Working for Free? Motivations for Participating in Open-Source Projects

Alexander Hars and Shaosong Ou

ABSTRACT: The success of the Linux operating system has demonstrated the viability of open-source software, an alternative form of software development that challenges traditional assumptions about software markets. Understanding why developers participate in open-source projects is crucial for assessing the impact of open-source software. Their motivations fall into two broad categories: internal factors (e.g., intrinsic motivation, altruism) and external rewards (e.g., expected future returns, personal needs). The results of a survey administered to open-source programmers are summarized.

KEY WORDS AND PHRASES: Altruism, human capital, intrinsic motivation, motivations, open-source software, software development.

The Linux open-source operating system is receiving a great deal of attention from software developers and users. Linux is touted as highly stable and reliable [13]. It has steadily increased its market share and brought about a consolidation of UNIX operating systems. Some commercial vendors have already taken extreme measures to counter the threat posed by open-source software. Sun, for example, has switched most of its Solaris operating system to an open-source license, eliminating a significant revenue stream. Compounding the issue, Linux is not an isolated phenomenon. Open-source software has become a viable alternative in many other software markets.

The open-source development model fundamentally changes the approaches and economics of traditional software development. Typically, open-source software is developed by an Internet-based community of programmers. Participation is voluntary, and participants do not receive direct compensation for their work. In addition, the entire source code is made available to the public. The developers devolve most of the property rights to the public, including the right to use, redistribute, and modify the software free of charge. This is a direct challenge to established assumptions about software markets and, as such, threatens the position of commercial software vendors.

The open-source phenomenon raises many interesting questions. Its proponents regard it as a paradigmatic change whereby the economics of private goods, built on the scarcity of resources, is replaced by the economics of public goods, where scarcity is not an issue. Critics argue that open-source software will always be relegated to niche areas, that it cannot compete with its commercial opponents in terms of product stability and reliability [15], and that open-source projects lack the capability to innovate. As commercial companies look for adequate responses to open-source software and legislators discuss its social implications, they need answers to a fundamental question: In the absence of direct compensation, what motivates the people who work for open-source projects? Is it true that open-source developers are altruists seeking to advance a good cause, or are there other explanations?

International Journal of Electronic Commerce / Spring 2002, Vol. 6, No. 3, pp. 25–39.

1086-4415/2002 $9.50 + 0.00.

History of Open-Source Software

The origin of open-source software can be traced back to the 1950s and 1960s, when software was sold together with hardware, and macros and utilities were freely exchanged in user forums. In the 1980s, as software was increasingly commercialized, Richard Stallmann, then a researcher at MIT, founded the Free Software Foundation (FSF), which provided a conceptual foundation for open-source software. While his "GNU Manifesto," which called for the development of a free alternative to UNIX (the NU stands for "not UNIX"), was criticized for its ideological baggage, his ideas are the basis of today's open-source movement [22]. Stallmann began a protracted community development effort called GNU, aiming to develop a free UNIX-like operating system. Although this effort was not successful, it led to the creation of an open-source infrastructure with tools and utilities on which subsequent open-source projects, such as Linux, could build. Today's Linux operating system is a mixture of software developed in the GNU project, a Linux kernel, and many additional components. Table 1 briefly summarizes the history of the development of open-source software.

Sources of Motivation

Much research has focused on discovering what motivates people. Maslow identified five needs that drive human activities, ranging from physiological needs to the need for self-actualization [16]. Deci emphasized the distinction between internal, psychological factors, which he called "intrinsic motivation," and external factors, which he called "external rewards" [6]. Intrinsic motivation refers to the desire to feel competent and self-determined. External rewards include such factors as direct or indirect monetary compensation, and recognition by others. The distinction between intrinsic motivation and external rewards is also emphasized by Herzberg, who views motivation as a function of three factors: ability of the individual over potential, ability over ability, and reinforcement behavior [11]. Similar distinctions are also made by Klandermans, who distinguishes social motivations, collective motivations, and reward motivations [14].

In the discussion that follows, the distinction between motivations rooted in the psychology of the individual (internal factors) and motivations that originate from the environment (external factors, rewards) will be used as the basis for identifying the factors that lead programmers to participate in open-source development projects.

