

**Incentives for Developers' Contributions and Product Performance
Metrics in Open Source Development: An Empirical Exploration**

Ernan Haruvy
School of Management
The University of Texas at Dallas
2601 North Floyd Road
Richardson, TX 75083-0688
Phone: 972-883-4865
Fax: 972-883-6727
E-mail: eharuvy@utdallas.edu

Fang Wu
Department of Marketing and Supply Chain Management
Eli Broad School of Management
Michigan State University
East Lansing MI 48824
Phone: 517-353-6381
Fax: 517-432-1112
E-mail: fangwu@msu.edu

Sujoy Chakravarty*
Economics Area
Indian Institute of Management
Vastrapur
Ahmedabad, Gujarat 380015
Phone: 91 79-2632-4915
Fax: 91 79 2630 6896
E-mail: sujoy@iimahd.ernet.in

*Contact Author

Incentives for Developers' Contributions and Product Performance Metrics in Open Source Development: An Empirical Exploration

Abstract

In open source software development, users rather than paid developers engage in innovation and development without the direct involvement of manufacturers. This paradigm cannot be explained by the two traditional models of innovation, the private investment model and the collective action model. Neither model in itself can explain the phenomenon of the open source model or its success. In order to bridge the gap between existing models and the open source phenomenon, we analyze data from a web survey of 160 open source developers. First, we investigate the motives affecting the individual developer's contributions by comparing and contrasting the incentives from both the traditional private investment and collective action models. Second, we demonstrate that there is a common ground between the private and collective models where private returns and social considerations can coexist. Third, we explore the effect of incentives on the output of innovation—final product performance. The results show that the motivations for individual developer's contributions are quite different from the incentives that affect product performance.

Keywords: Open Source Software, Incentives, Altruism, Developers' Contributions, Software Performance

Business may be imitating life as industries increasingly organize themselves into ecosystems. The ultimate form this evolution may take is the “open source” model of the software industry, in which communities of innovators contribute ideas and build on them beyond the boundaries of any corporate structure... No matter how big, no one company can match the energy and creativity of all those developers out there.

Business Week, April 2001

1. Introduction

Open source software development is a unique form of innovation. The developers—especially users—engage in innovation, development and consumption of a product without the direct involvement of manufacturers (von Hippel 2001). Developers are not directly paid for their efforts and the resulting code is disseminated freely, which makes the extraction of private returns a challenge. The open source software development model has drawn significant attention and generated much excitement in the business world and the developer community in recent years. Unlike many aspects of the rapidly changing landscape of information technology, open source development has emerged from the rubble of the Internet Bust stronger and more prominent. Along with Apache server software, the Perl language, sendmail, GNU and Linux have become pervasive in the corporate world. With the increasing number of open source projects and the growing importance of the open source movement, it is clear that this development paradigm is here to stay.

Although many open source projects are touted as great success stories, the open source phenomenon presents a difficult modeling task for both researchers and practitioners. First, it involves voluntary contributions by software developers without direct monetary incentives. “Why should thousands of top-notch programmers contribute freely to the provision of a public good” (Lerner and Tirole 2002, p.198)? Second, the assumptions and constraints of the two traditional innovation models – the private investment model and collective action model— cannot be examined in separately from each other (von Hippel and von Krogh 2003). In the private investment model, private returns are the key to production and innovation, so the source code is kept closed. In open source development, it is freely shared, and so private returns to manufacturers as well as to programmers must be obtained via less direct routes. In the collective action model, social considerations are critical but such considerations typically dissipate with

larger group sizes and free riding “Is there a common ground between these two innovation models where the incentives can coexist” (von Hippel and von Krogh 2003, p. 11)? Third, the effect of various incentives on the output of innovation remains unknown. Do different motives lead to improved performance of open source products? Is there a significant difference between the effects of incentives on contributions vis-à-vis the effect of incentives on product performance? These issues are crucial and will determine the viability of the open source model in the long run.

The objective of this study is to empirically examine the three distinct yet related issues outlined above. First, we explore the incentive structures that affect an individual developer’s contributions by comparing and contrasting the incentives from the traditional models of innovation. For the private investment model of innovation, we examine direct and indirect monetary incentives, such as job prospects, promotions and salary increases, as well as non-pecuniary motives, such as ego gratification and self-fulfillment. Lerner and Tirole (2002) argue that these “selfish” incentives are strong drivers of open source development. For the collective action model, we examine whether perceived benefits to society and political convictions are the drivers, a view endorsed by Raymond (2001) and Ghosh (2003). We map incentives from these two disparate paradigms in order to determine what guides the contribution behaviors in open source.

Our second aim is to empirically demonstrate the possibility that private returns and social considerations can coexist (von Hippel and von Krogh 2003). The “private-collective” model of innovation may provide an alternative way to explain motives in open source development, which neither conventional innovation models can do (von Hippel and von Krogh, 2003).

Third, the impact of the different incentives on the output of innovation is yet to be shown. In this paper, we examine the effect of different incentives on three dimensions of new product performance—product quality, product creativity and product development speed—in the context of open source development. We propose that the success of new product development will critically depend on striking the right balance between private investment incentives and collective action incentives.

We collect and analyze primary data from a cross-section of open source developers using web surveys to elicit their incentives behind open source contribution as well as their assessment of product performance. Our findings suggest that at the individual level, open source contributors are mainly guided by private level incentives while at the aggregate level, the output of innovation is mainly guided by social considerations. Specifically, such motives as contributing to society and undermining the power of large software houses have a positive and significant influence on product quality, creativity, and speed of product development

Our work joins a few studies in the recent literature that focus on empirical research into the motives behind open source development (e.g., Hars and Ou 2002; Hertel et al 2003; Kogut and Metiu 2001; Lakhani and von Hippel 2000). The incentives we examine are consistent with those investigated in the literature. Some researchers touch upon the quality aspects of open source products (e.g., Stamelos et al. 2002; Zhao and Elbaum 2003), but no one has comprehensively investigated the connection between open source product performance metrics and contribution behavior. We aim to fill this lacuna.

The study proceeds as follows. First, we briefly review the literature and the theoretical background for an integrated private-collective incentive model of innovation. Then, we examine the effect of incentives on open source developers' contributions. Next, we examine how these incentives influence the three dimensions of product performance—product quality, product creativity and speed of product development. The methodology and empirical findings of the study follow. Lastly, limitations of the study and future research opportunities are discussed.

2. Theoretical Background

2.1 Incentives behind the Private Investment Model of Innovation

The seminal works of Lerner and Tirole (2001, 2002) advocate rational economic behavior as the main motive in open source development. In particular, they claim that programmers signal their ability to potential employers by contributing to open source code. In other words, contributions are made with the expectation of future monetary rewards. Dwyer (1999) similarly suggests that contributors are motivated by rewards in the form of higher income that results from the effects of reputation. Hann, Roberts,

Slaughter, and Fielding (2002) maintain that some connection between salary and the rank of the contributors is possible, but no relationship between number of submissions (of program code) and salary has been found. A more direct monetary reward can be realized by developers who serve as consultants to their contributed code (Anez 1999; Bhattacharjee et al. 2001).

