1 Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects

Karim R. Lakhani and Robert G. Wolf

"What drives Free/Open Source software (F/OSS) developers to contribute their time and effort to the creation of free software products?" is a question often posed by software industry executives, managers, and academics when they are trying to understand the relative success of the F/OSS movement. Many are puzzled by what appears to be irrational and altruistic behavior by movement participants: giving code away, revealing proprietary information, and helping strangers solve their technical problems. Understanding the motivations of F/OSS developers is an important first step in determining what is behind the success of the F/OSS development model in particular, and other forms of distributed technological innovation and development in general.

In this chapter, we report on the results of a continuing study of the effort and motivations of individuals to contributing to the creation of Free/Open Source software. We used a Web-based survey, administered to 684 software developers in 287 F/OSS projects, to learn what lies behind the effort put into such projects. Academic theorizing on individual motivations for participating in F/OSS projects has posited that external motivational factors in the form of extrinsic benefits (e.g., better jobs, career advancement) are the main drivers of effort. We find, in contrast, that enjoyment-based intrinsic motivation—namely, how creative a person feels when working on the project—is the strongest and most pervasive driver. We also find that user need, intellectual stimulation derived from writing code, and improving programming skills are top motivators for project participation. A majority of our respondents are skilled and experienced professionals working in information technology–related jobs, with approximately 40 percent being paid to participate in the F/OSS project.

The chapter is organized as follows. We review the relevant literature on motivations and then briefly describe our study design and sample

characteristics. We then report our findings on payment status and effort in projects, creativity and motivations in projects, and the determinants of effort in projects. We conclude with a discussion of our findings.

Understanding Motivations of F/OSS Developers

The literature on human motivations differentiates between those that are intrinsic (the activity is valued for its own sake) and those that are extrinsic (providing indirect rewards for doing the task at hand) (Amabile 1996; Deci and Ryan 1985; Frey 1997; Ryan and Deci 2000). In this section we review the two different types of motivations and their application to developers in F/OSS projects.

Intrinsic Motivation

Following Ryan and Deci (2000, 56), "Intrinsic motivation is defined as the doing of an activity for its inherent satisfactions rather than for some separable consequence. When intrinsically motivated, a person is moved to act for the fun or challenge entailed rather than because of external prods, pressures, or rewards."[1] Central to the theory of intrinsic motivation is a human need for competence and self-determination, which are directly linked to the emotions of interest and enjoyment (Deci and Ryan 1985, 35). Intrinsic motivation can be separated into two distinct components: enjoyment-based intrinsic motivation and obligation/community-based intrinsic motivation (Lindenberg 2001). We consider each of them in the following sections.

Enjoyment-based Intrinsic Motivation Having fun or enjoying oneself when taking part in an activity is at the core of the idea of intrinsic motivation (Deci and Ryan 1985). Csikszentmihalyi (1975) was one of the first psychologists to study the enjoyment dimension. He emphasized that some activities were pursued for the sake of the enjoyment derived from doing them. He proposed a state of "flow," in which enjoyment is maximized, characterized by intense and focused concentration; a merging of action and awareness; confidence in one's ability; and the enjoyment of the activity itself regardless of the outcome (Nakamura and Csikszentmihalyi 2003). Flow states occur when a person's skill matches the challenge of a task. There is an optimal zone of activity in which flow is maximized. A task that is beyond the skill of an individual provokes anxiety, and a task that is below the person's skill level induces boredom. Enjoyable activities are found to provide feelings of "creative discovery, a challenge overcome

and a difficulty resolved" (Csikszentmihalyi 1975, 181). Popular accounts of programming in general and participation in F/OSS projects (Himanen 2001; Torvalds and Diamond 2001) in particular attest to the flow state achieved by people engaged in writing software. Thus F/OSS participants may be seeking flow states by selecting projects that match their skill levels with task difficulty, a choice that might not be available in their regular jobs.

