*Econ. Innov. New Techn.*, 2005, Vol. 14(7), October, pp. 617–635

Routledge
Taylor & Francis Group

# WHO IS NOT DEVELOPING OPEN SOURCE SOFTWARE? NON-USERS, USERS, AND DEVELOPERS

LINUS DAHLANDER* and MAUREEN McKELVEY†

*Department of Industrial Dynamics, Chalmers University of Technology, Vasa Building 2, 412 96 Gothenburg, Sweden*

*(Received 30 September 2004; Revised 15 December 2004; In final form 11 January 2005)*

The development of knowledge requires investment, which may be made in terms of financial resources or time. Open source software (OSS) has challenged much of the traditional reasoning by suggesting that individuals behave altruistically and contribute to a public good, despite the opportunity to free-ride. The lion's share of the existing literature on OSS examines communities, that is, those individuals whom are already part of the OSS community. In contrast, this paper starts from users with the requisite skill to use and develop OSS. This group of skilled individuals could potentially invest into the development of OSS knowledge, but they may or may not do so in actuality. This paper, therefore, explores three issues, which have not been extensively explored in the literature, namely, (1) how frequently a group of skilled people use OSS, (2) reasons for differences among users and non-users in terms of use and attitudes, and (3) how frequently, and why, some users contribute to OSS projects (and thereby become developers). In doing so, we consider the opportunity costs of use and development of OSS, which has been largely neglected in the literature. We find that the individuals have a rather pragmatic attitude to firms and that many are active in both firms and OSS community, which raises many questions for future research about the role and influence of firms on the development and diffusion of OSS.

*Keywords*: Opportunity cost; Motivation; Diffusion; Knowledge development; Public goods; Open source software

*JEL Classification*: O31; O33; O34

## 1 INTRODUCTION

This paper considers opportunity costs in relation to open source software (OSS) development. The development of knowledge requires investment. Nelson (1959) and Arrow (1962) opened a long running debate among economists about the public *vs.* private returns to investment in firm research and development (R&D) and in basic science. This debate has been challenged, in many ways, by the early literature on OSS, which initially presented the movement as an altruistic community. More recent OSS literature, however, asks whether and why individuals participate within a given community and has addressed the non-altruistic idea of the expected returns to the individual. Much existing literature on OSS examines communities, that is, those individuals whom are already part of the OSS community. The modern literature about the economics of science and innovation has also further nuanced our understanding of whom and why public and private R&D activities are pursued for economic reasons, in different societies.

* E-mail: linus.dahlander@mot.chalmers.se
† E-mail: maumck@mot.chalmers.se

ISSN 1043-8599 print; ISSN 1476-8364 online 
DOI: 10.1080/1043859052000344705

This literature has focused on the variety of actors involved in knowledge and innovation, such as firms, universities, public research organizations, and individuals (Fagerberg *et al.*, 2004). Each individual or organization may invest and innovate for different reasons, and their investment may come in various forms, such as financial resources, specific skills, and/or time. Given this, the question remains, why individuals and organizations which have the prerequisite skills to further develop knowledge actually do so for one particular task – or choose not to do so.

On the basis of this broader literature, the starting assumption of this paper is that the development of OSS requires individuals and/or organizations to invest into programming activities. Programming activities involve a range of things such as trying out code, debugging existing, writing new, and so on. Programming requires general skills and experience with software, also applied in order to develop specific knowledge. Who then invests in this type of knowledge development? And why? What are their opportunity costs? Where opportunity costs are defined as the highest valued alternative use of their time: 'the opportunity cost of any choice is the most valuable alternative foregone by using resources in a particular way' (Perman and Scouller, 1999, p. 41).

To address these questions, we must ask about reasons for using, developing, and diffusing OSS, as compared to opportunity costs for the individual. Therefore, this paper starts from individuals with the requisite skill to use and develop, that is, the group of skilled individuals whom could potentially invest into the development of OSS knowledge, but may or may not do so in actuality. The method in this paper is a questionnaire of individuals with at least a Master of Science in computer engineering. This paper thereby explores three issues, which have not been extensively explored in the literature, namely, (1) how frequently a group of skilled people use OSS, (2) reasons for differences among users and non-users in terms of use and attitudes, and (3) how frequently, and why, some users contribute to OSS projects (and thereby become developers). We also explore where the individuals work, and their attitude to that firms are getting increasingly interested in OSS. We are thereby able to analyze how porous the boundaries are between firms and communities in terms of whether these skilled individuals are active in both firms and the OSS community.

Our research design should be contrasted with existing studies (cf. Hertel *et al.*, 2003; Hars and Ou, 2002). The existing studies that analyze Linux, and OSS more broadly, usually start from the group of individuals whom are already involved and actively contributing and interacting. These existing studies provide important and valuable contributions to our understanding of the phenomenon, as discussed in Section 2. Moreover, the amplified interest in OSS has resulted in more people being involved in using and developing OSS software, but the fact remains that many individuals with sufficient knowledge are not involved. In fact, one could argue that there exists a massive skill base, which is not investing into software development. By this, we mean the group of persons with high level programming skills and which are also geographically dispersed but never or only occasionally contribute to OSS. If we consider the hype about OSS, their non-participation seems surprising. If we consider their choices in terms of opportunity costs related to innovation, their non-participation may be explained by varying costs and benefits in the innovation process. Our study focuses on a critical and early take-off phase of Linux, when future benefits were rather unknown. Since then, a growing number of scholars have become interested in OSS (von Hippel and von Krogh, 2003).

The paper proceeds as follows. Section 2 addresses the theoretical insights available about knowledge development, motivation, and diffusion of OSS code, to address the individual rationale for investing in the further development of OSS. Section 3 provides the methodological considerations for this study including the delimitation of the sample population and descriptive characteristics. Section 4 compares and contrasts the users and non-users, in terms

of their perceptions about OSS currently and in the future. Section 5 focuses only on the user group, especially users as related to users whom are also developers. Section 6 provides a discussion and conclusion for the paper, in terms of opportunity costs in relation to our three issues.

## 2 ACTORS, MOTIVATION, AND DIFFUSION

This section reviews the literature about OSS, to summarize the theoretical arguments in the existing literature in relation to the three issues identified earlier namely, (1) how frequently a group of skilled people use OSS, (2) reasons for differences among users and non-users in terms of use and attitudes, and (3) how frequently, and why, some users contribute to OSS projects (and thereby become developers). This can be summarized as use and development, where development can be further divided into own use and diffusion to the community.

These are fundamentally economic questions about which organizations invests in OSS, and why individuals are investing in skills and output to use, develop, and diffuse OSS. This question arises from the simple economic insight that actors have opportunity costs to the use of their resources including time. If OSS development is going to occur at all, then some actors must have incentives and resources to invest. The larger debate in economics of science and about OSS specifically can then be interpreted as one about which types of actors should be involved, costs and benefits of alternative ways of using those resources, public–private access to knowledge, and spill-overs.

The public–private investment into the development of knowledge may give rise to the tensions between public and private aims, when economic returns arise from the specific use of that knowledge within certain goods, services, and industries. OSS has increasingly been compared with open science (Dalle and David, 2003; Bezroukov, 1999) with its critical examination in peer-reviews and open diffusion of ideas. Like open science, OSS has been characterized as a public good given that it is non-excludable and non-rivalrous in consumption. Within university science, such tensions have become a major area of debate and of changing institutional structures in recent years (Nelson and Sampat, 2001).

