12-31-2002

Economic Incentives for Participating in Open Source Software Projects

Il-Horn Hann
University of Southern California

Jeff Roberts
Carnegie Mellon University

Sandra Slaughter
Carnegie Mellon University

Roy Fielding
Apache Software Foundation

ECONOMIC INCENTIVES FOR PARTICIPATING IN OPEN SOURCE SOFTWARE PROJECTS

Il-Horn Hann
Marshall School of Business
University of Southern California
Los Angeles, CA USA
hann@marshall.usc.edu

Jeff Roberts
Graduate School of Industrial Administration
Carnegie Mellon University
Pittsburgh, PA USA
jroberts@andrew.cmu.edu

Sandra Slaughter
Graduate School of Industrial Administration
Carnegie Mellon University
Pittsburgh, PA USA
sandras@andrew.cmu.edu

Roy Fielding
Chairman
Apache Software Foundation (ASF)
fielding@apache.org

Abstract

Using the Internet as a basis for communication, collaboration, and storage of artifacts, the open source community is producing software of a quality that was previously thought to be achievable only by professional engineers following strict software development paradigms. This accomplishment is even more astounding as developers contribute to the source code without any remuneration. Open source leaders as well as academics have proposed theories about the motivation of open source developers that are rooted in diverse fields such as social psychology and anthropology. However, Lerner and Tirole (2000) argue that developer participation in open source projects may, in part, be explained by existing economic theory regarding career concerns. This research seeks to confirm or disconfirm the existence of economic returns to participation in open source development. Our findings suggest that greater open source participation per se*, as measured in contributions made, is not associated with wage increases. However, a higher status in a merit-based ranking within the Apache Project is associated with significantly higher wages. This suggests that employers do not reward the gain in experience through open source participation as an increase in human capital. The results are also consistent with the notion that a high rank within the Apache Software Foundation is a credible signal of the productive capacity of a programmer.*

1 INTRODUCTION

Open source software development, i.e., public software development projects where participants can read, modify, and redistribute the software source code (OSI 2001), is arguably one of the most exciting phenomena in the software industry today. Open source has played a fundamental role in the development of the Internet by contributing to such remarkable software as TCP/IP, BIND, Sendmail, Linux, and Apache. The open source community has harnessed the Internet like no other by making it the critical piece of its communication and collaboration infrastructure. This *prima facie* simple innovation has resulted in a revolutionary organization of software production and has sparked discussion on a wide variety of issues, ranging from software development, information architecture, and standards as well as incentives and intellectual property rights.

One widely debated question is why open source programmers contribute voluntarily, thereby foregoing any direct remuneration that they could accrue while working on a commercial system. Often-quoted individual level motivations for participating in open source development projects cover a broad spectrum including scratching a "personal itch" with respect to software functionality, enjoyment, and desire to be "part of a team" (Ghosh 1998; O'Reilly 2000; Raymond 2000c). Others liken the open source community to a gift culture where the status of a participant depends on "what he gives away" (Raymond 2000b). Alternatively, Lerner and Tirole (2000) suggest that open source participation may in part be explained by existing theories of labor economics.

In this paper, we investigate whether economic incentives are associated with open source participation. There are two possible explanations. One explanation is based on human capital theory. In this view, open source software projects offer a learning opportunity to software developers. A wider or deeper knowledge base is then subsequently rewarded by employers. Alternatively, participation in open source projects itself can work as a signal of the programmer's hidden desirable traits to current or future employers. For example, a successful participation in open source projects may allow employers to screen programmers with desirable characteristics such as the ability to convince others about a design decision or to motivate other programmers.

In the following section, we briefly describe previous literature, discussing economic and non-economic factors that might lead to open source participation. This is followed by the argument for delayed incentives in open source participation in section 3. Section 4 presents an overview of our research site, the organization that governs the development process, as well as the data that we collected. In section 5, we develop the model used to estimate the delayed returns on open source participation and describe the data set used. The results are presented in section 6 and we conclude in section 7.

2 LITERATURE REVIEW

The open source movement has been of great interest for academics. Researchers with diverse backgrounds such as computer science, psychology, sociology, and economics have started to investigate the topic, making open source development a truly interdisciplinary research field. The first works in this rapidly developing field were often descriptive in nature (Raymond 2000c) followed by theory driven explanations (Lerner and Tirole 2000) and early empirical research (Koch and Schneider 2002; Lakhani and von Hippel 2000; Mockus et al. 2000). Many of the early explorations into the inner workings of the open source development process have sought to explain the mechanisms by which open source projects produce such seemingly high quality software (Fitzgerald and Feller 2001; Raymond 2000a). Of keen interest to the software engineering community is the stark contrast between open source methods and methods generally accepted as conventional software engineering wisdom. For example, in contrast to conventional practice, the open source development style seems to forgo a formal specification and design phase. Further, the parallel programming and review efforts seemingly circumvent Brooks' Law (Brooks 1995), leading to greater reliability and quality.