Expectations of future monetary reward are clearly not the only driving force behind open source contributions. Even advocates of selfish motives readily acknowledge the possibility of non-monetary considerations. Lerner and Tirole (2002) offer a possible complementary motive in the form of ego gratification, that is, the desire for peer recognition. This view is shared by Raymond (2001, p. 64): “The utility function Linux hackers are maximizing is not classically economic, but is the intangible of their own ego satisfaction and reputation among other hackers.” (Raymond 2001). Also in the personal-fulfillment category are mere enjoyment and personal needs (e.g., Ghosh 1998; O’Reilly 1998; Raymond 2000).

2.2 Incentives behind the Collective Action Model

In sharp contrast to the private motives view, Raymond (2001) likens the open source community to a gift culture, which suggests that altruism is one of the largest motives for contribution. The notion that people take pleasure in doing good is as old as economics itself: “How selfish so ever man may be supposed to be, there are evidently some principles in his nature, which interest him in the fate of others, and render their happiness necessary to him, though he derives nothing from it, except the pleasure of seeing it.” (Smith 1759, pt. 1, section 1) Attributed to Becker (1974) and Harsanyi (1978), the basic altruism model simply implies in this context that the utility of open source programmers is increasing in both their own monetary payoff and others’ monetary payoffs. Such behavior is also termed “pure altruism” (Andreoni 1990; Dawes and Thaler 1988). In contrast, the notion of “impure altruism” (Andreoni 1990; Dawes and Thaler 1988) suggests a warm glow that comes with “doing the right thing”. Impure altruism is generally described as satisfaction of conscience or of non-instrumental ethical mandates (Dawes and Thaler 1988).

Altruism may not be independent of distributional concerns. To the extent that regard for others depends on the distribution of benefits from the software, we may classify it as “fairness considerations”. Haruvy, Prasad, and Sethi (2002) posit fairness or equity considerations as the main set of motives that govern the behavior of open source contributors. As evidence, they discuss the outrage of the open source community at Red Hat for charging exorbitant prices for software, specifically Linux and Apache.¹ Marwell and Ames (1981), Kim and Walker (1984), and Isaac, McCue, and Plott (1985) find that people pay close attention to equity in public good contribution settings and that such behavior generates more socially efficient outcomes.

Political and social considerations have been observed to be critical (Ghosh, 2003). In particular, the desire to limit the power of large software companies and to promote the free flow and exchange of ideas has been at the forefront of the open source movement. The GNU manifesto, the defining document of the open source movement, lists social and political ideas as motivators for open source software development. Ghosh (2003) finds that the majority of open source programmers tend to regard political and social consciousness as critical motives in their decision to contribute to open source code.

2.3 An Integrated “Private-Collective” Model of Innovation

The open source Model deviates from the two traditional models of innovations in several ways (von Hippel and von Krogh 2003). In the private investment model, the key factor is the link between private investment and private returns. In the open source model, the lack of direct payment to contributors and the free dissemination of code make the connection between contribution and returns somewhat obscure. Less direct private rewards to the developer are nevertheless feasible. Consulting opportunities, job prospects and the possibility of promotion and pay raise may result from prominent contributions to open source. Another private reward is self-gratification.

In the collective action model, developers work together and split the rewards. The main advantage is greater efficiency and collaboration, but the major risk is free

¹ Red Hat is a distribution of LINUX (a UNIX-type command line interface created by Linus Torvalds) for which users pay, as opposed to distributions of LINUX like Debian, which are free to install.

riding by people who collect the rewards without putting in the effort. Typically, strong social relationships allow the group to punish free riding within itself. The collaboration in open source development is on a far greater scale than in the traditional collective model, however, free riding is virtually assured. Nevertheless, open source development continues despite evidence of users who are not developers, and it appears that contributors actually regard free riders as an asset (von Hippel and von Krogh 2003).

There are several possible explanations for this seeming contradiction. As stated above, there may be an indirect compensation mechanism whereby developers can be distinguished from non-developers, and private returns can be awarded indirectly. Also, “free riders” in the open source context may be regarded more accurately as lower-level contributors, who typically help in debugging and provide feedback. A third possibility is that social returns are somewhat broader in open source development than typically encountered in the traditional collective action model. Namely, we argue that in the open source context collective considerations are more of a socially benevolent and politically oriented nature, which is consistent with the social enforcement of the traditional collective action model but transcends small group interaction.

In order to determine what guides open source development, we examine the effects of four different incentives on developers’ contributions and the output of innovation. Two are from the private investment model: *monetary reward* and *self-fulfillment*. The other two are from the collective action model, *perceived benefits to society* and *political convictions*. By examining the incentives and their influence at both the individual developers’ level and the aggregate innovation output level, we aim to provide insights that may be helpful to programmers, managers, and policy makers who work in or with the software industry.

3. Individual Developers’ Contributions

We first examine the relationship between incentives and the contributions of individual developers. Individual contribution is conceptualized as a developer’s assessment of his or her efforts and productivity in open source software development.

3.1 Private Incentives

Under private incentives, we examine both monetary and non-monetary motives. Monetary motives are connected to direct or indirect future financial payoffs, such as job prospects, promotion, and salary increases. Self-fulfillment, a non-monetary private incentive, is defined as the personal gratification gained from working on an open source project, such as enjoyment and an elevated reputation.

Our monetary rewards component follows directly from Lerner and Tirole's (2002) career concern incentive. Even when there is no immediate monetary payoff, the potential for future financial compensation can be a key factor for developers' contributions to open source development. The future financial payoff includes the expected stream of returns over the programmer's lifetime. Even though the programmer "volunteers his time for free," the programmer's contribution to open source can serve as a strong signal of skill among peers and future employers, which may result in better job prospects, higher future salaries, and more freelance consulting projects. Open source projects typically recognize that major developers and programmers often list such projects on their resume. These practices suggest a signaling explanation with monetary compensation as a possible goal. In signaling models, which began with Spence's (1974) labor market signaling, an employer is willing to pay more for higher quality workers but can do so only if they reveal themselves by taking an action that is not optimal for lower quality workers.

Spence (1974) proposes that workers can signal their quality by the level of education they pursue. In essence the cost (disutility) of education is negatively correlated with ability, such that the more capable individuals expend less effort on education than less capable individuals. In a separating equilibrium, less able individuals do not find it profitable to signal (i.e. – pursue a high level of education), as the cost of education for them exceeds the potential gain in wage. Therefore, the equilibrium outcome comprises high quality workers receiving a higher level of education (thereby signaling their type) and higher wages and the low ability workers receiving a lower level of education and lower wages. The same paradigm applies to the signaling of ability in the job market for programmers. If software developers and prospective employers have common knowledge regarding the presence of a separating equilibrium, then the

credentials of open source contribution would serve to signal high programming ability. The signaling incentive would encourage open source contributions from prospective employees, if only to reveal their programming abilities (Dwyer 1999; Lerner and Tirole 2002).