Closely related to enjoyment-based intrinsic motivation is a sense of creativity in task accomplishment. Amabile (1996) has proposed that intrinsic motivation is a key determining factor in creativity. Amabile's definition of creativity consists of: (1) a task that is heuristic (no identifiable path to a solution) instead of algorithmic (exact solutions are known), and (2) a novel and appropriate (useful) response to the task at hand (Amabile 1996, 35). Creativity research has typically relied on normative or objective assessments of creativity with a product or process output judged creative by expert observers. Amabile (1996, 40), however, also allows for subjective, personal interpretations of creative acts. In particular, she proposes a continuum of creative acts, from low-level to high-level, where individual self-assessment can contribute to an understanding of the social factors responsible for creative output. Thus in our case, a F/OSS project dedicated to the development of a device driver for a computer operating system may not be considered terribly creative by outside observers, but may be rated as a highly creative problem-solving process by some individuals engaged in the project.

Obligation/Community-based Intrinsic Motivations Lindenberg (2001) makes the case that acting on the basis of principle is also a form of intrinsic motivation. He argues that individuals may be socialized into acting appropriately and in a manner consistent with the norms of a group. Thus the goal to act consistently within the norms of a group can trigger a normative frame of action. The obligation/community goal is strongest when private gain-seeking (gaining personal advantage at the expense of other group members) by individuals within the reference community is minimized. He also suggests that multiple motivations, both extrinsic and intrinsic, can be present at the same time. Thus a person who values making money and having fun may choose opportunities that balance economic reward (i.e., less pay) with a sense of having fun (i.e., more fun).

In F/OSS projects, we see a strong sense of community identification and adherence to norms of behavior. Participants in the F/OSS movement

exhibit strong collective identities. Canonical texts like *The New Hacker's Dictionary* (Raymond 1996), *The Cathedral and the Bazaar* (Raymond 2001), and the GNU General Public License (GPL) (Stallman 1999a) have created shared meaning about the individual and collective identities of the hacker[2] culture and the responsibilities of membership within it. Indeed, the term *hacker* is a badge of honor within the F/OSS community, as opposed to its pejorative use in popular media. The hacker identity includes solving programming problems, having fun, and sharing code at the same time. Private gain-seeking within the community is minimized by adherence to software licenses like the GPL and its derivatives, which allow for user rights to source code and subsequent modification.

Extrinsic Motivation

Economists have contributed the most to our understanding of how extrinsic motivations drive human behavior. "The economic model of human behavior is based on incentives applied from outside the person considered: people change their actions because they are induced to do so by an external intervention. Economic theory thus takes extrinsic motivation to be relevant for behavior" (Frey 1997, 13).

Lerner and Tirole (2002) posit a rational calculus of cost and benefit in explaining why programmers choose to participate in F/OSS projects. As long as the benefits exceed the costs, the programmer is expected to contribute. They propose that the net benefit of participation consists of immediate and delayed payoffs. Immediate payoffs for F/OSS participation can include being paid to participate and user need for particular software (von Hippel 2001a). Although the popular image of the F/OSS movement portrays an entirely volunteer enterprise, the possibility of paid participation should not be ignored as an obvious first-order explanation of extrinsic motivations. Firms might hire programmers to participate in F/OSS projects because they are either heavy users of F/OSS-based information technology (IT) infrastructure or providers of F/OSS-based IT solutions. In either case, firms make a rational decision to hire programmers to contribute to F/OSS projects.

Another immediate benefit relates to the direct use of the software product. Research on the sources of innovation has shown that users in general and lead users in particular have strong incentives to create solutions to their particular needs (von Hippel 1988). Users have been shown to be the source of innovations in fields as diverse as scientific instruments (Riggs and von Hippel 1994), industrial products (von Hippel 1988), sports equipment (Franke and Shah 2003), and library information systems (Mor-

rison, Roberts, and von Hippel 2000). Thus user need to solve a particular software problem may also drive participation in F/OSS projects.

Delayed benefits to participation include career advancement (job market signaling (Holmström 1999)) and improving programming skills (human capital). Participants indicate to potential employers their superior programming skills and talents by contributing code to projects where their performance can be monitored by any interested observer.[3] Similarly, firms looking for a particular skill in the labor market can easily find qualified programmers by examining code contributions within the F/OSS domain.