Many interesting questions remain about the economic exploitation of this type of knowledge. Our contribution here is to focus on opportunity costs, as related to incentives and mobilization of resources, from the perspective of actors whom have skills but must choose to participate, or not. This leads naturally into the debates about whether the knowledge involved is primarily public or private, and also about whether it is 'worth' spending the time to be involved in OSS.

Our argument about opportunity costs is closely related to the dynamics of developing further software (or innovations more broadly). Our rationale is that the further development of this type of knowledge is heavily dependent upon attracting additional individuals and organizations to participate. Software development along particular trajectories can die out if the incentives change, leading to a drop in momentum in development process. Thereby, it is important to compare and contrast those who do participate from those whom do not because the decisions will affect, in turn, the momentum and direction underlying further development. The overall potential for further development of OSS will be based on decisions about whether the individual invests. It will also depend upon decisions of organizations to invest in OSS through their employees or through the development of specific products based on OSS.

The theoretical arguments reviewed provide us with rationale for the research design, as well as for outlining alternative explanations. The focus of this literature review is thus about what is known, and also to point out which phenomena have not been studied and instead require further work.

### 2.1 Actors Who Contributes to Knowledge Development in OSS

Actors mainly consist of the skill base of thousands of user and developers who choose to voluntarily contribute with their expertise and experiences in different projects. Hence, their behaviour is important. Software creation involves experimentation and a high degree of trial-and-error learning (Garzarelli, 2003) and taking part in a project requires a certain degree of knowledge about programming. Self-selection exists, as people do not enter unless they can expect themselves to be able to contribute. People who claim have the sufficient skills to take part in the OSS community, but do not have them, are revealed in peer-reviews or the like are labelled as a 'bogus' or 'wannabe'.

Two opposing views exist in the literature about the types of actors involved. On one hand, much material stresses that the OSS community is based on a particular hacker culture with collective resources (Himanen, 2001; Raymond, 2000; Levy, 1984; Bergquist and Ljungberg, 2001). Some of the codes of conduct stem from academic environments, where the open code, diffusion of ideas, peer-reviews, and dependence on mutual help were developed and have remained as important ingredients of OSS development. People involved invest their own time to improve software and are also prepared to share their ideas with others. Therefore, this model can be argued to be a collective action model for innovation (Allen, 1983), given that the contributions are brought together in a common pool resource. The collective effort of thousand users and developers all over the world contributes to a public good, given that the information that they create is non-rivalrous and non-excludable. This is the dominant view of what OSS is about.

However, another view has recently emerged of the actors involved, and these reflect a rather different economic base for OSS. In contrast to many arguments that the people within open source constitutes mainly of computer geeks outside of work-life, Lakhani and Wolf (2003) find that the majority of the respondents of developers at sourceforge.net are skilled and experienced professionals working in IT-related jobs. A recent study of the Linux kernel by Hertel *et al.* (2003) shows that 20% of the developers received a salary for their Linux programming on a regular basis and another 23% at least sometimes receive some kind of financial reward. The study also indicated that the more the developers were paid, the more time they spent developing OSS. Hars and Ou (2002) show similar results on the Linux movement at large, where 16% are directly paid for programming and their share of the total effort were even higher. Open source programmers who worked for commercial companies contributed to 38% of the total working hours reported. Both studies showed that contributions to actual development of OSS is skewed, in the sense that among the contributors, the one getting paid are more active than the others (Hertel *et al.*, 2003; Hars and Ou, 2002).

In addition to the individual, we would like to stress the role of firms in OSS development. The OSS community has long feared that commercial interests would appropriate the code and thereby use it in a closed source fashion that ultimately would destroy the common pool resource. The community, therefore, protects their work using normative sanctions to those that do not behave according to the codex and use legal mechanisms such as licensing schemes with reversed copyright schemes (copyleft) (O'Mahony, 2003). Nonetheless, more recent research is beginning to investigate the interactions between the broader OSS community and firms as a particular actor in software development. Recently, firms have shown increased interests in OSS and are involved by releasing code to the community, establishing communities, or using existing OSS code in a good or service product. Firms thereby contribute to the knowledge development in many occasions by supporting and releasing code (Dahlander and Magnusson, 2004). Hence, firms seem to be acting, on the basis of some premise about the likely future benefits (appropriability), as well as contributing to the OSS development itself.

### 2.2 Motivations to Develop and Diffuse

The debates about sources of motivation can be largely grouped into economic explanations and other explanations, on the basis of psychology, and so on. These can be broadly termed 'extrinsic motivation', which refers to incentives outside the individual and 'intrinsic motivation', which stems from within individuals and their willingness to understand and learn. On the basis an outside reward (such as money and benefits), one can support a behaviour that the individual perhaps otherwise would not have done. People may, for example, contribute with their time for a direct pay-off, or a signalling incentive about future pay-offs (Lerner and Tirole, 2002). Intrinsic rewards encourage people to certain behaviour can include, in this case, that they think that OSS development is fun (Himanen, 2001) and offers personal challenges (Raymond, 2000). Empirical studies have confirmed that both intrinsic and extrinsic motivation factors do play a role in OSS (Hertel *et al.*, 2003; Lakhani *et al.*, 2002; Hars and Ou, 2002).

An increasing number of papers have addressed the motivation for the individuals to take part in the development of this public good. The behaviour observed is thought to challenge classical reasoning within economics because people contribute with time and effort, despite the possibility to free-ride. In this theoretical perspective, private agents will only provide a public good, if the private benefits exceeds the costs of producing it. Hence, it has been argued that self interest and opportunistic behaviour cannot explain why individuals voluntarily take part in OSS for free when they have valuable knowledge. The externalities can be of many kinds where the users get some kind of indirect benefit for contributing to OSS and various arguments that have been put forward include four main themes within the literature, (1) own use, (2) signalling effects, (3) intellectual challenges, and (4) ideological sources of motivation.

Firstly, there is the issue of own use as a motivating factor. Developers and contributors can get direct benefits from OSS because they develop solutions that they need themselves (Raymond, 2000). When developing software, there might be need to develop solutions after special needs that are hard to find. By developing customizations, users get exactly what they need which satisfies heterogeneous user needs (Franke and von Hippel, 2003).

Secondly, OSS may motivate individuals through signalling effects, especially reputation and recognition, when the OSS is diffused. An influential article by Lerner and Tirole (2002) argues that people have an incentive to contribute to OSS as long as there is a positive net benefit. The net benefit is equal to the immediate payoff and a delayed payoff. The delayed reward in turn consists of two related types of incentives: career concern incentives and ego gratification. A career concern incentive refers to future job offers and financial benefits from open source based companies. These two types of incentives are grouped together into the signalling incentive. Lerner and Tirole argues that the signalling incentives are stronger in OSS than in traditional software for three main reasons: (1) better performance measures because everyone can judge whether or not a task succeeded and if it was tricky; (2) the contributor is fully responsible and not beneath another person in the hierarchy as in traditional software; (3) the labour market is typically more fluid. Holmström (1999) argues that the signalling incentive is stronger dependent on the visibility, the impact of effort on performance, and the more informative on the performance about talent. Still, the signalling effect is known to be unevenly skewed, such that only a fraction of the high-skilled developers that become gurus such as Linus Torvalds, whereas most of them are rather anonymous. In a similar vein, it should be noted that all kind of reputation effects are worthless unless you can be identified.