Hypothesis 1: A higher perceived monetary leads to increased contributions to open source projects.

Monetary rewards are not the only private incentives that may influence open source contributions. Linus Torvalds wrote the core of Linux, which is possibly the most popular open source project to date. When he was asked why he released his work, he replied that initially it was an act of self-fulfillment; a sense of pride in what he had accomplished. (Ghosh1998). Raymond (2001, p. 32) alludes to this motivation of open source contributors when he states that “every good work of software starts by scratching a developer's personal itch.” That is, one aspect of self-fulfillment is defined as mere enjoyment or a personal need.

Other studies (Lovio-George 1992; Graham and Unruh 1990) show that, aside from direct monetary rewards, workers attribute their productivity to such rewards as a “pat on the back,” which provides satisfaction. Ego gratification — the desire for peer recognition— is proposed as a major incentive by Lerner and Tirole (2002) and Raymond (2001). We refer collectively to these two sets of incentives (enjoyment and ego gratification) as self-fulfillment, or personal non-monetary benefit. In addition, Bergquist and Ljungberg (2001) argue that gift-giving is the foundation of open source development. New ideas and prototypes are freely given away for peers to review and circulate, which is one way to guarantee the quality of the code. In the digital community, this type of gift-giving behavior also can reward the giver with a certain amount of fame and respect.

Hypothesis 2: Greater perceived self-fulfillment leads to increased contributions to open source projects.

3.2 Collective Incentives

The collective incentives examined here are *perceived benefits to society* and *political convictions* of the developers. Perceived benefits to society refer to motivations that are linked to altruism and social responsibility, that is, contributions to the “common good.” Political convictions refer to a developer’s belief in collectivizing in order to promote societal goals such as free competition and the limited power of large corporations. These two sets of incentives are rooted in the collective action model, which is centered on altruism. Limiting the power of major software companies (political convictions), exchanging knowledge (reciprocity), providing more software variety for users (altruism) and engendering new forms of cooperation (community building), can be important motives for contributors.

Altruism, reciprocity, a collaborative spirit and political convictions have long been cited as important reasons for the growth of the open source movement. Ghosh (2003) found that a large member of the open source contributors he surveyed cited non-selfish reasons as motivating factors. Raymond (2001) believes social considerations are crucial in the development of the open source community. If developers think their efforts will benefit society as a whole, they are more likely to contribute to a project and share code with others. Such benevolent behavior can complement other motivations for the open source development.

Hypothesis 3: A greater perceived benefit to society leads to increased contribution to open source projects.

In addition to concern for the welfare of society, a programmer may believe strongly in the open source paradigm as a tool for community building and as a way to mitigate of the power of large software companies. We would expect these political convictions to positively and significantly affect the desire to contribute to open source projects. Such convictions and a concern for society are not mutually exclusive. Strong political convictions generally arise from the belief that a particular legal, economic, or social structure would benefit society. The difference between the two sets of incentives lies in the indirect nature of political convictions as compared to a direct desire to benefit society. Thus, a programmer who is motivated primarily by political considerations may

want a change in the social order (e. g., less power for corporate giants or greater social equity) which he believes will lead to greater social welfare.

Hypothesis 4: Strong political convictions lead to increased contributions to open source projects.

These four categories of incentives may not cover all the motives that drive programmers. Humans are motivated by a complex web of social, environmental, physical, and other factors that are beyond the scope of this work. Nevertheless the private and collective dimensions encompass the main schools of thought regarding the incentives of open source contributors and can provide critical insights to the administration and organization of open source projects.

4. Open Source Product Development

An understanding of what makes contributors tick is critical to designing incentives for the open source community, but uncovering the link between incentives and different parameters of product development is equally crucial for the viability and future success of open source development. The general consensus among open source advocates is that this paradigm tends to result in software of better quality and higher reliability. When the source code is open, users as well as developers are actively involved in testing, reporting, and debugging the product (*Economist* 2001). Raymond (1999, p. 41) states that “given enough eyeballs, all bugs are shallow.” Nevertheless, there are concerns about open source product quality, and there is a lack of empirical evidence and validation on whether open source products perform better than proprietary software (Stamelos et al., 2002). One objective of our study is to explore this important aspect.

We examine the link between the incentives to contribute and various dimensions of product development. The incentives, monetary and non-monetary, may not affect product development and individual contributions in the same way. If we detect strong relationships between certain incentives and product development within a software

process,² then we can recommend administrative policy with regard to incentive structure for programmers. This would be especially useful for open source development that takes place in commercial software firms.³ Based on the new product development literature (e.g., Griffin 1993, 1997; Moorman 1995; Sarin and Mahajan 2001; Sethi 2000) and research on software performance metrics (e.g. Krishnan et al., 2000; Harter et al., 2000), our focus is on examining three characteristics of open source product development—product quality, product creativity and product development speed.

The literature views product quality and development speed as two important measures of software product development. According to Dekleva and Drehmer (1997), software processes are improved by such work practices as training, quality assurance, measurement, and design and code reviews. Krishnan et al. (2000) identify and utilize several metrics of productivity and quality of a software development process: size of code, quality, life-cycle productivity, usage of tools, personnel capability and front end resources. Boehm's (1981) COConstructive COst MOdel (COCOMO) for software cost estimation examines the relationship between development time and contribution effort: time declines exponentially with an increase in contribution effort. This suggests that development speed may rise with an increased level of contribution. Harter et al. (2000) employ measures similar to Krishnan et al. (2000) and investigate the relationship between software process maturity, quality, cycle time, and effort. They find that a higher level of software process maturity is associated with higher quality, reduced cycle time, and greater effort.

Technology-savvy users can enhance product quality in the open source sphere in a manner unrivaled by proprietary software. Their suggestions can lead to improvements in quality unattainable by standard debugging alone. Support for using this measure is provided by the Asundi et al. (2002) study of the Apache web server, which finds a large number of Apache users willing to report bugs in a structured manner as compared to commercial software development. Furthermore, Ljungberg (2000, p. 208) confirms that open source software often attains a “quality that outperforms commercial proprietary

² A software process is a set of defined procedures that lead to the development of software products. (Zahran, 1998).

³ JIKES, BSF, JFS and DPCL projects sponsored by IBM, the Darwin project by Apple Computers and the Grid Engine project by Sun Microsystems are some such examples from where we draw participants for our study.

software.” In addition, the high quality of open source software is often attributed to complete accountability for performance. Pavlicek (2001, p. 60) notes that in open source software development, “quality truly is job one when there is no corporate PR engine to come along afterward and gloss over your shortcuts.”