Internal Factors

Proponents of open-source development emphasize the selflessness of open-source participants. They argue that open-source programmers are not motivated by monetary incentives but by their personal hobbies and preferences, or else by the rewarding sense that they are working to increase the welfare of others. Because these motivations are ultimately rooted within the individual, they are classified as internal factors.

Year	Event
1950s and 1960s	Software source code is distributed without restrictions in IBM and DEC user groups, ACM's Algorithms Section, etc.
1969	Ken Thompson writes the first version of UNIX. Its source code is distributed freely throughout the seventies.
1978	Donald Knuth (Stanford) publishes TEX as free software.
1979	Following AT&T's commercializing of UNIX, UC Berkeley begins creating its own version of UNIX, BSD (Berkeley Software Distribution). Eric Allmann, a student at UC Berkeley, develops a program that routes messages between computers over ARPANET. It later evolves into Sendmail.
1983	Richard Stallmann publishes GNU Manifesto calling for free software, and establishes Free Software Foundation.
1986	Larry Wall creates Perl (Practical Extraction and Report Language), a versatile programming language used for writing CGI (Common Gateway Interface) scripts.
1987	Developer Andrew Tanenbaum releases Minix, a version of UNIX for PC, Mac, Amiga, and Atari ST. It comes with complete source code.
1991	Linus Torvalds publishes version 0.02 of a new UNIX variant that he calls Linux in a Minix newsgroup.
1993	FreeBSD 1.0 is released. Based on BSD Unix, it includes networking, virtual memory, task switching, and large filenames. Ian Murdock creates a new Linux distribution called Debian Linux.
1994	Marc Ewing forms Red Hat Linux. It quickly becomes the leading Linux distributor. Bryan Sparks founds Caldera with backing from former Novell CEO Ray Noorda.
1995	The Apache Group builds a new Web server, Apache, based on HTTPd 1.3 of the National Center for Supercomputing Applications (NCSA) and a series of patch files. It has become the dominant HTTP server.
1998	Netscape not only gives away Communicator 5.0 (Mozilla) but releases its source code. Computer Associates, Corel, IBM, Informix, Interbase, Oracle, Sybase, and other major software vendors announce plans to port their products to Linux. Sun announces plans to release source code for Java 2 to developers.
1999	Number of Linux users estimated at 7.5 million.
2000	Novell, Real, and other software companies release versions of their products that run on Linux.

Table 1. Open-Source Time Line.

Sources: [5, 8, 10, 20, 22].

Intrinsic Motivation

There are certain activities and behaviors that people naturally like to engage in, such as playing games or collecting coins. Deci calls this intrinsic motivation, and in his view it arises from an inborn need to feel competent and to self-determine one's relations with one's environment [6]. This explains why people pursue their hobbies with such intensity. Maslow recognizes the same needs but groups them differently [16]. He makes a distinction between self-actualization needs and esteem needs, that is, the "desire for a stable, firmly based, usually high evaluation of [oneself]" [16, p. 21]. He divides esteem needs into two subsets, one of which is more internally focused, while the other includes the desires for recognition, fame, and reputation, which will be discussed in the section on external factors.

Applied to the open-source context, this category describes programmers as motivated by the feelings of competence, satisfaction, and fulfillment that arise from writing programs. One respondent to the survey summarized below, for example, described his motivation as: "Innate desire to code, and code, and code until the day I die."

Intrinsically motivated (i.e., autonomous) goals are associated with most effortful behaviors, as compared to controlled personal goals (i.e., nonintrinsically motivated goals), and thus lead to a greater possibility of goal attainment [21]. Since all behaviors can be regarded as goals [1], so can the behavior of participating in open-source projects. Presumably, then, open-source programmers with intrinsic motivations will spend more time and effort in open-source projects.

If the open-source movement were solely based on this motivation, it would be at a disadvantage in comparison to commercial development. The motivation of the developers is not necessarily linked to the needs of the users. In cases where the community of users and the community of programmers are not identical, open-source software would have an inherent problem in regard to incorporating user needs.