Participants also improve their programming skills through the active peer review that is prevalent in F/OSS projects (Moody 2001; Raymond 2001; Wayner 2000). Software code contributions are typically subject to intense peer review both before and after a submission becomes part of the official code base. Source code credit files and public e-mail archives ensure that faulty programming styles, conventions, and logic are communicated back to the original author. Peers in the project community, software users, and interested outsiders readily find faults in programming and often suggest specific changes to improve the performance of the code (von Krogh, Spaeth, and Lakhani 2003). This interactive process improves both the quality of the code submission and the overall programming skills of the participants.

Study Design and Sample Characteristics

Study Design

The sample for our survey was selected from among individuals listed as official developers on F/OSS projects hosted on the SourceForge.net F/OSS community Web site. At the start of our study period (fall 2001), SourceForge.net listed 26,245 active projects. The site requires project administrators to publicly characterize their project's development status (readiness of software code for day-to-day use) as planning, pre-alpha, alpha, beta, production/stable or mature. Projects that are in the planning or pre-alpha stage typically do not contain any source code and were eliminated from the population under study, leaving 9,973 available projects for the sample.

We conducted two separate but identical surveys over two periods. The first was targeted at alpha, beta, and production/stable projects and the second at mature projects. Because of the large number of alpha, beta and production/stable projects and the need to mitigate the effects of

self-selection bias, we selected a 10 percent random sample from those projects and extracted individual e-mails from projects that listed more than one developer.[4] Those led to 1,648 specific e-mail addresses and 550 projects. The second survey's sample was selected by obtaining the e-mail addresses of all participants in mature projects that were on multiple-person teams. This procedure identified 103 projects (out of 259) with 573 unique individuals (out of 997).

We collected data through a Web-based survey. We sent personalized e-mails to each individual in our sample, inviting him or her to participate in the survey. Each person was assigned a random personal identification number (PIN) giving access to the survey. Respondents were offered the opportunity to participate in a random drawing for gift certificates upon completion of the survey.

The first survey ran from October 10 to October 30, 2001. During this time, 1,530 e-mails reached their destinations and 118 e-mails bounced back from invalid accounts. The survey generated 526 responses, a response rate of 34.3 percent. The second survey ran from April 8 to April 28, 2002. Of the 573 e-mails sent, all e-mails reached their destinations. The second survey generated 173 responses for a response rate of 30.0 percent. Close examination of the data revealed that 15 respondents had not completed a majority of the survey or had submitted the survey twice (hitting the send button more than once). They were eliminated from the analysis. Overall, the survey had 684 respondents from 287 distinct projects, for an effective response rate of 34.3 percent. The mean number of responses per project was 4.68 (standard deviation (sd) = 4.9, median = 3, range = 1–25).

Who Are the Developers?

Survey respondents were primarily male (97.5 percent) with an average age of 30 years[5] and living primarily in the developed Western world (45 percent of respondents from North America (U.S. and Canada) and 38 percent from Western Europe). Table 1.1 summarizes some of the salient characteristics of the sample and their participation in F/OSS projects.

The majority of respondents had training in IT and/or computer science, with 51 percent indicating formal university-level training in computer science and IT. Another 9 percent had on-the-job or other related IT training. Forty percent of the respondents had no formal IT training and were self taught.

Overall, 58 percent of the respondents were directly involved in the IT industry, with 45 percent of respondents working as professional pro-

Table 1.1
General characteristics of survey respondents

Variable	Obs	Mean	Std. Dev.	Min	Max
Age	677.00	29.80	7.95	14.00	56.00
Years programming	673.00	11.86	7.04	1.00	44.00
Current F/OSS projects	678.00	2.63	2.14	0.00	20.00
All F/OSS projects	652.00	4.95	4.04	1.00	20.00
Years since first contribution to F/OSS community	683.00	5.31	4.34	0.00	21.00

Table 1.2
Location and work relationship for F/OSS contributions

Is supervisor aware of work time spent on the F/OSS project?	Freq.	Percent
Yes aware	254	37.69
No, not aware	113	16.77
Do not spend time at work	307	45.55
Total	674	100.00

grammers and another 13 percent involved as systems administrators or IT managers. Students made up 19.5 percent of the sample and academic researchers 7 percent. The remaining respondents classified their occupation as "other." As indicated by table 1.1, on average the respondents had 11.8 years of computer programming experience.