A related line of argument that has been put forward is that signalling matters within broader contexts. One is that students contribute to various types of OSS as a part of their training. The obvious return in this case is that the students get a degree in the long term and recognition from the teacher in the short run. The idea of OSS is that the code is freely available, so schools

and universities with limited budgets can use it free of charge. Again, these students might be inspired by the recognition to contribute and in the end better status in the exchange economy (Raymond, 2000). Individuals may gain an enhanced reputation in the eyes of peers from making high-quality contributions to an open source project. Reputation and recognition from peers is certainly an important factor, when the OSS has reached a critical mass and many individuals follow the progress within the field. Katz and Shapiro (1985, 1986) shows that in many cases, the utility which users derive from the consumption of goods increases with the number of other agents consuming it. Hence, signalling could also be argued to affect the overall momentum of development.

Thirdly, individuals may be motivated because of the intellectual challenges. An important factor that motivates people to contribute is that they enjoy the work of programming itself (Raymond, 2000). Developing new and interesting functions stimulate the persons involved. Intellectual challenges and the fun part in programming have also been emphasized elsewhere (cf. Himanen, 2001).

Finally, a motivation can be ideology and the overall culture or movement of OSS. OSS can be understood as a social movement in that a large number of people collectively solve a problem that they have in common and mobilize support (O'Mahony, 2002). In theories that explain common pool resources based on rational calculation and self-interest, a social dilemma emerge where individuals through their 'rational' actions free-ride on others' work. In OSS, this does not seem to be the case as it seems people voluntarily give away their work and contribute to a public good. Mauss (1954) stated that general reciprocity is a type of social collaboration, based on impersonal favours. It arise when people care about that the system is well-functioning and is a form of insurance where you help some in need when you expect to be helped in turn when needed. It also emerges from the perception among people that the system is a resource and a worthy institution as such (Constant *et al.*, 1996). These individuals do not expect anything in direct return in a dyadic type of relationship, but hope to achieve a well-functioning system that they can benefit from. This type of reasoning is applied to OSS, where norms have emerged over time and are fostered by the people contributing. Experimental evidence shows that individuals tend to foster the common good, despite the possible advantage of free riding (David and Holt, 1993). Individuals derive a satisfaction of contribute, rather than just maximizing their own well-being.

Most of the earlier arguments lump together development and diffusion. Indeed, diffusion is relatively easy, given internet availability, from anywhere in the world. The question of diffusion is related to source of motivation, but in this case, why people diffuse their 'invention', that is, the outcome of their individual investment into developing knowledge.

Transferring information among the network is generally done when the user knows the answer on a question. This makes knowledge sharing cheap, because retrieval and posting costs are low (Lakhani and von Hippel, 2003). Contributions are, however, not costless in terms of the time you have to spend reading, interpreting, and answering the questions posed (see also von Krogh *et al.*, 2003 on the costs of joining a project).

In considering the links between development and diffusion, the question of how innovation organizes arises. People within the community make arguments that OSS has a number of advantages over traditional software, because the movement builds upon sharing and contributing among a large number of users. Raymond (2000) uses the metaphor of 'the cathedral and the bazaar' to illustrate the difference in economic organization between software creation using the ordinary hierarchy (the cathedral) or using a network of developers (the bazaar). Raymond argues that the bazaar will be superior, because a larger number of people are more likely to detect and solve problems. In this view, OSS enables people that possess different knowledge and skills interact and solve problems. Otherwise, they could not contribute and gain access to solutions, without the intermediary of a firm.

## 3 METHODOLOGICAL CONSIDERATIONS

### 3.1 Questions from the Literature Review

The literature review suggested a number of issues requiring attention, given our interest in opportunity costs. In terms of actors, two points seem to be missing in the current literature, and thereby interesting to explore systematically in empirical material. First, there is the role of skilled and experienced professionals working in IT-related jobs. The questionnaire thereby assesses individuals with a high level of education and potentially different work experiences, whom could potentially contribute. These individuals have opportunity costs of using their time on OSS as opposed to other types of programming. Secondly, the *ad hoc* empirical evidence increasingly indicates that firms are involved, as employers of skilled IT professionals, as developers of products based on OSS, and as active participants in developing hardware and software solutions. The questionnaire, therefore, also asks questions about their attitudes towards firm involvement and future expectations.

In considering motivations, a key question is 'why not?' Hence, one issue missing in the existing literature is why some individuals never, or only occasionally, use and develop OSS. This can be examined by comparing and contrasting the actions and attitudes of non-users, users, and developers (within a population). Another shortcoming is that the current discussions mostly focus on success stories where network externalities are evident, such as the Apache web server. Small niche solution on a very specific problem might not present all the motivational aspects detailed earlier, as these arguments are often made on the larger success stories. There are cases where OSS solutions fail to develop, and Linux has run into increasing problems with IPR conflicts and firm profitability. The questionnaire, therefore, also allows respondents to answer questions about why they do not engage in OSS, as well as motivation factors among users.

Similarly, the question of diffusing results can be separated from the development of improvements. The literature about diffusion suggests that the diffusion of information in terms of bugs, codes, improvements, ways of solving problems, and so on does occur. Yet, most existing empirical evidence examines points where people engage in diffusion, and not points where people could share and contribute their 'inventions' but decide not to do so. The papers which have posed questions about 'why diffuse as opposed to why not diffuse' tend to do so from a theoretical perspective, without considering empirical evidence. The questionnaire asks questions about whether and why individuals actually develop improvements, as opposed to just using the software, as well as whether or not the individual making improvements do, indeed, diffuse them further. Given the emphasis in much of the literature about the strong codes of conduct, the questionnaire also asks for their attitude and expectations about how involvement of firms can affect the future diffusion.

### 3.2 Selection of Participants and Data Collection

This study was initiated to examine the investment into knowledge development of OSS, including the broad potential but un-tapped skill-base. The survey thereby had to include individuals who have the education and work experience, but do not necessarily use and develop OSS products. To do so, the survey was targeted to participants with a degree in Master of Science in computer engineering. Selecting individuals with a high level degree has here used as a tool to identify a population of individuals with skills, from which a sample would receive the survey. The population of actual and potential developers of OSS obviously emerges from a broader base than people with a Master of Science degree, or in other words, other individuals obviously also have these skills. Our unique contribution as compared to the

existing literature, however, is consider the behaviour of developers, users, and non-users, and to identify then, we used the criteria of completed degree as a means to identify individuals known in advance to have the skills in developing and interpreting source code. The rationale for choosing these persons was that they likely have the skills of interpreting code and errors that might occur and contribute with bug reports and new lines of code (see also McKelvey, 2001). In addition, people with a degree in computer engineering probably make decision at their workplace about which operative system to choose, what kind of computers to buy, and so on.

Two universities were picked: Linköping University in Sweden and Rice University in the USA. Both universities have been involved in the development of research and have educated computer engineers during a longer period, from which a sample could be picked.