Motivations for open source participation have been explained from various theoretical perspectives including social psychological, cultural, or economic motivations. Eric Raymond, an evangelist of the open source movement, popularized social psychological or cultural explanations of open source participation. In the cultural view, the open source community's truly valuable and protection-worthy property is the ownership of ideas or programming projects. Given the abundance of resources, i.e., computing power, bandwidth, and disk space, social status is determined not by what you have, but what you give away. This leads to the "gift" culture, where the reputation of a programmer is primarily determined by his free contributions (Raymond 2000b). As a second explanation, Raymond (2000b) offers a "craftsmanship" model where the artisan aspects of programming motivate developers to create works to be admired not only by themselves but also by others. In both cases developers are motivated through the recognition of their contributions by their peers. Such an explanation finds theoretical support in social psychology (Clary et al. 1998; Mauss 1967).

From an economic perspective, a programmer will choose to contribute to an open source project if the benefits outweigh the costs of participation. The primary costs come in the form of opportunity costs for the time spent that could have been otherwise allocated to new or existing projects. Benefits can be categorized as immediate or delayed (Lerner and Tirole 2000). Immediate benefits include the increase of the personal use-value of a product and the satisfaction of having achieved something valuable. Delayed benefits involve the recognition among peers as well as rewards from current or future employers, such as higher wages, stock options, or simply more attractive jobs. For both motives, recognition and career concerns, a programmer uses his contributions to signal his capacity to the open source community, to the labor market at large, or even to both.

It is important to point out that some of these different explanations are overlapping. For example, a desire for a higher status within the gift culture may be as strong of an incentive to contribute as economic incentives. However, as noted by Lerner and Tirole, explaining participation by solely social or cultural factors remains a puzzle for several reasons. First, one could expect to reap similar benefits as part of a commercial software development team obviating the need to participate in an open source project. Second, it is not clear why such noble behavior would be limited to the field of software development (Lerner and Tirole 2000). Moreover, a separation of these motives is, for our purposes, not necessary. As Spence (1974) states, "A signal is a manipulable attribute or activity which conveys information... in general it is not necessary to insist that the actor, in manipulating the attribute, think of himself as signaling or conveying information."

A related stream of research is the work by Hars and Ou (2000, 2002). Their work conceptually discusses economic and non-economic motivations of participation and reports the results of a survey of participants of more than 40 different open source projects. In contrast, our work focuses on economic incentives, devises a model to test economic incentives, and tests the hypotheses on contributors of one specific open source project.

3 EXPLAINING OPEN SOURCE PARTICIPATION

Drawing further upon the labor economics literature, we can distinguish between two approaches to model the value of open source participation: human capital theory and signaling theory. Our data allow us to test both approaches.

Human capital explanations for the value of open source participation are straightforward: Participation allows developers to gain marketable technical skills (Becker 1962; Blaug 1976). This seems an undeniable and obvious benefit of participation. An explanation for open source participation consistent with human capital theory would maintain that open source participation is an investment in training that leads to higher earnings in the future. In other words, open source contributors submit their time and effort to gain valuable experience that can be applied to other, potentially better paid, programming jobs. As an investment, the choice to participate depends upon two considerations. First, the individual considers the opportunity cost associated with participation, and second, the individual considers the expected earnings in the job market after participation. Theory predicts that the greater the investment, the greater the return. Therefore, higher earnings should be correlated with higher levels of open source participation.