Payment Status and Effort in Projects

Paid Participants

We found that a significant minority of contributors are paid to participate in F/OSS projects. When asked if they had received direct financial compensation for participation in the project, 87 percent of all respondents reported receiving no direct payments. But, as table 1.2 indicates, 55 percent contributed code during their work time. When asked: "if a work supervisor was aware of their contribution to the project during work hours," 38 percent of the sample indicated supervisor awareness (explicit or tacit consent) and 17 percent indicated shirking their official job while

working on the project. The sum of those who received direct financial compensation and those whose supervisors knew of their work on the project equals approximately 40 percent of the sample, a category we call "paid contributors." This result is consistent with the findings from other surveys targeting the F/OSS community (Hars and Ou 2002; Hertel, Niedner, and Herrmann 2003).

Effort in Projects

We measure effort as the number of hours per week spent on a project. This measure has been used in previous F/OSS studies (Hars and Ou 2002; Hertel, Niedner, and Herrmann 2003) and provides an appropriate proxy for participant contribution and interest in F/OSS projects. Survey respondents were asked how many hours in the past week they had spent working on all their current F/OSS projects in general and "this project" (the focal project about which they were asked motivation questions) in particular. Respondents said that they had, on average, spent 14.1 hours (sd = 15.7, median = 10, range 0–85 hours) on all their F/OSS projects and 7.5 hours (sd = 11.6, median = 3, range 0–75 hours) on the focal project. The distribution of hours spent was skewed, with 11 percent of respondents not reporting any hours spent on their current F/OSS projects and 25 percent reporting zero hours spent on the focal project. Table 1.3 indicates that paid contributors dedicate significantly more time (51 percent) to projects than do volunteers.

Overall, paid contributors are spending more than two working days a week and volunteer contributors are spending more than a day a week on F/OSS projects. The implied financial subsidy to projects is substantial. The

Table 1.3
Hours/week spent on F/OSS projects

	Average (sd)	Paid contributor (sd)	Volunteer (sd)	*t* statistic (*p*-value)*
Hours/week on all F/OSS projects	14.3 (15.7)	17.7 (17.9)	11.7 (13.5)	4.8 (0.00)
Hours/week on focal F/OSS project	7.5 (11.6)	10.3 (14.7)	5.7 (8.4)	4.7 (0.00)

* Two-tailed test of means assuming unequal variances
Note: $n = 682$.

Table 1.4
Creativity in F/OSS projects

Compared to your most creative endeavour, how creative is this project?	Freq.	Percent
Much less	55	8.16
Somewhat less	203	30.12
Equally as creative	333	49.41
Most creative	83	12.31
Total	674	100.00

2001 United States Bureau of Labor Statistics wage data[6] indicated mean hourly pay of $30.23 for computer programmers. Thus the average weekly financial contribution to F/OSS projects is $353.69 from volunteers and $535.07 from paid contributors (via their employers).

Creativity and Motivation in Projects

Creativity and Flow

Respondents noted a very high sense of personal creativity in the focal projects. They were asked: "imagine a time in your life when you felt most productive, creative, or inspired. Comparing your experience on this project with the level of creativity you felt then, this project is. . . ." More than 61 percent of our survey respondents said that their participation in the focal F/OSS project was their most creative experience or was equally as creative as their most creative experience. Table 1.4 describes the response patterns. There was no statistical difference between the responses provided by paid and volunteer developers.

It may seem puzzling to nonprogrammers that software engineers feel creative as they are engaged in writing programming code. As Csikszentmihalyi (1975; 1990; 1996) has shown, however, creative tasks often cause participants to lose track of time and make them willing to devote additional hours to the task, a psychological state he calls "flow." It appears that our respondents may experience flow while engaged in programming. Table 1.5 indicates that 73 percent of the respondents lose track of time "always" or "frequently" when they are programming and more than 60 percent said that they would "always" or "frequently" dedicate one additional hour to programming ("if there were one more hour in the day").