Altruism

Altruism is a variant of intrinsic motivation in which one seeks to increase the welfare of others. It is the personal disposition at the opposite pole from selfishness—"doing something for another at some cost to oneself" [18, p. 5]. Since open-source programmers provide something for others (writing programs with open-source code) at their own expense (time, energy, opportunity costs, etc.), they belong to this category. As with other intrinsic behaviors, altruism can be presumed to be an important drive that motivates open-source programmers to participate in open-source projects. Altruism is widely held to be associated with positive norms and, following the theory of reasoned action [2], should have a positive influence on the level of participation in open-source projects.

Community Identification

Another internal motivation, here labeled "community identification," is a variant of altruism. It corresponds to Maslow's needs for belonging and love. Programmers may identify themselves as members of the open-source community and align their goals with those of the community. They may treat other members of the community as kin and thus be willing to do something that is beneficial for them but not for themselves. Altruistic behavior of this type is called "kin-selection altruism" by social psychological researchers [12]. Programmers with this variant of intrinsic motivation will be motivated to participate in open-source projects and help their kinship partners.

External Rewards

Open-source programmers may also be motivated by external factors. While the vast majority of open-source programmers are not directly compensated for their contributions, they may obtain indirect rewards by increasing their marketability and skill base or by selling related products and services. Another type of external reward is related to the fruits of the software. These two categories are known, respectively, as "future rewards" and "personal needs."

Future Rewards

Open-source programmers may view participation as an investment from which they will obtain future returns [7, p. 13]. The economics of such investments are well understood, but it is necessary to ask what sort of returns are attractive to participants in the open-source community. Overall, there are four kinds of returns.

Revenues from Related Products and Services

Open-source software provides many opportunities for selling related products and services. In the case of Linux, individuals and companies like RedHat have begun to offer commercial consulting, training, distribution, support, and implementation services. The open-source community generally endorses such income-generating activities, but the boundaries are sometimes fuzzy. However, this motivation incurs an inherent conflict. Improving the open-source software may reduce the potential for selling its related services or products, such as maintenance and troubleshooting.

Human Capital

Programmers may also participate in open-source projects to expand their skill base. Economists use the term "human capital" to designate personal skills, capabilities, and knowledge. Increasing one's human capital by means of education, training, learning, and practicing leads to better job opportunities, higher salaries, and more fulfilling jobs [3]. The "open" source code and the freedom to choose their tasks enable the open-source programmers to select learning experiences that meet their needs and interests. They also enable entry-level programmers, such as college students, to participate in real projects.

Self-Marketing

Programmers may regard working for open-source projects as an effective way to demonstrate their capabilities and skills. Their achievements in open-source projects can be used to reinforce their claims to programming competence. Participating in open-source projects, therefore, is a good advertising channel for those seeking to advance in the programming field. Advertisement is also associated with future returns.

The self-marketing argument has an important implication. The larger someone's contribution to open-source projects, the more likely that commercial software vendors will recognize his or her value, thereby increasing the incentive to apply these skills in a paid position. Thus the openness of open-source software may actually work against itself to some extent, by helping to lure the best programmers and most productive minds away from these projects into more profitable commercial development.

Peer Recognition

Peer recognition derives from the desire for fame and esteem [16], which is also associated with future returns. As Raymond noted in his historic paper about open-source development, "The Cathedral and the Bazaar" [19], the success of open-source software is to a great extent due to its early, fast, and frequent releases. Similarly, open-source programmers receive rapid, constructive feedback about the quality of their work. Feedback always has a positive effect in that it shows programmers that people are using their contributions. Thus the feedback mechanism is self-reinforcing, for it encourages the author to expend additional effort to perfect his code, which in turn attracts more favorable feedback.

Personal Needs

As the history of open-source software shows, open-source projects are often initiated because a programmer has a personal need for a certain kind of software. For example, the programming language PERL was created by Larry Wall when he needed to generate Web pages programmatically. Finding it too cumbersome to write his programs in C, he began to develop simple routines that could be reused and combined [17, p. 194]. Later he shared these routines with other programmers, who extended and refined them. The development of the Apache Web server followed a similar pattern. In 1995, many Web masters were still using the NCSA Web server, circumventing its many problems by writing their own patches. A core group of Web masters soon formed to share patches. They rewrote the Web server to include more patches, and the Apache Web server was born [9]. In both cases the driving force for participating in an open-source project was the personal need of a programmer (or of several programmers) for specific software functionalities [4, p. 159].

