical Examination of 100 Mature Open Source Projects, in: *First Monday*, Vol. 7, No. 6, [www.firstmonday.org].

Lakhani, Karim R. and Robert G. Wolf (2003): Why Hackers Do What They Do: Understanding Motivation Effort in Free/Open Source Software Projects, in: *MIT Sloan School of Management Working Paper*, No. 4425-03.

Lerner, J. and J. Tirole (2002), Some Simple Economics of Open Source, *Journal of Industrial Economics*, Vol. 50 (2), pp. 197-234.

Lindenberg, Siegwart (2001): Intrinsic Motivation in a New Light, in *Kyklos*, Vol. 54, pp. 317-342.

Linux Counter (2002), Online Information at [http://counter.li.org].

Mustonen, Mikko (2003), Copyleft – The Economics of Linux and other Open Source Software, *Information Economics and Policy*, Vol. 15 (1), pp. 99-121.

Myatt, David P. and Chris Wallace (2002): Equilibrium Selection and Public Good Provision: The Development of Open-Source Software, *Oxford Review of Economic Policy*, Vol 18 (4), pp. 446-461.

Raymond, Eric S. (2000a): Homesteading the Noosphere, Revision 1.22, 2000/08/24, first version 1998.

Raymond, Eric S. (2000b): The Cathedral and the Bazaar, Revision 1.51, 2000/08/24, first version 1997.

Ryan, R. M. and E. L. Deci (2000): Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions, in: *Contemporary Educational Psychology*, Vol. 25, pp. 54-67.

Schmied, Gerhard (1996): *Schenken: Über eine Form sozialen Handelns*, Leske + Budrich: Obladen.

Stallman, Richard (1999), The GNU Operating System and the Free Software Movement, in: Chris DiBona, Sam Ockman, and Mark Stone (eds.) *Open Sources: Voices from the Open Source Revolution*, O'Reilly: Sebastopol, CA.

Torvalds, Linus and Diamond, David (2001), *Just for Fun: The Story of an Accidental Revolutionary*, HarperBusiness.

Trittmann, Ralph, Dirk Stelzer, Andreas Hierholzer and Werner Mellis (2002): Managing Motivation in Software Development - A Case Study at SAP AG, in: Frey, B. S. and M. Osterloh (eds.) (2002): *Successful*

Management by Motivation: Balancing Intrinsic and Extrinsic Incentives, Springer: Berlin, pp. 251-273.

von Krogh, Georg and Eric von Hippel (2003), Special Issue on Open Source Software Development: Editorial, *Research Policy*, Vol. 32 (7), pp. 1149-1157.

Zeitlyn, David (2003): Gift economies in the deveopment of open source software: anthropological reflections, in: *Research Policy*, Vol. 32(7), Special Issue: Open Source Software Development, pp. 1287-1291.

NOT FOR PUBLICATION

A Appendix: Proposition 2 – derivatives of $\bar{t}_i$

Two useful inequalities: Lemma 1 and equation (1) and (2) combined to the inequality $u_i + g_i + r_i(p_i - c_i) > v_i$ for members of the community n. Subtracting v_i on both sides gives:

$$z_i + g_i + r_i(p_i - c_i) > 0 \quad \forall\, i \in [1, n] \tag{A.1}$$

Also recall the war of attrition assumption from Section 3, namely:

$$c_i - p_i > \frac{g_i}{r_i} \quad \forall\, i \in [1, n] \tag{A.2}$$

Own value of OSS and gift benefit: The derivatives of (6) with respect to z_i and g_i are identical:

$$\frac{\partial \bar{t}_i}{\partial z_i} = \frac{\partial \bar{t}_i}{\partial g_i} = -\frac{(z_i + g_i + r_i(p_i - c_i))\left(\frac{1}{z_i+g_i+r_i(p_i-c_i)} - \frac{z_i+g_i}{(z_i+g_i+r_i(p_i-c_i))^2}\right)}{r_i(z_i + g_i)} \tag{A.3}$$

which simplifies into

$$\frac{\partial \bar{t}_i}{\partial z_i} = \frac{\partial \bar{t}_i}{\partial g_i} = \frac{c_i - p_i}{(z_i + g_i)(z_i + g_i + r_i(p_i - c_i))} \tag{A.4}$$

By (A.2) and (A.1) both the numerator and the denominator in (A.4) are positive. Hence, $\frac{\partial \bar{t}_i}{\partial z_i} = \frac{\partial \bar{t}_i}{\partial g_i} > 0$.