The administration at the respective university provided either a total list of graduates or a sample of students as specified subsequently. The sample in this pilot study includes individuals divided into the year of graduation spanning from 1987 to 2001 (in total 15 years). Thereafter 10 graduates were randomly chosen in each stratum, to get a group with a presumed range of academic and work experience and different positions within the firms. The total sample was thus 300 persons (150*2) with a degree from Linkoping University or from Rice University. The surveys were sent by mail during 2001. A few years have passed since the survey, but the paper tells an intriguing story about the public/private interaction that still holds. In fact, this period is very interesting in the sense that much attention has been devoted to OSS thereafter (cf. von Hippel and von Krogh, 2003).

The questionnaires were distributed to the addresses of the respondents, as provided by the universities from their alumni directories. The questionnaire consisted of four main parts: (1) personal background information about the respondents with respect to their profession and their role within firms, (2) general perception of OSS, (3) detailed information about how they use it at work and at home, and (4) whether they made improvements and diffused those. In other words, the questionnaire included a multitude of question related to the actual use of OSS, decision making at firms, motivations to take part in the evolution, and their perception about the role of firms. The respondents were asked to mark each question with which they agreed. Many statements were paired with contradictory ones to assess the strength of agreement/disagreement. The questionnaire combined such closed questions about agreement over statements with more open questions where they could elaborate on their view of relevant issues.

The survey results treat these two groups of graduates as one sample and examine the aggregate results, with the exception of some illustrative citations from the open questions. Some survey questions were rather sensitive, such as the position of the respondents within firms, so all respondents were assured anonymity.

### 3.3 Sample Characteristics

The overall response rate was 36.7%. The response rate was acceptable from each of the universities, with some notable differences. In total, 75 questionnaires were returned from the graduates in Linköping. A number of these persons (25) answered 'No' to the control question whether they had a Master of Science degree in computer engineering from the university or specified that they did not choose to participate in the study. This left us with 50 questionnaires useful for analysis, resulting in 33.3% response rate. In the case of Rice University, fewer questionnaires were returned (62), but only two had to be excluded from the analysis, leaving us with a 40% effective response rate.

Table I outlines the sample characteristics of these respondents, in terms of received questionnaires, the number useful for analysis, and the response rate.

TABLE I Sample characteristics.

| *Sample* | *Sent* | *Received* | *Effective* | *Response rate (%)* |
|---|---|---|---|---|
| Linköping | 150 | 75 | 50 | 33.3 |
| Rice | 150 | 62 | 60 | 40 |
| All | 300 | 137 | 110 | 36.7 |

The overall response rate is acceptable as compared to the norms of business and social science research, and the responses come from a known population. We selected all respondents, and both sent and collected responses via postal mail.

Although this design is comparable to much research in business and social science, our approach needs to be compared with other surveys of OSS, especially those focusing on distributing questionnaires over the Internet. Many surveys of OSS have been conducted as Internet-based studies, where respondents consist of individuals whom voluntarily respond to a general statement at a website that people should complete a questionnaire (cf. David *et al.*, 2003; Hertel *et al.*, 2003; Lakhani *et al.*, 2002). One often referred example is sourceforge.net that reports surveys of several thousands responses. However, there are some methodological problems with this approach, despite the impressive number of responses. The sample drawn from these studies may be skewed in the sense that a special type of people may be attracted in completing them. No information is gathered about individuals who choose not to complete the survey. Several important projects are also not using sourceforge.net. Moreover, studies that use the Internet for questionnaires often have the problem with infinite populations (cf. Hertel *et al.*, 2003; Lakhani *et al.*, 2002), so that it is hard to estimate the response rate. As detailed earlier, our approach has been to identify the population through explicit criteria, select a sample, and analyze responses and response rates within that sample. Moreover, while an open internet survey is possible to conduct around our questions, the approach would likely not reach individuals not active in OSS but with relevant skills. For these reasons, we have chosen to see our study as a pilot study, in the sense that one cannot make to conclusive conclusions from this relatively small sample. Nevertheless, as detailed in the conclusions, this study highlights novel theoretical issues for further research, and these are issues likely not identified by more questionnaires of self-selected OSS developers and users.

Given our interest in non-users, users, and developers, it was necessary to first classify the sample divided into two groups, users and non-users. Non-users could include individuals whom had tested briefly Linux, but not used regularly. Hence, the survey included a question that asked them whether they had used Linux, as well as control questions, to distinguish between those who only tried and those who been users, to ensure that the groups were correctly separated. A similar design was used to distinguish only users from users who were also developers.

Table II summarizes the respondent demographics in terms of gender, start year, and end year of the education and whether they went on to take a PhD degree.

TABLE II Respondent demographics.

| *Gender* | | *Start year of MSc* | | *End year of MSc* | | *PhD degree* | |
|---|---|---|---|---|---|---|---|
| Male | 0.87 | Average | 1986 | Average | 1991 | No | 0.84 |
| Female | 0.13 | Std. | 4.44 | Std. | 4.39 | Yes | 0.16 |

*Note*: Std. = Standard deviation.

TABLE III Workplace and decisions at work about IT.

| *Work place* | | *Decisions at work* | | *Decision at work affects* | |
|---|---|---|---|---|---|
| Firm | 0.88 | No | 0.32 | Mainly self | 0.19 |
| University | 0.08 | Yes | 0.68 | Small group <10 | 0.49 |
| Unemployed | 0.04 | | | Medium group 10 < 100 | 0.21 |
| | | | | Large group >100 | 0.11 |

The vast majority of the graduates were male (87%), which is comparable to the number in the OSS community. Hars and Ou (2001), for example, find that 95% of the participants taking part in open source projects are male. On average, the respondents began their education in 1986 and ended that in 1991. All the respondents in this study had earned a Master of Science from Linköping or Rice University, which therefore is the lowest level of education reported. Among these graduates, as many as 16% have a PhD, but their PhDs could be in many related fields including computer systems, mathematics, computer science, databases, electrical engineering, and computer visioning. Given our interest in firms, we also asked where the individual worked, as well as whether the individual made IT decisions at work. Table III summarizes these results.

Table III shows that among the respondents, 88% are currently employed by a firm, whereas 4% were unemployed. The remaining 8% work in the academia. The survey also asked them to specify their industry or work tasks. Most of the respondents work within computer engineering related fields such as system development, software consulting, database applications, and electronics. Some respondents work within other industries such as aerospace, defence, genome projects, medicine, and oilfield services. This spread among industries and fields indicates that graduates within computer engineering diffuse the knowledge gained at university and later working experience to a variety of firms and industries.

This descriptive overview also shows that 68% make decisions at work, which affect investment in IT hardware or software. In 19% of the cases, the decisions affect mainly the respondent. In 49%, however, the decision affects a small group of up to 10 people. A medium group or a large group of people is affected by the decisions in 21% and 11% of cases, respectively. This question was asked to identify not only whether the respondents were working at firms but also whether they were making decisions regarding what computers to buy and what operative system to run on the computers.

## 4 COMPARING NON-USERS AND USERS

The first major classification of the results was into users and non-users, which is then used to analyze whether these two groups differ in their perception of Linux. This section, therefore, compares the two groups in terms of perceived benefits and drawbacks, involvement of firms, and future expectations about Linux development.