The existence of personal needs has important implications. First, it shows that participants in open-source projects may act rationally in accordance with their own self-interest. But if selling the software involves significant transaction costs, they will provide it for free. Second, it shows that there is a limit to the amount of effort programmers are willing to provide for free. The more complex a product is, and the less dependent on other software modules, the more likely its value can be identified and communicated, and the more likely that programmers will sell their software rather than provide it for free. Some cases may already be cited. Eric Allmann, for example, the founder of Sendmail, one of the most successful e-mail server programs, has started a company that provides an add-on product to Sendmail to simplify its configuration and administration. This is a large module useful to most adopters of Sendmail and thus can be marketed effectively. The third implication of personal need may be the most important. It shows that the interests of users and developers are often aligned—both are interested in improving functionality, both are willing to invest in improvements. However, traditional software houses structure their license agreements in a way that prevents customers from investing in their software by making modifications and by sharing the improvements with others. This would raise the value of the license to prospective buyers and thus increase the revenue stream or market position of the software vendor. However, because of the fear of piracy, software houses have given up considerable potential investment opportunities that customers are willing to take. From the perspective of leveraging needs in improved functionality, this category of motivation demonstrates a crucial oversight in the marketing and product evolution strategies of software companies.

Empirical Analysis

To better understand participation in open-source projects, it is necessary to have firsthand information from actual programmers. Following upon the analysis of the components of motivation mentioned above, an invitation to participate in a Web-based survey was sent via e-mail to 389 persons involved in open-source projects. Their e-mail addresses were collected over the Internet from open-source discussion lists and news groups, including both general open-source communities and specific open-source programmers' forums. The survey asked specific questions about the factors that affected participation in open-source projects and exploratory questions about general aspects of open-source software. A list of the operationalization questions is included in the appendix. The survey received 81 responses. Since two were invalid because of missing data, the response rate was 21 percent.

Respondent Demographics

The vast majority of the respondents were male (95 percent), and most were between 20 and 40 years old. Although most of the respondents had college degrees or higher, about a quarter had not gone beyond high school or grammar school. Almost half of the responding open-source participants were professional programmers who earned their living as salaried or contract programmers (*see Figure 1*). Interestingly, a significant number of respondents (16 percent) were directly paid for their open-source programming. Their share of the reported effort was even higher: 38 percent of total working hours were contributed by paid open-source programmers who worked for commercial companies that supported the open-source movement. Besides these programmers, there were a large number of professional (salaried and contract) programmers who participated voluntarily without direct compensation. The rest comprised students (14 percent) and people whose hobby was programming (28 percent).

As Figure 2 shows, the majority of the open-source respondents were engaged in more than one project. The projects in which they were most active in are listed in Table 2. While Linux had the highest share, the table shows an impressive number of lesser-known projects.