Another salient characteristic of open source development is the unparalleled speed of development when rapid information exchange occurs among tens of thousands of developers, users, and experts in various fields. According to Gilmer (2002), open source code evolves quickly by incorporating “best of breed” technologies. Hacker communities borrow the best ideas from other projects (which is not illegal for open source products) to develop a certain program in record time.

We also examine a third dimension, product creativity, or the degree of innovation inherent in a product (Moorman 1995; Moorman and Miner 1997). It indicates whether a product has the potential to meet untapped market needs or even create new demands (Olson, Walker and Ruekert 1995). Product creativity is one way to examine whether a product has a competitive advantage in the software world. The relationship between programmer incentives and product creativity is worth examining in the context of open source projects because innovation is a salient characteristic of open source as opposed to the rigid corporate routines of traditional software development. In this vibrant community of voluntary contributors, the level of commitment may surpass that of employees who merely get the job done. According to Jordan and Segelod (2002), software processes today yield products of increasing creativity and innovation that raise productivity and have a broad influence on the economy. Accordingly, creativity and innovativeness have long been metrics in the literature for product development and are identified as key factors in the software process. Von Hippel (2001) regards open source programmers as a user innovation community sensitive to the needs of others like themselves and with a high degree of skill and creativity. They often come up with products such as Apache, which routinely outperform competing commercial counterparts like Netscape and Microsoft server software. Francke and von Hippel (2002, p. 1213) describe the open source software as a “toolkit for innovation” whereby solutions in the software process are engendered by the creative output of the user

community. Shah (2003) and Hertel et al. (2003) believe creativity is a defining characteristic of the open source software development paradigm.

Using these three metrics for software development, we detail the linkages between incentives and product performance in the next two sections. Whether open source development can provide software of higher quality, developed faster, and of greater creativity than proprietary processes will determine the viability and long-term prospects for this model. Next, we present our hypotheses regarding the linkage between the four incentives behind OS development and the three product performance measures.

4.1 Private Motives and Product Development

The use of rewards to manage product-related performance is documented in the context of new product development teams in the traditional organization (e.g., Sarin and Mahajan 2001; Sethi 2000). Outcome-based rewards, such as bonuses linked to the profitability of a firm, have been shown to be positively related to product quality (Sarin and Mahajan 2001). In open source development, users contribute to software code voluntarily. Developers generally receive no immediate monetary rewards for their work (for exceptions, see Hars and Ou 2002), but it can serve as a strong signal of their future career potential (Lerner and Tirole 2002). This expected payoff can encourage users to contribute to open source software for the sake of their future welfare.

The nature of open source facilitates user involvement because lead users can play a key role in product development (von Hippel 2001). They can identify the need for innovations, and even help solve problems (Urban and von Hippel 1988; von Hippel 1986). They are the key source for innovation (von Hippel 1986). Their effort to fix bugs and add new features to an existing program addresses their own needs (Hertel et al. 2003) and enhances the quality of a product (Sethi 2000).

Future monetary rewards can speed up open source product development for several reasons. First, because of modularity, widely dispersed developers can work concurrently on improving the program (Kogut and Metiu 2001). The modular design and the source code availability allows for parallel debugging, which leads to faster software evolution (Zhao and Elbaum 2003). User participation and peer monitoring can help ensure the reliability of a product while reducing the cycle time. Second, monetary

rewards may help spark the creativity that goes into a product. The open source environment encourages creativity as there is no strict hierarchical management structure to suppress unconventional ideas and discourage directions that are not immediately profitable. Developers are working for something that matter to them, and they enjoy doing it (Torvalds, 1998). Third, software creativity and innovativeness are determined by the amount of knowledge acquired and assimilated through the learning process (Rindfleisch and Moorman 2001). The open and informal environment of open source encourages rapid knowledge sharing among developers who are working towards the creation of better quality software to compete with proprietary software.

Hypothesis 5: A higher perceived monetary reward leads to (a) higher product quality, (b) faster development speed, and (c) higher product creativity in open source projects.

The desire for self-fulfillment is likely to lead to creativity in one's chosen form of expression. This self-fulfillment is augmented when a person can work on ideas that address his or her needs (Fischer and Nakakoji 1992). Most open source projects begin with addressing an individual need and thus a high degree of problem-solving innovativeness is built into them. Because the open source community gives a developer considerable control over the distribution of code, the creator of new software can share the code with other members of the community. Doing so helps to create a climate that encourages a broad range of people to join in and find more creative solutions to the problem at hand (Fischer 1999). Thus, a desire for self-fulfillment in conjunction with the highly interactive open source environment can lead to creative answers to programming challenges. This argument is consistent with the motivations posited by Raymond (2001) and Linus Torvalds, who developed the Linux kernel (Ghosh, 1998). Accordingly, we conjecture that self-fulfillment is positively associated with open source product creativity.

Hypothesis 6: Greater perceived self-fulfillment leads to higher product creativity in open source projects.

4.2 Collective Motives and Product Development

As previously discussed, developers may contribute to open source projects because of altruism, that is, they want to increase the welfare of other people in society. Developers motivated by altruism (among other possible motivations) allow the open source community free and unhindered access to their creative output. This benevolent act has obvious ramifications for quality control since other users and peers can test and inspect the newly generated code (Bergquist and Ljungberg 2001). Thus, collective motives behind open source contribution may actually lead to an enhancement in product performance. Linux, which is one of the most highly regarded open source projects, was created because large number of Unix developers were unhappy that end-users relied too heavily on Microsoft's proprietary server software, which among its other shortcomings was alarmingly vulnerable to virus attacks. They wanted a new system of higher reliability and security, and today Linux has been adopted by a number of large commercial firms, including IBM, Apple, and Sun (West 2003). Therefore, a "social orientation" can lead to the development of better software. Ghosh (2003) asserts that the need to collectivize in solidarity and subvert the power of large corporate entities plays a significant role in raising the level of software development.

Since the open source environment can allow for code writing, testing and debugging at the same time, this overlap in the new product development process can significantly reduce cycle time (e.g., Griffin 1993, 1997). Therefore, the same altruism that can lead to higher product quality also can result in shorter turnaround times for open source projects.

The association between social motives and creativity is worth examining as well. We conjecture that perceived benefits to society and political convictions can be conducive to more creativity among open source developers during their course of creating a better product. As argued by Fischer (1999, p.10), a major challenge for social creativity is to "allow end-users to become co-developers of systems." Because the bedrock of the open source model is user participation, we argue that it is well equipped to overcome this challenge. Collaborative knowledge sharing, feedback in the form of bug reports, and a commitment to community building ensure that a more innovative

solution to the problem is obtained through open source as opposed to a more proprietary development pattern (Fischer 1991).