The first result, which was interesting and a bit surprising is the size of the non-user group. Of the 110 respondents, 50 persons claimed that they are Linux users, whereas 60 claimed that they did not use Linux. 'Use' was perceived as using Linux regularly. The percentage of users was lower than expected, given that the use of Linux in the academia is common among engineering students and given that these people have the skills required to install, adjust, and use the operative system.[1] This first result indicates that the potential group of users is actually

[1] This result may reflect the age distribution of the sample, as opposed to existing studies of current students.

much larger than those who participate and thereby opens up questions about why and how individuals with similar educational profiles make such different choices.

Each respondent had to agree and mark the specific statement in the questionnaire, to be reported here. Many statements were paired with contradictory ones. The column marked 'Fisher's test' refers to whether the differences between the two groups were significant. Note that some of the variables had a low response rate and it is therefore hard to draw any conclusions from these ones.

The second result is the set of questions about the perceived benefits and drawbacks of Linux, among all the respondents and among the two groups of 'user' and 'non-user'. Table IV outlines the perception among the users of the perceived benefits and drawbacks of Linux.

Table IV reveals a number of interesting differences about the respondents' perception of Linux depending on whether or not they used the operative system.

In general, the users were much more positive than the non-users, as could be expected. Users believed Linux was more reliable (98% *vs.* 42%) and flexible (82% *vs.* 36%) than the other group. More interestingly, few respondents of all agreed that it was unreliable or not flexible. Moreover, note that there was no significant difference between the groups on this statement. Users of Linux were also more convinced that Linux functions their purposes (41% *vs.* 5%) and that Linux could be considered as a good operative system (90% *vs.* 42%). Only one of all the respondents believed that Linux is not a well-functioning operative system. Among the respondents, some agreed that the Linux kernel is good, but the applications for it are bad (16% *vs.* 12%), but the difference was not significant.

The next result is the attitudes towards the involvement of firms in OSS. Table V summarizes these results.

Table V thus outlines the respondents' attitudes towards firms' involvement in OSS development, as well as commercial alternatives. Again, the empirical results are somewhat surprising. On one hand, only 12% agreed with the statement 'Linux is better than commercial alternatives for my purposes', despite the fact that 45% used Linux regularly. Still, the attitude towards commercial vendors differs significantly. The group that have been using Linux more often agree with the statement that they use it as an alternative to Microsoft product (31% *vs.* 17%) and that it is better than commercial alternatives (22% *vs.* 3%). On the other hand, 61% of all

TABLE IV Perceived benefits and drawbacks of Linux.

| | *All respondents (n = 110)* | | *Used Linux (n = 50)* | | *Not used Linux (n = 60)* | | *Fisher's test* |
|---|---|---|---|---|---|---|---|
| *Statement* | *n* | *%* | *n* | *%* | *n* | *%* | *Chi-square* |
| Reliable | 73 | 0.68 | 48 | 0.98 | 25 | 0.42 | *** |
| Unreliable | 2 | 0.02 | 0 | 0.00 | 2 | 0.03 | NOT |
| Flexible | 61 | 0.56 | 40 | 0.82 | 21 | 0.36 | *** |
| Not flexible | 3 | 0.03 | 2 | 0.04 | 1 | 0.02 | NOT |
| Linux functions for my purpose | 23 | 0.21 | 20 | 0.41 | 3 | 0.05 | *** |
| Linux is a good operating system | 69 | 0.64 | 44 | 0.90 | 25 | 0.42 | *** |
| The Linux kernel works well, but not the applications | 15 | 0.14 | 8 | 0.16 | 7 | 0.12 | NOT |
| Linux does not work well | 1 | 0.01 | 0 | 0.00 | 1 | 0.02 | NOT |
| It is hard to get everything to work | 30 | 0.28 | 19 | 0.39 | 11 | 0.19 | * |
| You need to be able to program to use Linux | 15 | 0.14 | 11 | 0.22 | 4 | 0.07 | * |

*Note*: $^{**}P < 1\%$, $^{**}1\% < P < 5\%$, $^{*}5\% < P < 10\%$, NOT, $10\% < P$. Fisher's test was used to calculate differences between the mean in the group of people that have used Linux and the ones who had not.

TABLE V Attitude towards firms within OSS.

| Statement | *All respondents (n = 110)* | | *Used Linux (n = 50)* | | *Not used Linux (n = 60)* | | *Fisher's test Chi-square* |
|---|---|---|---|---|---|---|---|
| | *n* | % | *n* | % | *n* | % | |
| It is worrying that firms are interested in Linux | 3 | 0.03 | 1 | 0.02 | 2 | 0.03 | NOT |
| It is a good sign that firms interested | 66 | 0.61 | 39 | 0.80 | 27 | 0.46 | *** |
| Programmer will keep making improvements even firms are interested | 60 | 0.56 | 39 | 0.80 | 21 | 0.36 | *** |
| Programmer will stop making improvements because firms are interested | 7 | 0.06 | 3 | 0.06 | 4 | 0.07 | NOT |
| Users should put improvements on the net | 28 | 0.26 | 16 | 0.33 | 12 | 0.20 | NOT |
| I want an alternative to Microsoft | 33 | 0.31 | 23 | 0.47 | 10 | 0.17 | *** |
| Linux is better than commercial alternatives for my purposes | 13 | 0.12 | 11 | 0.22 | 2 | 0.03 | *** |

*Note*: $^{***}P < 1\%$, $^{**}1\% < P < 5\%$, $^{*}5\% < P < 10\%$, NOT, $10\% < P$. Fisher's test was used to calculate differences between the mean in the group of people that have used Linux and the ones who had not.

respondents agreed with the statement 'It is a good sign that firms are interested in Linux'. Users of Linux were more positive than the other towards that large firms are interested in Linux (80% *vs.* 46%). The respondents also seem convinced that programmers will keep making improvements even though large firms are interested (80% *vs.* 36%). All in all, the respondents were rather positive towards the increased interests of firms in OSS.

The fourth result has to do with attitudes about individual recognition and expectations about the future, as summarized in Table VI.

Table VI shows the perception of recognition as a motivating factor. Linux users to a larger extent agreed that recognition is important than the others (49% *vs.* 20%). The respondents were optimistic about future developments of Linux and that it will probably work better in the future (84% *vs.* 49%).

TABLE VI Recognition and the future.

| Statement | *All respondents (n = 110)* | | *Used Linux (n = 50)* | | *Not used Linux (n = 60)* | | *Fisher's test Chi-square* |
|---|---|---|---|---|---|---|---|
| | *n* | % | *n* | % | *n* | % | |
| Recognition is important for those who make improvements | 36 | 0.33 | 24 | 0.49 | 12 | 0.20 | * |
| Recognition is unimportant for those who make improvements | 8 | 0.07 | 5 | 0.10 | 3 | 0.05 | NOT |
| Linux will work better in future | 70 | 0.65 | 41 | 0.84 | 29 | 0.49 | *** |
| Linux will work worse in future | 2 | 0.02 | 0 | 0.00 | 2 | 0.03 | NOT |

*Note*: $^{***}P < 1\%$, $^{**}1\% < P < 5\%$, $^{*}5\% < P < 10\%$, NOT, $10\% < P$. Fisher's test was used to calculate differences between the mean in the group of people that have used Linux and the ones who had not.