Table 1.5
"Flow" experienced while programming

Ratings on "flow" variables	How likely to lose track of time when programming (%)	How likely to devote extra hour in the day to programming (%)
Always	21.39	12.92
Frequently	51.33	47.14
Sometimes	22.27	34.51
Rarely	4.28	4.11
Never	0.74	1.32
Total	100	100

Note: $n = 682$.

Again, there was no significant statistical difference between the answers provided by volunteers and paid contributors.

Motivations to Contribute

Table 1.6 provides a ratings breakdown of the motivations to contribute to the focal F/OSS project. Respondents were asked to select up to three statements (the table shows the exact wording used in the survey) that best reflected their reasons for participating and contributing to "this" project. As discussed in the literature review, motivations can be put into three major categories: (1) enjoyment-based intrinsic motivations, (2) obligation/community-based intrinsic motivations, and (3) extrinsic motivations. We find evidence for all three types of motivations in F/OSS projects.

User needs for the software, both work- and nonwork-related, together constitute the overwhelming reason for contribution and participation (von Hippel 1988; 2001a; 2002; 2005), with more than 58 percent of participants citing them as important. But, since we asked separate questions about work- and nonwork-related user needs, we also report that 33.8 percent of participants indicated work-related need and 29.7 percent participants indicated nonwork-related need as a motive for participation. Less than 5% of respondents rated both types of user needs as important.[7]

The top single reason to contribute to projects is based on enjoyment-related intrinsic motivation: "Project code is intellectually stimulating to write" (44.9 percent). This result is consistent with our previous findings regarding creativity and flow in projects. Improving programming skills, an extrinsic motivation related to human capital improvement, was a

Table 1.6
Motivations to contribute to F/OSS projects

Motivation	Percentage of respondents indicating up to three statements that best reflect their reasons to contribute	Percentage of volunteer contributors	Percentage of paid contributor	Significant difference (*t* statistic/*p* value)
Enjoyment-based intrinsic motivation				
Code for project is intellectually stimulating to write	44.9	46.1	43.1	n.s.
Economic/extrinsic-based motivations				
Improve programming skills	41.3	45.8	33.2	3.56 ($p = 0.0004$)
Code needed for user need (work and/or nonwork)*	58.7	—	—	—
Work need only	33.8	19.3	55.7	10.53 ($p = 0.0000$)
Nonwork need	29.7	37.0	18.9	5.16 ($p = 0.0000$)
Enhance professional status	17.5	13.9	22.8	3.01 ($p = 0.0000$)
Obligation/community-based intrinsic motivations				
Believe that source code should be open	33.1	34.8	30.6	n.s.
Feel personal obligation to contribute because use F/OSS	28.6	29.6	26.9	n.s.

Table 1.6
(continued)

Like working with this development team	20.3	21.5	18.5	n.s.
Dislike proprietary software and want to defeat them	11.3	11.5	11.1	n.s.
Enhance reputation in F/OSS community	11.0	12.0	9.5	n.s.

Notes: Aggregation of responses that indicated needing software for work and/or nonwork-related need. Not an actual survey question. Overlap in user needs limited to 4.9 percent of sample.
n.s. = not significant, $n = 679$.

close second, with 41.8 percent of participants saying it was an important motivator.

Approximately one-third of our sample indicated that the belief that "source code should be open," an obligation/community motivation, was an important reason for their participation. Nearly as many respondents indicated that they contributed because they felt a sense of obligation to give something back to the F/OSS community in return for the software tools it provides (28.6 percent). Approximately 20 percent of the sample indicated that working with the project team was also a motivate for their contribution. Motivations commonly cited elsewhere, like community reputation, professional status, and defeating proprietary software companies (Raymond 2001; Lerner and Tirole 2002), were ranked relatively low.