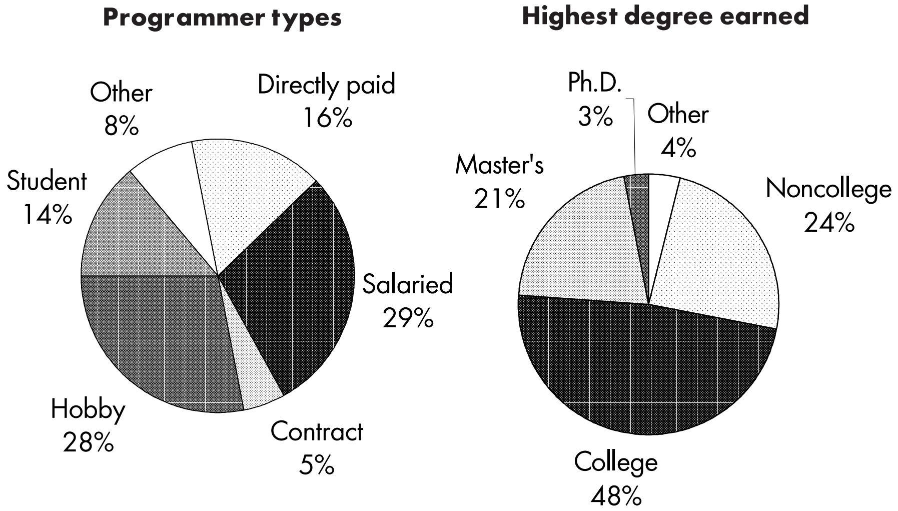

Figure 1. Respondent Demographic

Motivations

In the questionnaire, 16.5 percent of the respondents rated high on altruism. Almost 30 percent of the respondents identified strongly with the open-source community or had a kinlike relationship with other open-source programmers. Of the 79 valid responses, eleven (13.9 percent) selected "Selling related products or services" when asked why they participated in open-source projects. The fraction of human capital investors who aimed at improving their own human capital was remarkably high: 51 of the 79 respondents (70.9 percent) chose "Improving my programming skills" when asked why they participated in open-source projects. Several other respondents said they participated in order to learn skills other than programming that would improve their human capital. For example, one of them put "Learn English and teamwork" when asked the same question. In terms of self-marketing, when asked the motivation for participation, some respondents explicitly indicated that they participated in open-source projects because "it demonstrates my abilities" or "I can use it as a reference." More than half of the respondents (41 out of 79, 51.9 percent) selected "because I build a network of peers" as the reason for participating in open-source projects. Similar statements by other respondents included "hope to gain positive reputation" and "because it is a expression of personal liberty." The expectation about programming for personal needs was confirmed by responses like "Need the . . ." and "[My motivation is] to develop tools I need to do my job."

Table 3 shows the percentages of respondents who ranked high or very high on each of the eight motivation subcategories, together with the correlation coefficients of the subcategories with level of effort, as measured by the self-reported number of working hours/times spent in open-source projects each week. As can be seen, the participants ranked highest on the human capital and self-determination categories. The highest correlation coefficients

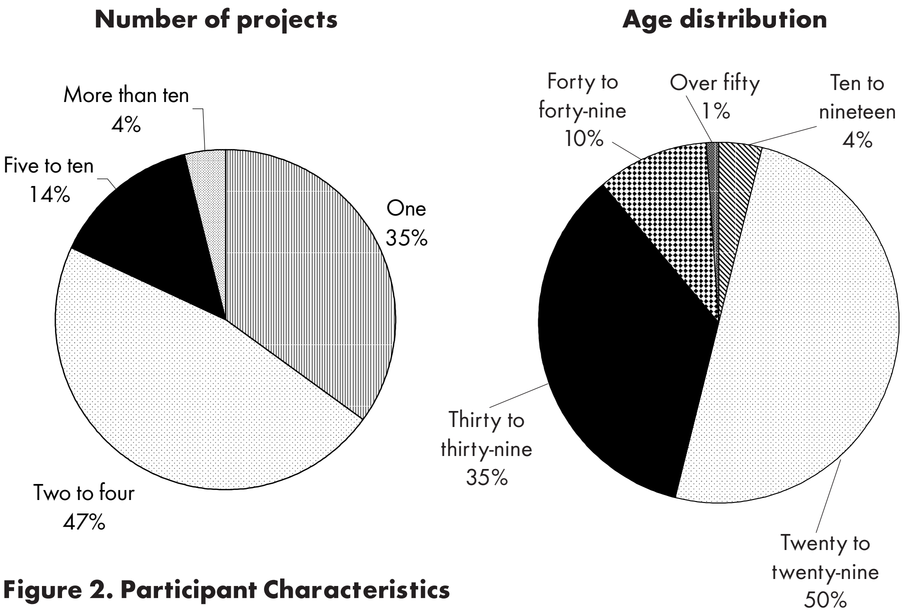

Figure 2. Participant Characteristics

Multiple occurrences	Single occurrences			
Linux (21)	AbiWord	Genes	LTSP	Tabindex
Midgard (4)	Analog	Gtk Explorer	MPLS for Linux	The COG Engine
Perl (3)	Cons	Hover Carnage	NetBSD	Vaxbb
GNU (3)	CPUlab	K2W RPG	OpenClassroom	Wftk
PhP (3)	Debian	Voyageur Immobile	Qwik Mail	WINE
Harbor Project (2)	Dents	Lesstif	Rasteroids	Xsu
Java (2)	Esky	Libdbg	SANE	
FreeBSD (2)	FreeCiv	Libsndfile	Smail3	
	Gdb	LTPlus	Sqwebmail	

Table 2. Respondents' Primary Open-Source Projects.

for external factors were selling products, self-marketing, and personal need. Among the internal factors, altruism correlated more strongly with effort than self-determination and community identification.