Hypothesis 7: A greater perceived benefit to society leads to (a) higher product quality, (b) faster development speed, and (c) higher product creativity in open source projects.

Hypothesis 8: Stronger political convictions lead to (a) higher product quality, (b) faster development speed, and (c) higher product creativity in open source projects.

5. Methodology

5.1 Data Collection

Data were collected by emailing 2000 programmers listed as contributors on open source web pages and on selected open source developer lists. A second round of email was sent two weeks later. Respondents were assured of confidentiality and were given a brief summary of the proposed study as well as their roles as participants. Of the 164 responses received, four were unusable due to missing values. A test for non-response bias was completed following the Armstrong and Overton (1977) procedure to detect differences in the means between early and late respondents. Based on the analysis, no significant differences were found between the sample groups on the variables of interest.

5.2. Characteristics of Survey Participants

Although we did not ask for nationality, the email return addresses represented at least 18 countries and five continents (North America, South America, Europe, Asia, and Australia). More than 90 distinct open source projects were represented (not counting modules as separate projects), of which the most common were Apache, Linux, Open Office, and Mozilla.

Among respondents, 19% were self-employed and 11% were unemployed. The rest were employed by a firm. The level of education was relatively high: 21% with a Ph.D., 16% with a master's degree, 31% with a bachelor degree, 26% still in college, and 6% still in high school. The vast majority (74%) had more than five years of programming experience. Most had participated in the open source movement for more

than five years, and the average was 4.7 years. The majority of respondents (52%) were involved in no more than two open source projects. Fifteen respondents had worked on at least ten separate projects, which brought the average to about four projects per contributor. Approximately 80% of respondents indicated that they dedicated no more than one or two hours per day to open source and 10.6% dedicated 5 hours or more per day.

5.3 Measures, Reliability and Validity

Relevant measures for the constructs were identified in the literature, modified, and refined according to information gained during preliminary personal interviews. All items were measured on 7-point Likert scales (1 = strongly disagree, 7 = strongly agree). The respondents were asked to reflect upon a specific open source project to which they had contributed in the past year. Following Anderson and Gerbing (1988), we purify the measures by assessing the reliability and unidimensionality of each construct. We first examine item-to-total correlations within each construct and delete items with low correlations. We then subject the items to principal component analysis (PCA) and confirmatory factor analysis (CFA) using EQS. During the purification, items that do not load heavily on the primary factor or those that have high cross-loadings are dropped. The CFA yield adequate model fit for the proposed factor structure ($\chi^2 = 642.962$ with degree of freedom (d.f.) = 497, Comparative Fit Index (CFI) = 0.945, Bollen Fit Index (IFI) = 0.947, Root Mean Squared Error of App. (RMSEA) = 0.043). All items loaded on their respective constructs and are statistically significant. Furthermore, the Cronbach alpha for each construct is above the 0.70 suggested by Nunnally (1978), which indicates adequate reliability. Table 1 lists the measures used in the final study, the factor loadings, and construct reliability. Descriptive statistics and correlations are presented in Table 2.

<TABLE 1 ABOUT HERE>

<TABLE 2 ABOUT HERE>

6. Analysis and Results

We first examine the ratio of unselfish (average of perceived social benefits and political convictions) and selfish motives (average of monetary rewards and self-fulfillment). The results reveal that, in sharp contrast to the well-accepted selfish motives theory of Lerner and Tirole (2001, 2002), unselfish motives are far more predominant than monetary considerations. Responses to the relevant survey items are almost without exception clustered around the high end of the Likert scale. Hierarchical clustering methods reveal no more than one cluster of significant size for greater majority of the respondents (86%); the unselfish-to-selfish ratio is equal to or greater than to the 1.25 at the mode. Especially noteworthy is the bin immediately to the right of the mode, with 27% of respondents, in which unselfish motives are 50% greater than selfish motives.

6.1 The Relationship of Incentives to Individual Contribution

We use a logistic regression to examine the relationship between incentives and individual contributions. The dependent variable, individual contribution, is converted into a binary variable. It has a value of 1 for Likert ratings of 6 or 7, indicating a high level of contribution, and 0 for ratings 1 through 5, indicating a low to moderate level of contribution. As shown in Table 3, monetary rewards and self-fulfillment both positively influence developers' contributions. Thus, H1 and H2 are supported. Contrary to what we expected, perceived benefits to society are not significantly associated with developers' contributions and political convictions are negatively rather than positively associated. Thus, H3 and H4 are not supported. Although the open source community as a whole is known to espouse political and social convictions (Ghosh 2003), it appears that these do not influence the more active contributors. The negative association for political motives however does not necessarily imply that politically minded individuals are unlikely to join the open source movement.

<TABLE 3 ABOUT HERE>

To determine whether the open source community consists of two segments or subpopulations, we use the mixture model approach (Kamakura and Russel 1989), which

allows all the parameters in the choice model to be heterogeneous. The Kamakura-Russel method adopted here assumes the underlying distribution of parameters to be discrete and is a special case of the random coefficient model, which typically assumes a continuous distribution. The advantage of this approach is that consumer segments may be identified (Briesch et al. 1997). The log likelihood from the two segments (-87.03) is not significantly different from that of one segment (-91.28). The results show that the populations of our sample are fairly homogeneous.

6.2 The Relationship of Incentives to Product Development

To examine the effect of incentives on software development, we conduct seemingly unrelated regressions (controlling for correlations of the error terms between the regression equations) to uncover some patterns regarding the effect of incentives on the metrics for software development (see table 4). We control for the demographic variables of education, employment, and the degree of open source participation to account for the possibility that these affect the contributor's perceptions in assessing product success. We also account for a product development stage and the perceived level of a firm's expenditures in the regression equations for product quality and product development speed, but not in the equation for product creativity. Except for the degree of open source participation, which is significant only in the quality regression, none of the demographic effects are significant. The rest of the estimates are reported in table 4.

<TABLE 4 ABOUT HERE>

The results show that monetary rewards and perceived benefits to society positively affect all three dimensions of product performance. Thus, H5 and H7 are supported at the 5% significance level. In other words, a perception of higher future rewards and the desire to provide benefits to society encourage learning and social creativity among developers. The result is a software product developed faster and of higher quality compared to the proprietary process. *The coexistence of these two*

incentives indicates common ground for private and collective motives in open source development.

Self-fulfillment does not seem to have a significant influence on product quality and product development speed. It does seem to be significantly associated with product creativity however, as H6 is supported at the 10% level of significance. This result shows that individual creativity is channeled by the open source model, and it reflected in the higher level of product creativity under open source development. That is, product creativity is enhanced as a result of the self-fulfillment obtained from generating code. Political convictions are positively associated with product quality and development speed, but they do not appear to have any significant influence on product creativity, so H8 is only partially supported. In other words, political convictions may make programmers work faster and focus on quality, but do not affect product creativity. Thus product creativity has little to do with the intensity of the political ideology to which a developer subscribes.