Another source of an obligation/community motivation is the level of identification felt with the hacker community. Self-identification with the hacker community and ethic drive participation in projects. Respondents to our survey indicated a strong sense of group identification, with 42 percent indicating that they "strongly agree" and another 41 percent "somewhat agree" that the hacker community is a primary source of their identity.[8] Nine percent of the respondents were neutral and 8 percent were somewhat to strongly negative about the hacker affiliation.[9]

Table 1.6 also indicates significant differences in motivations between paid contributors and volunteers. The differences between the two groups are consistent with the roles and requirements of the two types of F/OSS participants. Paid contributors are strongly motivated by work-related user need (55.7 percent) and value professional status (22.8 percent) more than volunteers. On the other hand, volunteers are more likely to participate because they are trying to improve their skills (45.8 percent) or need the software for nonwork purposes (37%).

To better understand the motives behind participation in the F/OSS community, and the reason that no one motivation, on its own, had more than 50% importance, we decided to do an exploratory cluster analysis to see whether there were any natural groupings of individuals by motivation type. We used *k*-means cluster analysis, with random seeding. The four-cluster solution provided the best balance of cluster size, motivational aggregation, stability, and consistency and is presented in table 1.7. The motivations that came out highest in each cluster have been highlighted.

Cluster membership can be explained by examining the motivation categories that scored the highest in each cluster. Cluster 3 (29 percent of the

Table 1.7
Cluster results based on motivations and paid status

Motivations	Cluster 1 (%)	Cluster 2 (%)	Cluster 3 (%)	Cluster 4 (%)
Work need	**91**	8	12	28
Nonwork need	11	**100**	0	2
Intellectually stimulating	41	45	**69**	12
Improves skill	20	43	**72**	19
Work with team	17	16	**28**	19
Code should be open	12	22	42	**64**
Beat proprietary software	11	8	9	**19**
Community reputation	**14**	8	11	13
Professional status	**25**	6	22	18
Obligation from use	23	20	6	**83**
Paid for contribution	**86**	18	26	32
Total percentage of sample in each cluster	25	27	29	19

Note: $n = 679$.

sample) consists of individuals who contribute to F/OSS projects to improve their programming skills and for intellectual stimulation. None of the members of this cluster noted nonwork-related need for the project and very few, 12 percent, indicated work-related need for the code. Members of this group indicated an affinity for learning new skills and having fun in the process. The actual end product does not appear to be a large concern; both enjoyment-based intrinsic motivation and career-based extrinsic motivation are important to this group.

All members of cluster 2 (27 percent of the sample) indicate that nonwork-related need for the code is an important motive for their participation. The primary driver for this group is extrinsic user need. Similarly, cluster 1 (25 percent of the sample) represents individuals who are motivated by work-related need with a vast majority (86 percent) paid for their contributions to F/OSS projects. This cluster can also be thought of as composed of people with extrinsic motivations. Cluster 4 (19 percent of the sample) consists of people motivated primarily by obligation/community-based intrinsic motivations. A majority of this cluster report group-identity-centric motivations derived from a sense of obligation to the community and a normative belief that code should be open.

The cluster analysis clearly indicates that the F/OSS community is heterogeneous in motives to participate and contribute. Individuals join for a variety of reasons, and no one reason tends to dominate the community or to cause people to make distinct choices in beliefs. These findings are consistent with collective action research, where group heterogeneity is considered an important trait of successful social movements (Marwell and Oliver 1993).

Determinants of Effort

Our findings so far have confirmed the presence of all three types of motivations, with no clear and obvious determinants of effort. We do note that paid contributors work more hours than volunteers. Given that there were not that many significant differences in motivations between paid and volunteer contributors, though, we are left with an open question regarding the effect the types of motivation (intrinsic vs. extrinsic) on effort in projects. To address the question, we ran an ordinary least squares (OLS) regression on the log of hours/week[10] dedicated to the focal project.