The patterns differed strongly, however, when different groups of programmers were considered. Students and hobby programmers rated higher than average on internal motivation and were also very much concerned about human capital, but external factors, especially human capital, were not highly correlated with level of effort. Students and hobby programmers seemed to be more strongly motivated by altruism and community identification.

The picture was different for salaried and contract programmers, who ranked higher than average on self-determination and personal need. Surprisingly, for this group determination and peer recognition needs correlated negatively with level of effort. The strongest positive correlation pointed to selling related services and products. Moreover (and this was true as well for the paid programmers), the need for peer recognition was negatively correlated with effort.

	All		Students and hobby programmers		Salaried and contract programmers		Programmers paid for open-source development	
Motivation	**Percent**	**Corr. with effort**	**Percent**	**Corr. with effort**	**Percent**	**Corr. with effort**	**Percent**	**Corr. with effort**
1. Internal								
Self-determination	79.7%	0.072	81.8%	–0.015	92.6%	–0.303	61.5%	0.221
Altruism	16.5%	0.192	24.2%	0.356	11.1%	0.061	7.7%	–0.163
Community identification	27.8%	0.116	36.4%	0.361	18.5%	–0.130	30.8%	–0.307
2. External								
2.1 Future rewards								
Selling products	13.9%	0.363	6.1%	0.011	3.7%	0.488	53.8%	0.304
Human capital	88.3%	0.139	96.9%	0.080	88.5%	0.073	84.6%	0.065
Self-marketing	36.7%	0.317	33.3%	0.206	29.6%	0.208	69.2%	0.424
Peer recognition	43.0%	–0.021	42.4%	–0.023	48.1%	–0.145	46.2%	–0.178
2.2 Personal need	38.5%	0.304	36.4%	0.301	38.5%	0.186	38.5%	0.328

Table 3. Motivations of Open-Source Programmers.

The third group, programmers who were paid for participating, differed markedly. They were stronger than average in terms of the desire to sell related products, self-marketing, and personal need. The strongest correlation with effort existed in self-marketing. Negative correlations existed for peer recognition and community identification. These results suggest that the different groups had very different motivations for participating in open-source projects.

The preliminary analysis with the correlation coefficients also suggests that external factors are more significant than the internal motivations so frequently advanced by proponents of the open-source movement. However, the analysis of correlations has notable shortcomings, as it only considers the apparent relationship between two variables. A further study, therefore, will examine the interdependencies between multiple variables, using multivariate analysis methods, such as factor analysis and structural equation modeling (SEM).

Conclusion

As the preceding discussion demonstrates, motivations for participating in open-source projects proved to be more complex than expected. While internal factors, such as intrinsic motivation, altruism, and identification with a community, played an important role, so did external factors, such as direct compensation and anticipated return. Factors that promised future monetary rewards, such as building human capital and self-marketing, were also more significant than expected. Personal need for a software solution was another key factor that has not yet received sufficient attention. The survey found that the developers who participated in open-source projects fell into several groups. Hobbyists and students were the most internally motivated. Salaried and contract programmers, in contrast, hoped to sell related products and services. A surprisingly large number of developers were paid for their open-source efforts. They were the ones most concerned with self-marketing and fulfilling their personal software needs. In light of these findings, it is evident that the open-source movement can draw from a diverse set of motivations, many of them based on external rewards. Thus the movement is poised to become a strong competitor to traditional software development.

REFERENCES

1. Ajzen, I. From intentions to actions: A theory of planned behavior. In J. Kuhl and J. Beckmann (eds.), *Action Control: From Cognition to Behavior.* Berlin: Springer-Verlag, 1985, pp. 267–275.