7. Discussion and Implications

As open source becomes an increasingly common form of software development, firms need to understand which incentives will enhance the productivity of their programmers. Managers of profit-oriented corporations have in the past been reluctant to harness the power of the open source movement, but today companies such as IBM, Apple, Sun, Oracle, Compaq, and Dell are increasingly adopting this concept. Unfortunately, there are no clear guidelines as to the best way to manage incentives and reap the benefit from open source initiatives.

Our goal was to provide survey evidence on the motivations of open source programmers and to link such motives to both individual contributions and product development characteristics. Although motives stem from a mosaic of economic, social, and political realms, we have classified them into two categories—private and collective. Our survey of open source programmers reveals that such private incentives as future monetary rewards and self-fulfillment positively influence individual developers' direct contributions. This finding is consistent with Lerner and Tirole's (2002) proposition that

selfish motives are the primary driver of open source contribution. A different pattern of motives emerges, however, when we map the connections between motives and open source product development parameters. Social considerations seem to dominate as drivers of product quality and product development speed, though monetary considerations continue to remain important after controlling for education, employment, the degree of open source involvement, product development stage and firm expenditures. With regard to innovation output, there appears to be common ground for private and collective motives. This is consistent with research by von Hippel and von Krogh (2003). Yet, our results should be interpreted with caution. Although the survey responses are unequivocal, they may not fully reflect the strength of emotions. Quite a few respondents sent emails expressing indignation at survey items which suggested monetary considerations could possibly motivate their contributions to open source projects.

Several managerial implications can be derived from our findings. First, monetary incentives alone are not likely to result in successful software development in the open source realm. Attention should be paid to social considerations in order to ensure product quality. In firms that adopt the open source model, managers may need to focus on sources of productivity that traditionally have not been deemed very important.

Second, it may be desirable to encourage self-fulfillment as an organizational goal for employees, given our finding that it affects product creativity. Accordingly, firms may need to place greater emphasis on recognition and praise for creative work.

Third, political considerations appear the least consequential in terms of creativity but highly relevant for product quality and speed of development. Therefore, small firms might be advised to support the open source movement, emphasize a business policy that champions collaborative work processes, and avoid the intense employee competitiveness that characterizes some large corporations. Major firms such as IBM and Apple Computers have also created communities of developers (projects such as JIKES and DPCL sponsored by IBM and Open Darwin sponsored by Apple Computers) within the organization that are relatively free of corporate hegemony and have a high degree of control over their product.

Fourth, it appears that individual level contributions are highly influenced by monetary rewards although our survey reveals a critical role of non-pecuniary motives as well. This may reflect the characteristics of our sample, which is composed of open source developers only. They may have decided to join the open source movement for ideological reasons and then became more interested in potential monetary rewards. In other words, ideological motivations may matter more at the time a person decides whether or not to join an open source project, but diminishes in importance as he gathers more open source programming experience. Future research can incorporate programmers who are contemplating writing open source code for the first time and examine their decision choice of joining the open source movement. We expect that social considerations probably matter to a large extent in that decision.

Fifth, the different patterns of incentives that drive individual contributions and the aggregate level of innovation output suggest some practices for managing open source contributions in the corporate environment. At the individual level, we can put more emphasis on providing such factors as indirect future income and the need for self-fulfillment that boost the developers' contributions. For example, give a developer responsibility for managing the creative output of a project, which would have long-term career benefits. . Also, enjoyment may be enhanced by creating a state of the art code submission and integration interface that gives developers the creative freedom, without bureaucratic and administrative restrictions. At the innovation output level, because social considerations play an important role, it may be necessary to foster a sense of community, emphasize collective behavior, and allow for a wide range of political orientations in order to promote a creative atmosphere.

The results of this study provide some guidance for future research on open source development. It has been suggested that a hybrid open source model may perform well in software development firms. Here the traditional software firm incorporates open source practices in order to better manage the software development process (Sharma et al. 2002). West (2003) examines three case studies in which proprietary software firms, such as Apple, IBM and Sun, incorporate the open source model into their software development platform. Future research can focus on such efforts, particularly the

appropriate governance forms and incentive structures that promote innovation in a hybrid environment of this type.

References

- Anderson, James C. and David W. Gerbing (1988), "Structural Equation Modeling in Practice: A Review and Recommended Two-Step Approach," *Psychological Bulletin*, 103 (3), 411-423.
- Andreoni, J. (1990) Impure Altruism and Donations to Public Goods: A theory of warm glow giving. *Economic Journal*, Vol. 100, pp. 464-477.
- Anez, Juancarlo (1999), Programmers or entrepreneurs? *Borland Developer Network*, November 01, 1999, <http://community.borland.com/article/print/0,1772,10550,00.html>
- Armstrong, J. Scott and Terry S. Overton (1977), "Estimating Nonresponse Bias in Mail Surveys," *Journal of Marketing Research*, 16 (August), 396-400.
- Asundi, J., R. Kazman and V.S. Arunachalam (2002) "Understanding the Design process in Open Source Software Projects: The Case of the Apache Webserver", mineo.
- Becker, Gary S. (1974), "A Theory of Social Interaction," *Journal of Political Economy*, 82, 1063-1093.
- Bergquist, Magnus and Jan Ljungberg (2001), "The Power of Gifts: Organizing Social Relationships in Open Source Communities," *Information Systems Journal*, 11, 305-320.
- Bhattacharjee, A., B. Mishra, A., Prasad, S. Raghunathan (2001), Open source versus Closed source: Software Quality in Monopoly and Competitive markets. Working paper, The University of Texas at Dallas.
- Boehm, B. W. (1981), *Software Engineering Economics*. Prentice-Hall Inc. New Jersey, NJ.
- Briesch, Richard A. Lakshman Krishnamurthi, Tridib Mazumdar, and S.P. Raj (1997), "A Comparative Analysis of Reference Price Models," *Journal of Consumer Research*, 24(2), 202-215.
- D'andrade, R. 1978, "U-Statistic Hierarchical Clustering" *Psychometrika*, 4:58-67.
- Dawes, R. and R. Thaler (1988), "Cooperation," *Journal of Economic Perspectives*, 2, 187-197.
- Dekleva, S. and Drehmer, D. (1997). Measuring software engineering evolution: A Rasch calibration. *Information Systems Research*, 8(1), 95 – 105.
- Dwyer, Jr., G. P. (1999), *The Economics of Open Source and Free Software*, Working Paper.
- Economist (2001), Survey: out in the open, 359, S8-S14.
- Fischer, Gerhard (1991), "Supporting Learning on Demand with Design Environments," International Conference on the Learning Sciences, 165-172.
- Fischer, Gerhard (1999) "Domain-Oriented Design Environments: Supporting Individual and Social Creativity", in J. Gero and M.L. Maher (eds): "Computational Models