Table 1.8 presents the standardized[11] values of the coefficients of significant variables in the final regression. A personal sense of creativity with a F/OSS project has the largest positive impact on hours per week. Being

Table 1.8
Significant variables in regression of log (project hours/week) and motivations

Variable	Standardized coefficient	*t*-statistic (*p*-value)
Creative project experience	1.6	6.00 (0.000)
Paid status	0.88	3.12 (0.002)
Like team	0.84	2.76 (0.004)
Enhance community reputation	0.56	2.00 (0.046)
Differential hours	–1.6	–6.00 (0.000)
IT training	–0.6	–2.28 (0.023)

Note: *r*-Square = 0.18, $n = 630$.

paid to write code and liking the team have significant positive effects that are approximately half the weight of a sense of creativity. Caring about reputation in the F/OSS community has about one-third the impact as feeling creative with a project. Number of hours dedicated to other F/OSS projects has a negative impact equal to that of creativity on the current project. We can see that various F/OSS projects compete for time, and that distractions from other projects can reduce the hours spent on the focal project. Having formal IT training also reduces the number of hours spent on a project.

As mentioned in the literature review, proponents of intrinsic motivation theories have assembled an impressive array of experimental evidence to demonstrate that extrinsic rewards have a negative impact on intrinsic motivations. An obvious test in our study is to examine the impact of the interaction between being paid and feeling creative on the number of hours per week dedicated to a project. Regression analysis showed that there was no significant impact on the hours per week dedicated due to the interaction of being paid and feeling creative. Hours per week dedicated to a project did not decline, given that those who are paid to contribute code also feel creative about that project.

Researchers engaged in studying creativity have traditionally used third-party assessments of innovative output as measures of creativity. Thus our finding that a sense of personal creativity is the biggest determinant of effort in F/OSS projects may be due to the inherent innovative nature of the project itself and not to personal feelings of creativity. Since we have multiple responses from many projects, we can test whether the creativity felt is endogenous to the project or to the individual. Results from a fixed-effects

regression (Greene 2000) showed that a personal sense of creativity in a project is still positive and significant, indicating that the sense of creativity is endogenous and heterogeneous to the people within projects.

Discussion

The most important findings in our study relate to both the extent and impact of the personal sense of creativity developers feel with regard to their F/OSS projects. A clear majority (more than 61 percent) stated that their focal F/OSS project was at least as creative as anything they had done in their lives (including other F/OSS projects they might have engaged in). This finding is bolstered by the willingness of a majority of survey participants to dedicate additional hours to programming, and, consistent with attaining a state of flow, frequently losing track of time while coding. These observations are reinforced by the similar importance of these creativity-related factors for both volunteer and paid contributors.

The importance of the sense of creativity in projects is underscored by examination of the drivers of effort in F/OSS projects. The only significant determinants of hours per week dedicated to projects were (in order of magnitude of impact):

- Enjoyment-related intrinsic motivations in the form of a sense of creativity
- Extrinsic motivations in form of payment
- Obligation/community-related intrinsic motivations

Furthermore, contrary to experimental findings on the negative impact of extrinsic rewards on intrinsic motivations (Deci, Koestner, and Ryan 1999), we find that being paid and feeling creative about F/OSS projects does not have a significant negative impact on project effort.

Therefore, work on the F/OSS projects can be summarized as a creative exercise leading to useful output, where the creativity is a lead driver of individual effort.

Programming has been regarded as a pure production activity typified as requiring payments and career incentives to induce effort. We believe that this is a limited view. At least as applied to hackers on F/OSS projects, activity should be regarded as a form of joint production–consumption that provides a positive psychological outlet for the participants as well as useful output.

Another central issue in F/OSS research has been the motivations of developers to participate and contribute to the creation of a public good.

The effort expended is substantial. Individuals contribute an average of 14 hours per week. But there is no single dominant explanation for an individual software developer's decision to participate in and contribute to a F/OSS project. Instead, we have observed an interplay between extrinsic and intrinsic motivations: neither dominates or destroys the efficacy of the other. It may be that the autonomy afforded project participants in the choice of projects and roles one might play has "internalized" extrinsic motivations.

Therefore, an individual's motivation containing aspects of both extrinsic and intrinsic is not anomalous. We have observed clusters of individuals motivated by extrinsic, intrinsic, or hybrid extrinsic/intrinsic factors. Dominant motives do not crowd out or spoil others. It is consistent for someone paid to participate in the F/OSS movement to be moved by the political goals of free software and open code.