While attainment of a skill may be an important result of participation, proponents of a sorting or signaling theory of labor markets argue that participation serves as a signal of individual productive capacities to current and future employers (Weiss 1995). Given a distribution of inherent productivity among potential open source participants, the more productive developers would like to signal their superior productivity to employers (Spence 1973). This is even more important when it comes to software productivity. It is very well known that the productivity difference between an average and a top programmer can be quite large (Weinberg 1998). One study of superstar programmers, for instance, found that the top 1 percent produced 1,272 percent more code than the average. At the same time, due to the nature of programming activities, it might be difficult for a programmer to convey fully his or her productive capacities. While it might be relatively easy to identify the star programmers, it is much more difficult to identify above average programmers who have a good understanding of the problem and often develop an efficient solution for the problem at hand. Further, the level of contributions per se might not be the best indicator of productive capacity. Open source projects represent very large-scale, distributed development projects involving thousands of contributions from hundreds of developers (Mockus et al. 2000; O'Reilly 2000). High ability contributors typically make many submissions to the code base, but it is the depth of their understanding, the efficient design of the solution, and their ability to persuade, to get people "on board" with their ideas and strategies that represent the true quality of their contribution. While possible, as a practical matter, it is difficult for employers to efficiently evaluate these qualities based on individual source level contributions. It seems reasonable then that employers seek a reliable proxy that is correlated with these desirable characteristics indicative of or obtained through successful open source participation. If potential employers can use open source participation as a signaling mechanism, then the existence of a "credential" or observable measure of successful participation would allow firms to make inferences about a developer's productive capacity. In so far as open source participation indicates ability or motivation, it can be used either by employers to screen potential employees or by applicants to signal these desirable traits.

4 RESEARCH SITE AND DATA COLLECTION: THE APACHE HTTP PROJECT AND THE APACHE SOFTWARE FOUNDATION

To determine whether there are economic returns to participation in open source development, we investigate three open source projects under the control of the Apache Software Foundation (ASF). The Apache HTTP (Web) server and associated projects are some of the most successful open source products to date. The Apache server, the original ASF project, and its derivatives, have a dominant 63 percent share of the Web server market (Netcraft 2001). Since its inception, the Apache Web server has had over 7,000 source code contributions from over 400 different open source developers (Mockus et al. 2000).

4.1 The Apache Software Foundation (ASF)

The ASF is a not-for-profit corporation that provides the legal, organizational, and financial infrastructure for the software projects gathered under the ASF open-source umbrella. Each of the ASF projects operates autonomously including all aspects of product development. ASF projects are characterized by a "collaborative, consensus-based development process, an open and pragmatic software license, and a desire to create high quality software that leads the way in its field" (Apache 2001). Membership in the ASF is by invitation only and is based on a strict meritocracy. The ASF encompasses seven subprojects related to the development of a full-featured Web server product offering.

(1) The Apache server project is a freely available source code implementation of an HTTP (Web) server. It is the project around which the Apache Group initially formed.

(2) The Apache Portable Runtime project is a free library of C data structures and subroutines designed to facilitate porting the Apache HTTP Server to a host of disparate operating systems.

(3) The Jakarta project consists of all Apache related server side Java projects. Jakarta consists of over 18 Java related subprojects.

(4) The Apache/Perl project is the integration of the Perl programming language implemented as an Apache HTTP server module.

(5) The PHP project is a server side embedded scripting language implemented as an Apache HTTP server module.

(6) The Tool Control Language (Tcl) project is an umbrella for Tcl-Apache integration efforts. These projects combine the Apache Web server with the Tcl scripting language.

(7) The Apache XML project is home for Apache XML related activities. There are more than nine XML related subprojects.

Although any of the Apache projects could provide an interesting vehicle to explore our research question, we have chosen to concentrate our data collection efforts on the HTTP, Jakarta, and Mod_Perl projects for the following two reasons. First, these projects are by far the largest, both in terms of the number of developers and the number of contributions. Second, access to archival data for these projects has proved to be less problematic than for some of the smaller projects.

4.2 The Apache Career

A common characteristic of open source projects is the presence of a strong project leader (Raymond 2000a). Apache, however, is unique among open source projects in this regard. Since its inception, the Apache project has operated under a model of shared leadership and responsibility (Fielding 1999). This model of shared responsibility is reflected in the principles of the meritocracy that define advancement within the ASF (Apache 2001). As a meritocracy, status, responsibility, and benefits are commensurate with contribution. There are five observable levels of recognition or rank within the ASF. In order of increasing status, these are developer, committer, ASF member, project management committee member, and ASF board member. In all cases, advancement is in recognition of an individual's commitment and contributions to an Apache project. This hierarchy within the ASF makes the Apache project uniquely positioned to evaluate open source participation. Ideally, data for identifying economic returns to a variable serving as a signal in labor markets would contain exogenous variation in the signal status among individuals with similar levels of human capital (Tyler et al. 2000). Participants in ASF projects possess such a variable or credential: their rank or status within the ASF.