2. Ajzen, I., and Fishbein, M. *Understanding Attitudes and Predicting Social Behavior.* Englewood Cliffs, NJ: Prentice Hall, 1980.

3. Becker, G.S. Investment in human capital: A theoretical analysis. *Journal of Political Economy*, *70* supplement (1962), 9–49.

4. Behlendorf, B. Open source as a business strategy. In C. DiBona, S. Ockman, and T. Stone (eds.), *Open Sources*. Sebastopol, CA: O'Reilly, 1999, pp. 149–170.

5. Comerford, R. The path to open-source systems. *IEEE Spectrum, 36*, 5 (May 1999), 25–31.

6. Deci, E. *Intrinsic Motivation.* New York: Plenum Press, 1975.

7. DiBona, C.; Ockman, S.; and Stone, T. (eds.). *Open Sources.* Sebastopol, CA: O'Reilly, 1999.

8. Edwards, J. The changing face of freeware. *IEEE Computer, 31,* 10 (October 1998), 11–13.

9. Fielding, R.T. Shared leadership in the Apache project. *Communications of the ACM, 42,* 4 (April 1999), 42–43.

10. Gonzalez Barahona, J.; Heras Quiros, P.; and Bollinger, T. A brief history of free software and open source. *IEEE Software, 16,* 1 (January/February 1999), 32–33.

11. Herzberg, F. *The Managerial Choice.* Salt Lake City: Olympus, 1982.

12. Hoffman, M.L. Is altruism part of human nature? *Journal of Personality and Social Psychology, 40*, 1 (1981), 121–137.

13. Hollenbeck, D. Linux: Composability and reliability for the creative technologist. *IEEE Software, 16,* 1 (January/February 1999), 33.

14. Klandermans, B. *The Social Psychology of Protest.* Cambridge, MA: Blackwell, 1997.

15. Lewis, T. The open-source acid test. *IEEE Computer, 32*, 2 (February 1999), 125–128.

16. Maslow, A.H. *Motivation and Personality.* New York: Harper, 1987.

17. O'Reilly, T. Hardware, software and infoware. In C. DiBona, S. Ockman,

and T. Stone (eds.), *Open Sources*. Sebastopol, CA: O'Reilly, 1999, pp. 189–196.
18. Ozinga, J. *Altruism*. Westport, CT: Praeger, 1999.
19. Raymond, E. The cathedral and the bazaar. www.tuxedo.org/~esr/writings/cathedral-bazaar/cathedral-bazaar.html (10/20/1999).
20. Seltzer, L. Milestones in the open-source movement. *PC Magazine* (March 1999), www.zdnet.com/pcmag/features/opensource/390819.html.
21. Sheldon, K.M., and Elliot, A.J. Not all personal goals are personal: Comparing autonomous and controlled reasons for goals as predictors of effort and attainment. *Personality and Social Psychology Bulletin, 24*, 5 (May 1998), 94–105.
22. Stallmann, R. GNU manifesto. www.gnu.ai.mit.edu/gnu/manifesto.html, 1985.
23. Stallmann, R. Why software should be free. www.gnu.ai.mit.edu/philosophy/shouldbefree.html, 1992.

ALEXANDER HARS (hars@usc.edu) is an assistant professor of information systems, specializing in knowledge management, enterprise modeling, and applications of speech technologies, at the Marshall School of Business, University of Southern California. His research interests are knowledge-based analysis of information systems, business reference models, and conversation systems for requirements engineering. He is the founder of an innovative infrastructure for information systems research (cybrarium.usc.edu) that experiments with new approaches for electronic networks of scientific knowledge.

SHAOSONG OU (sou@marshall.usc.edu) is a Ph.D. student in the information and operations management department, Marshall School of Business, University of Southern California. He has an M.A. in economics from the University of Illinois, Chicago, and a B.A. in economics from Huazhong University of Science and Technology, China. His research interests include software development and architecture, systems of intelligent agents, knowledge management, communication technologies and organizational impacts, virtual organization management, and information-technology infrastructure in electronic commerce.