Programming cost and *homo ludens* payoff: The derivatives of (6) with respect to c_i and p_i are identical in absolute value but with opposing signs:

$$\frac{\partial \bar{t}_i}{\partial c_i} = -\frac{\partial \bar{t}_i}{\partial p_i} = -\frac{1}{z_i + g_i + r_i(p_i - c_i)} \tag{A.5}$$

By (A.1) the denominator in (A.5) is positive. Hence, $\frac{\partial \bar{t}_i}{\partial c_i} < 0$ and $\frac{\partial \bar{t}_i}{\partial p_i} > 0$.

Discount rate: The derivative of (6) with respect to r_i is:

$$\frac{\partial \bar{t}_i}{\partial r_i} = \frac{p_i - c_i}{r_i(z_i + g_i + r_i(p_i - c_i))} + \frac{\log\left(\frac{z_i+g_i}{z_i+g_i+r_i(p_i-c_i)}\right)}{r^2} \tag{A.6}$$

Which is the sum of a negative and a positive term. We want to show that $\frac{\partial \bar{t}_i}{\partial r_i} < 0$. Using (A.6) we can state our problem as:

$$\log\left(\frac{z_i + g_i}{z_i + g_i + r_i(p_i - c_i)}\right) < \frac{r_i(c_i - p_i)}{z_i + g_i + r_i(p_i - c_i)} \tag{A.7}$$

Adding and subtracting $\frac{z_i+g_i}{z_i+g_i-r_i(c_i-p_i)}$ on the right-hand side (A.7) can be restated as:

$$\log\left(\frac{z_i + g_i}{z_i + g_i + r_i(p_i - c_i)}\right) < \frac{z_i + g_i}{z_i + g_i + r_i(p_i - c_i)} - 1 \tag{A.8}$$

Define $a = \frac{z_i+g_i}{z_i+g_i+r_i(p_i-c_i)}$. By (A.1) and (A.2) we have $a > 1$. Inequality (A.8) can now be stated as:

$$a - \log(a) > 1 \tag{A.9}$$

which is true for all $a \in R^+ | \, a \neq 1$. Hence, $\frac{\partial \bar{t}_i}{\partial r_i} < 0$.

NOT FOR PUBLICATION

B Why is OSS a public good?

A crucial prerequisite for our examination of the OSS phenomenon is the assumption that such software is indeed a public good. A closer look at the requirements for its classification as a public good shows that the license terms make the difference between open source and commercial proprietary software. Due to the fact that software is an immaterial good, the use of the program code by one individual does not affect its use by another individual. Therefore software is non-rival in its consumption (Houghton (1992, p.5); Quintas (1994)). Thus, the first characteristic of a public good is fulfilled by *all* software programs (Atkinson and Stiglitz (1980)). As to the second characteristic – non-excludability – it is important to note that development of OSS goes back to the late seventies, when the free exchange of software source codes was common (Stallman (1999)). Only with the emergence of proprietary software at the beginning of the eighties have commercial enterprises started to exclude users from the use of their software via copyright licenses and by distribution of compiled software, etc. Since then, the public good character of some software programs has become a distinguishing feature. It is obvious that those programmers who wanted their software to remain open to anyone interested and who wanted to prevent a commercial ‘hijacking’ of the software had to make sure that their software remained non-excludable. As a reaction to this challenge the Free Software Foundation (FSF) was founded to guarantee the free access to the software of this group by developing corresponding licenses. Different licenses, e.g. GNU GPL, GNU LGPL[14], were introduced which ensure that the source code of software can be copied, modified, distributed freely and – most importantly – that all further developments fall under the same open source license (FSF (2004)). Taking the non-rivalry of software together with the FSF software licenses, which guarantee non-excludability, OSS qualifies as a public good.