Other issues merit further investigation. The presence of paid participants—40 percent of our study sample—indicates that both IT-producing and IT-using firms are becoming important resources for the F/OSS community. The contribution of firms to the creation of a public good raises questions about incentives to innovate and share innovations with potential competitors. In addition, the interaction between paid and volunteer participants within a project raises questions about the boundaries of the firm and appropriate collaboration policies.

In conclusion, our study has advanced our understanding of the motivational factors behind the success of the F/OSS community. We note that the F/OSS community does not require any one type of motivation for participation. It is a "big tent." Its contributors are motivated by a combination of intrinsic and extrinsic factors with a personal sense of creativity being an important source of effort.

Notes

We would like to thank the developers on the SourceForge.net F/OSS projects for being so generous with their time while answering our survey. We would also like to thank the following colleagues for their helpful comments and feedback during the early versions of this chapter: Jeff Bates, Jim Bessen, Paul Carlile, Jonathon Cummings, Joao Cunha, Chris DiBona, Jesper Sorensen, and Eric von Hippel. The following colleagues at BCG were extremely helpful during the study: Mark Blaxill, Emily Case, Philip Evans and Kelly Gittlein. Mistakes and errors remain ours.

1. The subject of intrinsic motivation has been well studied in psychology; for reviews see Deci, Koestner, and Ryan (1999) and Lindenberg (2001).

2. *Hacker* as in *The New Hacker's Dictionary* (Raymond 1996): "hacker: n. [originally, someone who makes furniture with an axe] 1. A person who enjoys exploring the details of programmable systems and how to stretch their capabilities, as opposed to most users, who prefer to learn only the minimum necessary. 2. One who programs enthusiastically (even obsessively) or who enjoys programming rather than just theorizing about programming. 3. A person capable of appreciating hack value. 4. A person who is good at programming quickly. 5. An expert at a particular program, or one who frequently does work using it or on it; as in "a Unix hacker." (Definitions 1 through 5 are correlated, and people who fit them congregate.) 6. An expert or enthusiast of any kind. One might be an astronomy hacker, for example. 7. One who enjoys the intellectual challenge of creatively overcoming or circumventing limitations. 8. [deprecated] A malicious meddler who tries to discover sensitive information by poking around. Hence "password hacker," "network hacker." The correct term for this sense is *cracker*.

3. The widespread archiving of all F/OSS project-related materials like e-mail lists and code commits enables a detailed assessment of individual performance.

4. The "greater than one developer" criteria was used to ensure selection of projects that were not "pet" software projects parked on SourceForge.net, but rather projects that involved some level of coordination with other members.

5. At time of study.

6. Available at http://www.bls.gov/oes/2001/oes_15Co.htm, accessed April 2, 2003.

7. A detailed examination of the difference in project types between those that stated work-related needs and those that stated nonwork-related needs showed that there was no technical difference between them. A majority of the projects that were described as nonwork were of sufficient technical scope and applicability that firms also produced similar proprietary versions. We therefore see a blurring of distinction in the software produced for work and nonwork purposes. The general-purpose nature of computing and software creates conditions such that a similar user need can be high in both work and nonwork settings.

8. Respondents were given the definition of "hacker" in note 2 when asked the question about identity.

9. The results were identical when controlled for paid contributor status on a project.

10. We chose to use the log of project hours/week because of the skewness in the reported data. A log transformation allows us to better represent the effects of small changes in the data at the lower values of project hours/week. It is safe to argue that there is a significant difference between 4 versus 8 project hours/week and 25 versus 29 project hours/week. The magnitude of the effort expended is much greater at the

lower values of the measure and the log transformation allows us to capture this shift. Since the log of zero is undefined, all zero values were transformed to 0.00005, giving us the desired impact for a very small and insignificant value.

11. Standardizing the variables to allows us to make comparison across all motivation factors, since the original variables had different underlying values. All variables in the regression were transformed so that the mean = 0 and the variance = 1.