Individual reasons for initial involvement in any Apache project vary. Typical reasons cited include reporting a problem or "bug," or fixing a problem in the software that has become a nuisance or impairs usage. Another reason is to extend existing functionality or add new features required by the user or the user's organization. For the majority of contributors, there is a single encounter with the project. Some developers, however, choose a deeper level of involvement and continue to make contributions. If developers' contributions are significant and consistent over a period of time they may be nominated for an increase in rank from developer to *committer*. The practical significance of attaining the rank of committer on any Apache project is the privilege of submitting code changes directly to the source code repository as opposed to going through an intermediary to have the changes included in the product. An existing ASF member may nominate committers who continue their involvement in the project for ASF membership. ASF membership is largely a matter of recognition and carries with it a certain prestige in the Apache community. ASF members are eligible to be nominated to the ASF Board of Directors or to serve on a project committee. Project committee members are responsible for all aspects of managing an Apache subproject including project plans and roadmaps, release schedules, etc. The ASF Board of Directors makes decisions regarding corporate governance as well as decisions regarding the addition of new projects under the ASF organizational umbrella.

4.3 The Sample and the Data

The data for this research come from two primary sources: Apache project archives and a targeted survey of Apache participants. Archival data are open source project artifacts such as e-mail and source code archives, source code version control meta-data, and developer Web sites. From these archives, we extracted information pertaining to Apache career advancement as well as individual contributor participation in the development of Apache projects. Survey data came from a questionnaire targeted to Apache contributors. The purpose of this survey was to augment developers' Apache contribution data with their demographic and job history data.

From the archival data, we identified 1,348 contributors. Dr. Roy Fielding, the chairman of the Apache Software Foundation, introduced the survey to the contributors via e-mail in November 2001; 233 e-mails were undeliverable. Of the remaining 1,115 contributors, 325 filled out the survey, yielding a response rate of 29 percent. For this research, we deleted all contributors who earned income in year 2000 outside of the United States or chose not to report any income data. This yielded a sample of 137 contributors. Of those, 93 participants started to contribute in 2000 or later. These people were assigned to the group no rank. The total wages earned in 2000 in the group no rank was statistically not significantly different from the group with rank developer and were subsequently pooled.

5 MODELING DELAYED RETURNS TO OPEN SOURCE PARTICIPATION

We estimate essentially Mincerian wage models that have been traditionally used to test the impact of education on log-earnings (Mincer 1974). The unit of observation is a contributor's number of contributions, rank and background information in 1999. In our setting, the total wage is a lagged function of open source contribution, rank within the Apache Software Foundation, accumulated work experience, accumulated programming skills, education, firm size, firm type (publicly listed or private), job switch, and industry in year 1999.

The dependent variable is the log of the sum of wages and bonuses in the year 2000. As was noted before, delayed returns of open source participation may include other benefits, the most important of which are stock options. However, we chose not to ask for detailed information about stock options for the following reasons. First, open source participants working at start-ups told us that they were not allowed to disclose the number of stock options that they were holding, making it impossible to estimate the value of an option. Further, a significant percentage of respondents work at private firms, which makes it difficult to assess the value of these options.

The independent variables are:

- $Contribution_{1999}$: The unit of analysis is commonly known as a *patch*. Patches are analogous to modification requests (MRs) in traditional software development environments. Unlike MRs in traditional environments, patches in an open source environment result from largely random developer submissions and have no formal designation or means of tracking. This research follows the method used by Mockus et al. (2000) to reconstruct patches from source code archives. For each patch, we extract and retain common software metrics including lines of code added and deleted, the date of submission, the names and number of source code files affected by the change, and change log entries. In our equation, $Contribution_{1999}$ measures the number of patches submitted and accepted in 1999. If contributions are a good proxy for the learning experience of an open source programmer, we expect $Contribution_{1999}$ to have a positive effect on wages.

- $Committer^{+}_{1999}$: This variable is 1 if the rank of the contributor is *committer* or above in 1999 and 0 otherwise. From an Apache career perspective, the long career path of an Apache contributor (five ranks) and the natural scarcity of higher rank contributors made it more practical to focus on the first distinctive promotion from a developer to a committer. This approach was supported statistically: The wages of contributors with rank higher than committer were not significantly different from those of committers. If rank is a signal of productive capacity as a programmer, we expect $Committer^{+}_{1999}$ to have a positive effect on wages.

- $WorkExperience_{1999}$: The total number of years of work experience of a contributor in 1999. Typically, we expect that wages increase with work experience, but that the percentage increase declines with higher work experience.

- $ProgrammingExperience_{1999}$: The number of years of experience in programming languages and programming tools in 1999. As characteristics such as depth of understanding, design quality of a solution, and programming efficiency rise with experience, this variable proxies for the abilities of the programmer.