In summary, the empirical material based on the questionnaire led to rather surprising results. Some of the results may be due to the timing of the survey, for example, an early period of Linux. Still, within this group of, in total, 110 skilled persons with a Master of Science in computer engineering, 50 had used Linux at all. Our interest in whom will invest, and thereby whom will give momentum to the overall knowledge development, implied that our research design differs from most existing studies. By starting from a potential pool of skilled labour, people with at least a Master of Science degree in computer engineering, our results indicate that they are not necessarily attracted to using or developing OSS.

The persons surveyed have a high educational level and thereby likely have the ability to solve complex problems through their expertise. We have shown that ∼45% of the respondents claimed that they had use Linux. Put it differently, these results shows that 55% of skilled individuals are not involved and provide perspective to all studies that starts from the ones already involved (cf. Hertel *et al.*, 2003; Hars and Ou, 2002; Lakhani and Wolf, 2003). This comparison between users and non-users of OSS implies that explaining OSS cannot only be in terms of benefits to developers, such as in the Lerner and Tirole (2002) tradition. This is part of the story, but we must also ask 'why not?' Moreover, explanations can take in other aspects of attitudes, relative to expected returns. Users are, for example, more likely to be positive to the operative system in terms of reliability and flexibility.

## 5 ANALYZING USERS AND USERS/DEVELOPERS

This section analyzes only the group of users of Linux, as identified by the survey (50 out of 110). This section thereby relies on empirical material which is in some ways more comparable to existing studies of motivation of individuals. One main difference, however, is that our survey explicitly asks whether users are only users or they are users as well as developers. Those who were developers were also asked whether or not they had diffused their improvement further, thereby sharing and contributing to the movement. Table VII outlines the frequency to which the users agreed to a series of statements about whether or not they had contributed with improvements or bug-reports.

The most surprising overall result, found in Table VII, is that very few users had also made improvements. Of the 50 respondents, five had made improvements to the Linux kernel, nine had made improvements to a Linux application, and three had made improvements to

TABLE VII Improvements and diffusion.

| | *Used Linux (n = 50)* | |
|---|---|---|
| *Statement* | *Yes* | *No* |
| Have you made improvements to the Linux *kernel*? | 5 | 41 |
| Have sent comments to authors? | 1 | 2 |
| Have changes been sent over the web? | 0 | 3 |
| Have changes been included? | 0 | 3 |
| Have made improvements to a Linux *application*? | 9 | 38 |
| Have sent comments to authors? | 7 | 2 |
| Have changes been sent over the web? | 7 | 2 |
| Have changes been included? | 7 | 2 |
| Have you made improvements to a *device driver for Linux*? | 3 | 44 |
| Have sent comments to authors? | 1 | 2 |
| Have changes been sent over the web? | 1 | 2 |
| Have changes been included? | 1 | 2 |

a device driven. Of those who had made improvements, even fewer had diffused them further. Out of the 46 that answered the question to whether they had in some way contributed with improvements to the Linux *kernel*, only five agreed with a statement that they had done so. Even among this group, in only one case, the individual sent a comment with a bug-report to the responsible author. Among the 47 respondents to whether they had made improvements to a Linux *application*, nine claimed that they had done so. Among these persons, seven of them had sent comments and code contribution and got it included in a new release. In other words, improvements to a Linux application were more frequent than in the Linux kernel case. In total, three of 47 respondents had made an improvement for a *device driver* and only one of these shared it with others. In short, among the users, it varies depending on the type of software (kernel, application, device) to whether they tried to make improvements and not all among the ones that made improvements made them available to others.

After the earlier series of statements, the questionnaire had an open question. The respondents were asked to explain in their own word why they had or not improved software, and why or why not these were diffused to others. Among the 50 users, 44 answered the open question. Table VIII draws out a categorization of the open question and the frequency of the statements within each category.

A variety of explanations were given by the respondents. We have chosen a few illustrative citations, as to their explanations for not improving the software.

Lack of time was the most common reason given, indicating the high opportunity costs.

> I have not made any improvements mainly because of the time involved in setting up and coming up the learning curve. I already have a full time job doing programming, so I'd be looking for interesting small projects to begin with. I haven't yet spent time getting in touch with the developer community to learn what projects might be out there.

Independent of that you have a decent education and some of the necessary skills, it requires investment in time to be able to contribute. You have to find interesting projects that are interesting to develop. Other persons also felt the same that it was hard to combine work and be a serious contributor.

> The simple answer is that I work too hard. Neither school nor work has left me with enough surplus time to dedicate to serious Linux contributions. I have enough trouble maintaining balance in my life as it is.

Other explanations were that many solutions already exist, so that there has not been a reason to work with improving the software.

TABLE VIII Overview of the respondents who answered the open questions.

| *Category* | *Frequency* |
|---|---|
| Main reasons not to develop | |
| Lack of time | 13 |
| Not necessary | 10 |
| Not used enough | 6 |
| Other | 2 |
| Main reasons not diffusing | |
| High degree of specialization | 3 |
| Found existing solution before diffusing | 2 |
| Other | 1 |
| Main reasons to diffuse | |
| Ego | 2 |
| Useful for others | 2 |
| Other | 1 |

> There has not been a need to make improvements. I have found existing solutions to the problems I have experienced.

Even among the sub-group that worked with improvements and bug-reports, far from all contribute with their experiences to others. Some felt that customizations can be so user specific that others will not benefit from them, and therefore decided to keep them for themselves.

> I have written my own applications. They are not available on the web, since they are very specific for my custom applications.

A feeling that other users will not benefit from the improvement, make them to not diffuse the software.

> I had some problems with a graphic card and decided to make a new piece of software that dealt with that. I never made the improvements available on the web due to the high grade of specialization. I thought that not many users used that type of card.

Other respondents had made changes and were willing to share them, but at the time it was finished, someone else had already solved the problem.

> I adjusted a device driver timeout and in the next version of device driver, that change was included, before I submitted it.

Among the one that did contribute, they claimed to do so for a number of reasons, categorized as well as intrinsic and extrinsic sources of motivation.

> I wrote the software for fun and I have improved several so they would suit my purposes better. Ego made me put them on the web. Plus, if the improvements are included I don't need to remake the wheel for the next release

Another person claimed:

> I wanted to give something back to the free software community. And pointing out bugs and making improvements to software I use every day just makes sense. I point out bugs to close software vendors, provided that there seems to be any hope that my reports will be listened to. But I can't make fixes to closed-source software. As for improvements it's always nice to see your name on a piece of software, even if only in the Changelog.

In summary, the empirical work on users (only) and user/developers has provided insight that many are 'free riders'. In other words, this paper has shown that among the active users, only a small percentage 7–24% (depending on type) contributes to the wider community with their improvements to the kernel, application, or a device driver. In other words, contributions are skewed, which has also been seen in other results that show a few persons, which contribute with the vast majority of the contributions (cf. Hertel *et al.*, 2003).

Intriguingly, though, this is not just due to them not trying to improve OSS. Even among the developers that make improvements, not all diffuse their improvement to others. Therefore, they have invested resources to make the improvements, assumedly use it for their own purposes; but for various reasons, do not diffuse these further in the community. The tentative evidence, here, suggest that users only diffuse when it can benefit others. For example, users do not diffuse if the software has a too high degree of specialization. This could be thought of a somewhat different category of 'free riders' when compared with the users who do not develop or improve software at all. In some sense, they are 'free riders' in the sense that they do not contribute to the public good. Instead, they only use their improvements for themselves. However, they are certainly not 'free riders' in the sense of not caring about development of the public good. They do wish for cumulative improvements to the public good, but have specific reasons for not diffusing improvements.