Appendix

Questions Used to Operationalize the Motivation Categories

General Issues

1. Why do you participate in open-source projects? Check all that apply.
   A. Programming is fun.
   B. It is a noble cause.
   C. I can change/extend the software to fit my specific needs.
   D. Expect to sell products or services related to it.
   E. Helps me improve my programming skills.
   F. Build a network of peers.
   G. I am paid to do this job.
   H. Other.

2. I spend most of my programming time as a:

A. Salaried programmer.
B. Contract programmer.
C. Hobby programmer.
D. Student.
E. Other.

3. Do you receive direct compensation (e.g., salary, contract) for your participation in the project?
A. Yes.
B. No.
(Measured using seven-point scale with extremes in the parentheses)

Intrinsic Motivation

1. Writing programs is fun. (strongly agree/strongly disagree)
2. I enjoy writing programs. (strongly agree/strongly disagree)
3. Programming gives me a chance to do the jobs I feel I do the best. (strongly agree/strongly disagree)
4. Participating in the project gives me a feeling of accomplishment. (strongly agree/strongly disagree)
5. Participating in the project gives me a feeling of competence. (strongly agree/strongly disagree)
6. Participating in the project gives me a feeling of effectiveness. (strongly agree/strongly disagree)
7. I rate my participation as an important activity for myself. (strongly agree/strongly disagree)

Extrinsic Rewards

1. I am paid to work for the project. (strongly agree/strongly disagree)
2. I receive some form of explicit compensation (e.g., salary, contract) for participating in the project. (strongly agree/strongly disagree)
3. For me, working for the project is: (extremely profitable/not profitable)
4. Comparing to other programming jobs, working for the project is: (very well paid/very poorly paid)

Personal Needs

1. How often do you use the software for yourself (excluding programming or testing activities)? (always/never)
2. The software is critical for my business or my work: (strongly agree/strongly disagree)
3. My participation in the open-source project ensures that the software provides functionality that matches my unique and specific needs. (strongly agree/strongly disagree)
4. It is hard for commercial software to meet my ever-changing needs. (strongly agree/strongly disagree)

5. Being able to fix problems with the software myself is one of the great advantages of open-source software. (strongly agree/strongly disagree)

Future Returns

1. Experience from the project raises my skill level of programming. (strongly agree/strongly disagree)
2. Because of my involvement in the project, I will be able to get a better job. (strongly agree/strongly disagree)
3. In one way or another I will make money from my participation in the project. (strongly agree/strongly disagree)
4. Participating in the project makes me more marketable. (strongly agree/strongly disagree)
5. I will sell products related to the project. (strongly agree/strongly disagree)
6. I will sell consulting, training, implementation or customization services related to the project. (strongly agree/strongly disagree)

Altruism

1. I don't care about money. (strongly agree/strongly disagree)
2. You can always trust an open-source programmer. (strongly agree strongly disagree)
3. Recognition from others is my greatest reward. (strongly agree/strongly disagree)
4. Open-source programmers should help each other out. (strongly agree/strongly disagree)
5. I deeply enjoy helping others—even if I have to make sacrifices. (strongly agree/strongly disagree)
6. Open-source programmers are a big family. (strongly agree/strongly disagree)
7. I am proud to be part of the open-source community. (strongly agree/strongly disagree)

Effort Level

1. Actually, how often do you work for the project? (more than once a day/not at all)
2. Actually, how many hours a week do you spend in the project? (more than 60/less than 5)

Demographic Characteristics

1. My gender is: (male/female)
2. The year of my birth is: (1940/ . . . /1990)
3. My highest educational degree is:

A. Grammar school.
B. High school.
C. Associate degree.
D. College degree.
E. Master's degree.
F. Doctoral degree.
G. Other.

4. My marital status is:
A. Single.
B. Married.
C. Separated.
D. Divorced.
E. Widowed.
F. Other.

5. My annual household income is: (up to $10,000/ . . . /over $100,000)

6. My primary occupation is:
A. Clerical/Administrative.
B. Craftsman/Craftswoman.
C. Educator/Full-time student.
D. Executive/Manager.
E. Factory operator/Laborer.
F. Homemaker.
G. Military.
H. Professional/Technical.
I. Retired/Not working.
J. Sales.
K. Self-employed.
L. Service worker.
M. Other.