- of Creative Design IV", Key Centre of Design Computing and Cognition, Sydney, Australia, 1999, pp 83-111.
- Fischer, Gerhard. and K. Nakakoji (1992), "Beyond the Macho Approach of Artificial Intelligence: Empower Human Designers—Do not Replace Them," *Knowledge Based Systems Journal*, 5(1), 15-30.
- Francke, N. and von Hippel, E. (2003) "Satisfying Heterogeneous User Needs via Innovation Toolkits: The Case of Apache Security Software." *Research Policy*, 32, 1199-1215.
- Ghosh, R. A (1998), "First Monday Interview with Linus Torvalds: what motivates free software developers?" http://www.firstmonday.dk/issues/issue3_3/torvalds/
- Ghosh, R. A. (2003), "Understanding Free Software Developers: Findings from the FLOSS Study, <http://opensource.mit.edu/papers/ghosh.pdf>, also forthcoming in *Making Sense of the Bazaar: Perspectives on Open Source and Free Software*," ed. J. Feller, B. Fitzgerald, K. Lakhani and S. Hissam, O'Reilly.
- Gilmer, B. (2002), "Open Source Development" *Broadcast Engineering*, September.
- Graham, G. H. and J. Unruh (1990), The Motivational Impact of Non-Financial Employee Appreciation Practices on Medical Technologies, *Health Care Supervisor*, 8(3), 9-17.
- Griffin, Abbie (1993), "Metrics for Measuring Product Development Cycle Time," *Journal of Product Innovation Management*, 10, 112-125.
- Griffin, Abbie (1997), "The Effect of Project and Process Characteristics on Product Development Cycle Time," 34(1), 24-36.
- Gusfield, Joseph (1978), *Community: A Critical Response*, New York: Harper & Row.
- Hann, II-Horn, Jeff Roberts, Sandra Slaughter and Roy Fielding (2002), "Why Do Developers Contribute to Open Source Projects? First Evidence of Economic Incentives," working paper.
- Hars, Alexander and Shaosong Ou (2002), "Working for Free? Motivations for Participating in Open-Source Projects," *International Journal of Electronic Commerce*, 6(3), 25-39.
- Harsanyi, John C. 1978. "Economics and Ethics: Altruism, Justice, Power: Discussion," *American Economic Review*, V. 68(2), 231-32.
- Harter, D., Krishnan, M. S. and Slaughter, S. 2000. Effects of Process Maturity on Quality, Cycle Time, and Effort in Software Product Development. *Management Science*, 46(4) 451-466.
- Haruvy, E., A. Prasad, and S. Sethi (2002), Harvesting Altruism in Open Source Software Development, forthcoming in *Journal of Optimization Theory and Applications*.
- Hertel, G., Niedner, S., and S. Hermann (2003). "Motivation in Open Source Projects: An Internet-based Survey of Contributors to the Linux Kernel. *Research Policy*, Volume 32(7), 1159-1177

- Isaac, R.M., K.F. McCue, and C.R. Plott, "Public Goods Provision in an Experimental Environment," *Journal of Public Economics*, 26, 1985, 51-74.
- Johnson, S.C. 1967, "Hierarchical Clustering Schemes" *Psychometrika*, 2:241-254.
- Jordan, G. and Segelod, E. (2002). Software Innovativeness – Knowledge Acquisition, External Linkages and Firm Developmental Processes. FE rapport 2002-392, School of Economics and Commercial Law, Göteborg University
- Kamakura, Wager and Gary J. Russel (1989), "A Probabilistic Choice Model for Market Segmentation and Elasticity Structure," *Journal of Marketing Research*, 26(4), 379-390.
- Kim, O. and M. Walker, "The Free Rider Problem: Experimental Evidence," *Public Choice*, 43, 1984, 3-24.
- Kogut, Bruce and Anca Metiu (2001), "Open-Source Software Development and Distributed Innovation," *Oxford Review of Economic Policy*, 17(2), 248-264.
- Krishnan, M.S., C. H. Kriebel, S. Kekre, and T. Mukhopadhyay (2000), An Empirical Analysis of Productivity and Quality in Software Products, *Management Science*, 46(6), 745-759.
- Lakhani, K. R. and E. von Hippel (2003), "How Open Source Software Works: 'Free' user-to-user assistance," *Research Policy*, 32, 923-943.
- Lerner, J., and Tirole, J. (2001), The Open Source Movement: Key Research Questions, *European Economic Review*, 45, 819-826.
- Lerner, J., and Tirole, J. (2002), *Some Simple Economics of Open Source*. *Journal of Industrial Economics*, L(2), 197- 234.
- Ljungberg, J. (2000), "Open Source Movements as a Model for Organizing" *European Journal of Information Systems*, December 2000.
- Lovio-George, C. (1992), What Motivates Best? Sales and Marketing Management, April, 1992.
- Marshall, Gordon (1994), *The Concise Oxford Dictionary of Sociology*, Oxford: Oxford University Press.
- Marwell, Gerald and Ames, Ruth E. "Economists Free Ride, Does Anyone Else? Experiments on the Provision of Public Goods, IV." *Journal of Public Economics*, 15(3), June 1981, pp. 295-310.
- Maxwell, K, Van Wassenhove, L. , and S. Dutta (1999), Performance Evaluation of General and Company Specific Models in Software Development Effort Estimation, *Management Science*, 45(6), 787-803.
- Moorman, Christine (1995), "Organizational Market Information Processes: Cultural Antecedents and New Product Outcomes," *Journal of Marketing Research*, 32(August), 318-335.
- and Anne S. Miner (1997), "The Impact of Organizational Memory on New Product Performance and Creativity," *Journal of Marketing Research*, 34(Feb), 91-106.