However, one could turn the question around and say that many users, and also developers, are expected to be 'free riders', given the philosophy of OSS. In fact, one can assume that use is positive, if it stimulates network externalities through use. As such, those who use without

developing are necessary to continue to attract the small percentage whom will both use, develop, and diffuse their improvements. It suggests that the momentum of which contributions will be widely diffused, and which ones developed but used primarily by the person doing the programming, depends on perceived benefits to the individual and community, as well as how many persons are working on the same problem.

## 6 DISCUSSION AND CONCLUSIONS

This section considers the empirical results, in light of the theoretical understanding about public–private investment into R&D and about OSS as a phenomenon. This paper has focused on three issues, which we argued have been neglected in OSS studies, namely, (1) how frequently a group of skilled people use OSS, (2) reasons for differences among users and non-users in terms of use and attitudes, and (3) how frequently, and why, some users contribute to OSS projects (and thereby become developers). In discussing our interpretation of results, we suggest areas for future research, in which the growing community could pay more attention.

The first point is that development of OSS can be analyzed as an economics problem of public–private appropriability. This leads to questions about whom invests resources to innovate, how different ways of organizing the innovation process affects outcomes, and why there are expectations of different types of returns. To do so, one must combine existing theoretical literature about how and why actors contribute to OSS with an understanding of how opportunity costs affect software development, in a dynamic, evolutionary perspective. This paper has distinguished between use, development, and diffusion, and has considered the dynamic costs/benefits of being involved in OSS, as opposed to alternative ways of accessing the same software functionality. The results presented in previous sections can be summarized in Table IX.

The second point is to reconsider the public–private debates. Although the focus has been on OSS being like an ideal basic science, in fact, major debates have raged about the role of basic science and universities, relative to economic issues. The literature about OSS has so far neglected the tensions between public goods and appropriation of economic returns. Some examples exist, such as discussion of OSS attempts to prevent through copyleft (O'Mahony, 2003), but it appears that these tensions and tradeoffs between public goods and appropriation of economic returns are more visible than presumed so far, and they take a different form than at universities.

TABLE IX Summary of the discussion.

| | *Benefit of being involved in OSS* | *Dynamic costs and benefits expected if involved in OSS* | *How else than individual investment could they access software functionality?* |
|---|---|---|---|
| **Use** | Network externalities<br>Human capital investment | Time to learn skills/use general programming skills for other purposes | Work at a company and use there<br>Purchase ready-made OSS good-service bundle<br>Purchase alternative program |
| **Development** | Solve own problems<br>Intellectual challenges | Forego income, time/personal satisfaction | Free rider<br>Work at company and test develop there |
| **Diffusion** | Information wants to be free/public good<br>Signalling incentives<br>Ego | Signals in other areas/future jobs | Free rider |

The third point is the need to reconsider the incentives to develop OSS, in terms of both individuals and organizations. Our results strongly suggests that those individuals which have so far been studied within OSS literature represent just a fraction of all the people, which in various ways are involved in and affected by OSS. When the end of knowledge development is a public good, most traditional reasoning has suggested that people choose to free-ride. Research on OSS has focused how the OSS community overcomes that problem and what motivates people to be developers (Franke and Shah, 2003; Hertel *et al.*, 2003; Lakhani *et al.*, 2002). Our research results give some perspective to these studies, by illustrating that these developers that diffuse their ideas to other just represent a needle in a haystack. There are many 'free-riders' in any case, but perhaps they need to be there, to gain network externalities.

Moreover, as the source code is open for anyone to screen and change, OSS allows for customizations among users, and improvements may not need to be diffused to the wider community. The benefits for developers in diffusing their innovations increases with the number of people involved (Harhoff *et al.*, 2003), as they get more recognition and that other people are more likely to continue working on the same thing. This 'momentum' may come through users, as well as through the Internet, as a technological infrastructure. The technological system of the Internet, which enables globally and distributed access to knowledge, gives a much larger base to recruit new developers. The question of interest thereby changes. It is about which improvements needs to be circulated and which kept individually. It is about how 'attractive' an area is to enough users and developers to continue improving, when compared with having too few and stagnating. This is a question that goes far beyond the sociological question of community. Such a dynamic approach may help us begin to explain (and predict) when OSS may increasingly become a good and service product (commodity). We must begin to attack the economic problem, of what are the advantages of having firms, as compared to loose networks for this type of software development.

The fourth point is that individuals invest not only as individuals but also as employees. The descriptive analysis and the empirical results highlighted the importance of the individuals' context of work. The main point here is the role of firms. Both non-users and users make IT decisions, mainly in firms, but they were also very positive towards firm involvement. This result is important as the perception of individual experts may help explain the growing interest among firms in OSS. Earlier studies have shown that firm engage in OSS to get adoption among a large number of users and developers and thereby promote standards, improving the pace of technological development, and create lock-ins and first mover advantages (cf. Bonaccorsi and Rossi, 2003; Dahlander and Magnusson, 2004). We contend that individuals may provide a way to understand the potential links between the open source community and commercial firms. This is vital to understand as individuals are vital for the further development of OSS. They may be involved in OSS for other reasons than just spending free time, such as keep track of the development as part of their job. Given that many also are employed by firms, their decisions also affect a wider range of people than just the OSS community.

The final point is the need to better understand the broad framework of institutions and firm–community interactions within the investment of OSS knowledge. In contrast to the hacker culture debate (Himanen, 2001), the individuals analyzed here are very positive to the need to get firms involved and see it possible to combine firms and the OSS community. They believe that developers will keep making improvement, despite the fact that firms are involved in OSS. Generally, they also perceive it as a good sign that firms are interested in OSS, and even more surprisingly, users are even more positive that non-users. When compared with existing literature on OSS, the sample seems more influenced by pragmatism than by ideology. It may be so that studies that deal with how the OSS community protects the public good focus too much on the leadership and do not take

into account the average users. Future studies should, therefore, consider explicitly dealing with the perception of OSS among a larger group than just the 'core' of the OSS community.

## 7 DIRECTIONS FOR FUTURE RESEARCH

The results and theoretical framework developed in this paper indicate many interesting directions for future research. The first point is that the study of OSS should be widened from the current focus on core developers for considering a broader and more diverse of actors whom are affecting the development and diffusion of OSS. Current large-scale surveys need to carefully consider whether their results are skewed towards certain types of actors, as well as how to further develop surveys to explore our novel comparisons between those who use, develop, and diffuse. Our results suggest that this approach may raise crucial questions, even if this specific study has a somewhat limited generalizability from a small sample size. Still, the decisions and actions of different actors involved in use, development, and diffusion will likely help explain the differential momentum, found between OSS and commercial software, as well as within different types of OSS.

Secondly, the characteristics of the respondents, as well as the results, indicate interesting research because the boundary between the commercial firms and the sharing communities are getting increasingly porous. This paper provides some evidence that the attitudes toward firms are rather pragmatic. One could also point out that the individuals surveyed here are taking part in two worlds simultaneously. On one hand, many of the individuals are working in commercial firms with their regular duties. On the other hand, some are active in developing OSS solutions, specified here as being part of work, part of leisure time, or during both work and leisure. Studies looking at the firm perspective, mentioned earlier, have found that for-profit firms sometimes sponsor projects or devote personnel to work in existing OSS projects. Research could explore the extremely interesting questions related to what degree regular duties in firms and working in communities overlap and explore these issues among a group of people. This could potentially cast light on to be understood with the earlier point, namely, explaining more about the actors involved in OSS and their interrelations that affects its future momentum.