[14]GNU is a recursive acronym for ‘GNU’s not Unix’, further, GNU General Public License (GNU GPL), GNU Lesser General Public License (GNU LGPL).

Additional References:

Atkinson, Anthony B. and Joseph E. Stiglitz (1980), *Lectures on Public Economics*, McGraw-Hill: London, New York etc.

Houghton, John W. (1992), Pricing and Market Characteristics of Computer Software, *Policy Research Paper*, No. 27, Centre for International Research on Communication and Information Technologies.

Quintas, Paul (1994), Paths of Innovation in Software and Systems Development Practice, in: *The Management of Information and Communication Technologies: Emerging Patterns of Control*, R. Mansell (ed.), ASLIB, London.

Diskussionsbeiträge
des Fachbereichs Wirtschaftswissenschaft
der Freien Universität Berlin

2004

2004/1 MUCHLINSKI, Elke
Kontroversen in der internationalen Währungspolitik. Retrospektive zu Keynes-White-Boughton & IMF. 32 S.
Volkswirtschaftliche Reihe

2004/2 MUCHLINSKI, Elke
Central Banks – Transparency or Bafflement? From “monetary mystique” to “creative ambiguity”
Volkswirtschaftliche Reihe

2004/3 STRAUSZ, Roland
Regulating Availability in Bargain Sales. 17 S.
Volkswirtschaftliche Reihe

2004/4 STRAUSZ, Roland
Deterministic Mechanisms are Optimal in Standard Principal-Agent Models. 22 S.
Volkswirtschaftliche Reihe

2004/5 RENDTEL, Ulrich / Ulrich KALTENBORN,
The Stability of Simulation-based Estimation …..
Volkswirtschaftliche Reihe

2004/6 LADWIG, Birgit
Transparenz von Zentralbanken im Kontext der Debatte um die Europäische Zentralbank
Volkswirtschaftliche Reihe

2004/7 RIESE, Hajo
Arbeitsmarktpolitik und ökonomische Theorie – zur Degenerierung einer Forschungspraxis
II: Das erkenntnistheoretische Defizit
Volkswirtschaftliche Reihe

2004/8 EISEND, Martin
Metaanalyse – Einführung und kritische Diskussion
Betriebswirtschaftliche Reihe

2004/9 THADEWALD, Thorsten / Herbert BÜNING
Jarque-Bera-Test and its Competitors for Testing Normality-A Power Comparison 22 S.
Volkswirtschaftliche Reihe

2004/10 BERGER, Helge / Carsten HEFEKER
One Country, One Vote? : Labor Market Strukture and Voting Rights in the ECB 18 S.
Volkswirtschaftliche Reihe

2004/11 RIESE, Hajo
Arbeitsmarktpolitik und ökonomische Theorie- zur Degenerierung einer Forschungspraxis
III: Die markttheoretischen Aspekte des Theoriedefizits
Volkswirtschaftliche Reihe

2004/12 KRUSCHWITZ, Lutz / Andras LÖFFLER / Arnd LODOWICKS
Zur Bewertung insolvenzbedrohter Unternehmen.
Betriebswirtschaftliche Reihe

2004/13 BAR-YOSEF, Sasson / Itzhak VENEZIA
Experimental Study of Implications of SFAS 131: The Effects of the New Standard on the Informativeness of Segment Reporting. 45 S.
Betreibswirtschaftliche Reihe

2004/14 VENEZIA, Itzhak / Zur SHAPIRA
Do Professional Investors Behave Differently than Amateurs After the Weekend? 21 S.
Betriebswirtschaftliche Reihe

2004/15 STURM, Jan-Egbert / Helge BERGER / Jacob de HAAN
Which Variables Explain Decisions on IMF Credit?
An Extreme Bounds Analysis. 29 S.
Volkswirtschaftliche Reihe

2004/16 NITSCH, Volker
Zipf Zipped
Volkswirtschaftliche Reihe

2004/17 CORNEO, Giacomo
Verteilungsarithmetik der Rot-grünen Einkommensteuerreform. 13 S.
Volkswirtschaftliche Reihe

2004/18 CORNEO, Giacomo
The Rise and Likely Fall of the German Income Tax, 1958 – 2005
Volkswirtschaftliche Reihe