- Nunnally, Jum C. (1978), *Psychometric Theory*, (2 edition). New York: McGraw-Hill Book Company.
- Olson, Eric M. Orville C. Walker, Jr. and Rober W. Ruekert (1995), "Organizing for Effective New Product Development: The Moderating Role of Product Innovativeness," *Journal of Marketing*, 59(Jan.), 48-62.
- O'Reilly, T. (1998), *The Open Source Revolution. Release 1.0*, November 1998.
- Pavlicek, R. (2001), "In the Name of Quality" *Infoworld*, 12.10.01.
- Raymond, E. S. *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly, Sebastopol, CA, 2001.
- Rindfleisch, Aric and Christine Moorman (2001), "Acquisition an Utilization of Information in New Product Alliances: A Strength-of-Ties Perspective," *Journal of Marketing*, 65(April), 1-18.
- Sarin, Shikhar and Vijay Mahajan (2001), "The Effect of Reward Structures on the Performance of Cross-Functional Product Development Teams," *Journal of Marketing*, 65(April), 35-53.
- Sethi, Rajesh (2000), "New Product Quality and Product Development Teams," *Journal of Marketing*, 64(April), 1-14.
- Shah, S. (2003). *Understanding the Nature of Participation and Coordination in Open and Gated Source Software Development communities*. MIT Sloan School of Management, working paper.
- Sharma, S., V. Sugumaran & B. Rajagopalan (2002), "A Framework for Creating Hybrid-Open Source Software Communities," *Information Systems Journal*, 12, 7-25.
- Smith, Adam (1759), *The Theory of Moral Sentiments*. Edinburgh.
- Spence, A. M. (1974), *Market Signaling*. Cambridge: Harvard University Press, Cambridge, Mass.
- Stallman, R. *The GNU Manifesto*, <http://www.gnu.org/gnu/manifesto.html>, 1985.
- Stamelos, I., L. Angelis, A. Oikonomou and G. L. Bleris (2002), "Code Quality Analysis in Open Source Software Development," *Information Systems Journal*, 12, 43-60.
- Urban, Glen L. and Eric von Hippel (1988), "Lead User Analyses For the Development of New Industrial Products," *Management Science*, 34(5), 569-582.
- Von Hippel, Eric (1986), "Lead Users: A Source of Novel Product Concepts," *Management Science*, 32, 791-805.
- von Hippel, Eric (2001), "Innovation by User Communities: Learning from Open Source Software." *MIT Sloan Management Review*, Summer 2001.
- von Hippel, Eric and Georg von Krogh (2003), "Open Source Software and the 'Private-Collective' Innovation Model: Issues for Organization Science," *Organization Science*, 14(2), 209-223.

Weber, Max ([1922] 1978), *Economy and Society*, Berkeley: University of California Press.

West, Joel (2003), "How Open is Open Enough? Melding Proprietary and Open Source Platform Strategies," *Research Policy*, 32, 1259-1285.

Zahran, S. (1998). *Software Process Improvement: Practical Guidelines for Business Success*. Addison-Wesley Longman Ltd., Essex, U.K.

Zhao, Luyin and S. Elbaum (2003), "Quality Assurance Under the Open Source Development Model," *The Journal of Systems and Software*, 66, 65-75.

Table 1: Construct Measurement, CFA Standardized Loadings, and Construct Reliability

Measures	CFA Loading	t-value	α
Product Quality			.893
• The overall quality of this OS product is better than that of competitors.	.891	14.02	
• The customers of this OS product often perceive this product to be better than competing products.	.749	10.77	
• The OS product meets customers' needs.	.767	11.13	
• Compared to other competitive products, this OS product is of better quality in terms of its functionality.	.797	11.78	
• Compared to other competitive products, this OS product is of better quality in terms of its reliability.	.781	11.42	
• Compared to other competitive products, this OS product is of better quality in terms of its security.	.578	7.68	
Product Development Speed			.792
• The OS product development for this product is faster than other open source software of similar nature.	.586	7.40	
• The OS product development for this product is faster than that of closed source software of similar nature.	.822	11.28	
• The OS product development for this product is faster than I had expected.	.727	9.65	
• The cycle time for this OS product development is shorter than that of similar software in the industry.	.639	8.23	
Product Creativity			.896
• Overall, this OS product is considered to be creative.	.716	10.01	
• This OS product challenges existing ideas for its product category.	.688	9.42	
• This OS product offers new ideas to its product category.	.772	11.09	
• This OS product is considered to be interesting.	.812	11.99	
• This OS product spawns new ideas for other products.	.779	11.26	
• This OS product encourages fresh thinking.	.804	11.80	
Perceived Benefit to Society			.787
▪ By contributing to this OS project, I make a contribution to society.	.765	10.28	
▪ By contributing to this OS project, I make a contribution to the open source community.	.852	11.71	
▪ This OS project benefits a large number of end-users.	.681	8.94	
Self-Fulfillment			.821
▪ I value highly the admiration and respect of my peers.	.802	11.15	
▪ I obtain self-fulfillment from open source contributions.	.610	8.33	
▪ Recognition from my peers validates my programming abilities.	.708	9.48	
▪ I feel gratified when my OS contribution is acknowledged.	.777	10.69	
Perceived Monetary Reward			.853
▪ By contributing to this project, I expect better job prospects in the future.	.910	13.44	
▪ By contributing to this project, I expect an increase in future salary.	.787	11.08	

▪ By contributing to this project, I am more likely to get freelance consulting jobs.	.718	9.86	
Political Convictions			.765
• Large software companies should not have as much power as they currently have.	.593	7.52	
• OS development promotes free and fair competition.	.710	9.91	
• The presence of a large OS community such as this one helps in limiting the power of large software companies.	.825	11.35	
• The presence of an OS community has given rise to more freedom and innovation in this project than there would be in proprietary software.	.602	7.66	
Developers' Contributions			.882
• I contributed a high volume of code to this open source project.	.779	11.44	
• I invested a large share of my time on this open source project.	.904	14.32	
• I exerted a great deal of effort in making contributions to this open source project	.919	14.71	
• Relative to other open source projects out there, I contributed more to this specific project.	.634	8.66	
Comparative Fit Index (CFI) = 0.945; Bollen Fit Index (IFI)=0.947, Root Mean Squared Error of App. (RMSEA)=0.043, $\chi^2 = 642.962$ with d.f. = 497.			

Table 2: Mean, Standard Deviation, and Correlation of the Constructs Used in the Study

Constructs	Correlations									
	Mean	Std	V1	V2	V3	V4	V5	V6	V7	V8
Product Quality (V1)	32.64	5.91	—							
Product Development Speed (V2)	17.61	4.61	.39	—						
Product Creativity (V3)	31.57	6.38	.43	.31	—					
Perceived Benefit to Society (V4)	18.31	3.23	.31	.25	.30	—				
Political Convictions (V5)	24.22	3.73	.25	.24	.13	.26	—			
Monetary Reward (V6)	12.61	4.20	.21	.15	.26	.08	-.14	—		
Self-Fulfillment (V7)	22.59	3.66	.20	.11	.27	.22	.09	.26	—	
Developers' Contribution (V8)	16.48	6.44	-.01	-.07	.18	.02	-.28	.19	.15	—

Table 3: Relationship between Incentives and Individual Developers' Contributions

	Developers' Contributions
Monetary Rewards	.29**
Self-fulfillment	.37*
Perceived Benefits to Society	0.01
Political Convictions	-.51**

Log-likelihood = -91.276

**p<0.05, *p<0.10

Table 4: Relationship between Incentives and Product Development

	Product Quality	Development Speed	Product Creativity
Monetary Rewards	0.34**	0.18*	0.32**
Benefits to Society	0.28*	0.22*	0.55**
Political Convictions	0.36**	0.29**	0.13
Self-fulfillment	0.07	0.07	0.26*
Development Stage	0.88**	0.47**	-----
Firm Expenditures	-0.001	0.002**	-----

**p<0.05, *p<0.10