Thirdly, the results here indicate that OSS is not only about doing, or more specifically, that there is a gap between attitudes and action, which needs further elaboration. The results indicated that people think it is important to develop and diffuse, but it is just a fraction that actually do so. The success of OSS projects largely depends on how frequent users become developers and actively take part. Future studies may explain the mechanisms in which this take place and this line of research may revise the thinking of OSS from a 'unique phenomena' to a phenomenon similar to other types of voluntary activities such as charity and volunteering.

### *Acknowledgements*

The editor and two anonymous reviewers provided excellent feedback and suggestions for improvements on an earlier draft. The usual caveats apply.

### *References*

Allen, R.C. (1983) Collective Invention. *Journal of Economic Behaviour and Organization*, **4**, 1–24.

Arrow, K.J. (1962) Economic Welfare and the Allocation of Resources for Innovation. In Nelson, R. (Ed.) *The Rate and Direction of Inventive Activity*. Princeton, NJ: Princeton University Press.

Bergquist, M. and Ljungberg, J. (2001) The Power of Gifts: Organizing Social Relationships in Open Source Communities. *Information Systems Journal*, **11**(4), 305–320.
Bezroukov, N. (1999) Open Source Software Development as a Special Type of Academic Research (Critique of Vulgar Raymondism). *First Monday*, **4**(10).
Bonaccorsi, A. and Rossi, C. 2003. Comparing Motivations of Individual Programmers and Firms to Take Part in the Open Source Movement: From Community to Business. Working Paper. Available at: http://opensource.mit.edu/papers/bnaccorsirossimotivationlong.pdf.
Constant, D., Sproull, L. and Kiesler, S. (1996) The Kindness of Strangers: The Usefulness of Electronic Weak Ties for Technical Advice. *Organization Science*, **7**(2), 119–135.
Dahlander, L. and Magnusson, M. (2004) Relationships Between Open Source Software Companies and Communities: Observations from Nordic Firms. Working Paper, Chalmers University of Technology.
Dalle, J-M. and David, P.A. (2003) The Allocation of Software Development Resources in 'Open Source' Production Mode. In Feller, J., Fitzgerald, B., Hissam, S. and Lakhani, K. (Eds) *Making Sense of the Bazaar: Perspectives on Open Source and Free Software*. Sebastapol: O'Reilly Associates.
David, D.D. and Holt, C.A. (1993) *Experimental Economics*. Princeton, NJ: Princeton University Press.
David, P.A., Waterman, A.H. and Arora, S. (2003) First Full Findings from FLOSS-US: The Free/Libre and Open Source Developer Survey for 2003. Working Paper, Stanford Project on Economics of Open Source Software Funded by NSF.
Fagerberg, J., Mowery, D. and Nelson, R. (2004) *The Oxford Handbook of Innovation*. Oxford: Oxford University Press.
Franke, N. and Hippel, E. von (2003) Satisfying Heterogenous User Needs via Innovation Toolkits: The Case of Apache Security Software. *Research Policy*, **32**, 1199–1215.
Franke, N. and Shah, S. (2003) How Communities Support Innovative Activities: An Exploration of Assistance and Sharing Among End-Users. *Research Policy*, **32**, 157–178.
Garzarelli, G. (2003) Open Source Software and the Economics of Organization. In Birner, J. and Garrouste, P. (Eds) *Markets, Information and Communication: Austrian Perspectives on the Internet Economy*. London: Routledge.
Harhoff, D., Henkel, J. and Hippel, E. von (2003) Profiting from Voluntary Information Spillovers: How Users Benefit by Freely Revealing Their Innovations. *Research Policy*, **32**, 1753–1769.
Hars, A., and Ou, S. (2002) Working for Free? Motivations for Participating in Open-Source Project. *International Journal of Electronic Commerce*, **6**(3), 25–39.
Hertel, G., Niedner, S. and Herrmann, S. (2003) Motivation of Software Developers in Open Source Projects: An Internet-Based Survey of Contributors to the Linux Kernel. *Research Policy*, **32**, 1159–1177.
Himanen, P. (2001) *The Hacker Ethic and the Spirit of the Information Age*. New York: Random House.
Holmström, B. (1999) Managerial Incentive Problems: A Dynamic Perspective. *Review of Economic Studies*, **66**, 169–182.
Katz, M.L. and Shapiro, C. (1985) Networks Externalities, Competition, and Compatibility. *The American Economic Review*, **75**(3), 424–440.
Katz, M.L. and Shapiro, C. (1986) Technology Adoption in the Presence of Network Externalities. *The Journal of Political Economy*, **94**(4), 822–841.
Lakhani, K.R. and Wolf, R.G. (2003) Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. Working Paper, MIT Sloan School of Management, n. 4425-03.
Lakhani, K.R. and Hippel, E. von (2003) How Open Source Software Works: 'free' User-to-User Assistance. *Research Policy*, **32**, 923–943.
Lakhani, K.R., Wolf, B., Bates, J. and DiBona, C. (2002) The Boston Consulting Group Hacker Survey. Available at: http://www.osdn.com/bcg/
Lerner, J. and Tirole, J. (2002) Some Simple Economics of Open Source. *Journal of Industrial Economics*, **52**, 197–234.
Levy, S. (1984) *Hackers: Heroes of the Computer Revolution*. New York: Anchor Books.
Mauss, M. (1954) *The Gifts, Forms and Functions of Exchange in Archaic Societies*. London: Cohen & West.
McKelvey, M. (2001) Internet Entrepreneurship: Why Linux Might Beat Microsoft. In Coombs, R., Green, K., Richards, A. and Walsh, V. (Eds) *Technology and the Market*. Cheltenham: Edward Elgar.
Nelson, R.R. (1959) The Simple Economics of Basic Scientific Research. *The Journal of Political Economy*, **67**(3), 297–306.
Nelson, R.R. and Sampat, B.N. (2001) Making Sense of Institutions as a Factor Shaping Economic Performance. *Journal of Economic Behaviour and Organization*, **44**, 31–54.
O'Mahony, S. (2002) The Emergence of a New Commercial Actor: Community Managed Software Projects. Unpublished PhD Dissertation, Stanford University.
O'Mahony, S. (2003) Guarding the Commons: How Community Managed Software Projects Protect Their Work. *Research Policy*, **32**, 1179–1198.
Perman, R. and Scouller, J. (1999) *Business Economics*. Oxford: Oxford University Press.
Raymond, E. (2000) *The Cathedral and the Bazaar*. Available at: http://catb.org/~esr/writings/cathedral-bazaar/
von Hippel, E. and von Krogh, G. (2003) Open Source Software and the 'Private-Collective' Innovation Mode: Issues for Organization Science. *Organization Science*, **14**(2), 209–223.
von Krogh, G., Spaeth, S. and Lakhani, K.R. (2003) Community, Joining, and Specialization in Open Source Software Innovation: A Case Study. *Research Policy*, **32**, 1217–1241.