- Education: *Masters*/*PhD*: *PhD* is 1 if the contributor's highest obtained degree is a Ph.D. and is 0 otherwise. *Masters* is 1 if the contributor's highest obtained degree is a Master's degree and is 0 otherwise. Higher education should lead to higher levels of wages.

- $JobSwitch_{2000}$: $JobSwitch_{2000}$ is 1 if the contributor has switched jobs in 2000 and is 0 otherwise.

- $FirmSize_{1999}$: $FirmSize_{1999}$ is 1 if the contributor's employer had more than 200 employees in 1999 and is 0 otherwise.

- $FirmPublic_{1999}$: $FirmPublic_{1999}$ is 1 if the contributor's employer was a public company in 1999 and is 0 otherwise.

- Industry: $SWIndustry_{1999}$/$EcomIndustry_{1999}$: These variables are set to 1 if the contributor's employer belongs to the software or e-commerce industry respectively, and are set to 0 otherwise.

6 RESULTS AND INTERPRETATIONS

Our sample contains responses from developers, committers, ASF members, and project committee members, with the maximum of 172 contributions. Overall, open source participants earned total wages of $82,000, with a mean of $80,000 for people with no rank or with rank developers and a mean of $110,000 for people with rank of committer. People with higher rank earned, on average, slightly less than committers. In our sample, only 39 percent of all developers contributed in 1999 to an open source project, while all committers and 75 percent of ASF members and project committee members contributed. Our sample is certainly well educated, with 11 percent of the contributors have a doctoratal degree and 23 percent have a Master's degree. However, none of the contributors with rank committer and above obtained a Ph.D. and none of the contributors with rank higher than committer have a master degree. As expected, a correlation between rank and contribution exists (0.496) and is significant. The highest correlation exists between work experience and programming experience (0.524) and is also significant.

The general model regresses the log of total wages in the year 2000 against the independent variables ($Contribution_{1999}$, $Committer^{+}_{1999}$, $WorkExperience_{1999}$, $ProgrammingExperience_{1999}$, *Masters, PhD*, $JobSwitch_{2000}$, $FirmSize_{1999}$, $FirmPublic_{1999}$, $SWIndustry_{1999}$, and $EcomIndustry_{1999}$). In the first model, we assess the effect of contribution on total wages by omitting the rank variable. In second model, we test the effect of rank by omitting the contribution variable. Last, we include both variables to test both effects. The coefficient for $Contribution_{1999}$ is not significant in both specifications (first and third specification), indicating that contributions *per se* do not increase wages. Other specifications such as the sum of contributions until 1999 also did not yield significance. On the other hand, the coefficient for $Committer^{+}_{1999}$ is positive and significant. The wage of contributors with rank committer or above is on average about 29 percent higher than that of developers after controlling for education, programming skills, work experience, job switch, and firm characteristics. These results suggest that employers of contributors in general do not reward participants for their learning experience in the open source project. However, the higher wage paid to contributors with higher rank is consistent with the idea that the rank conveys sought-after, but typically hard-to-observe, characteristics that distinguish above-average programmers.

The coefficients of the control variables are generally in the predicted direction. Contributors with a doctoratal degree are paid about 20 to 22 percent more than college graduates. The coefficient for Master's degree holders is, surprisingly, negative, but not significant. Greater work experience increases the salary by 13 to 14 percent (significant), but with increasing work experience, the increase in salary is growing more slowly (significant). The coefficient for years of programming experience is not significant, this is due to the high correlation between years of programming experience and years of work experience causing the non-significant result. A job switch increases the salary on average by 15 to 17 percent (significant). Public firms also pay more than private firms by about 15 to 16 percent more (significant), while firm size is insignificant. Last, contributors in software companies pay significantly more than firms in other industries, about 12 to 13 percent.

7 CONCLUSION

There are many interesting questions surrounding the open source phenomenon. The research presented here seeks to explore one of the more puzzling aspects regarding open source participation: Why do developers participate? We establish two plausible theoretical foundations for the existence of returns to participation from the economics literature, namely, human capital and signaling theory. Our analysis suggests that employers do not reward the accumulation of experience in open source projects. Rather, higher open source rank is associated with higher wages, even when controlling for work experience and programming experience. This is consistent with the notion that firms make inferences about productivity differences based on the rank of the contributor.

Given the great interest in the success factors of open source projects, this study establishes an important insight: Firms may accept the easily verifiable rank within an open source project as an indicator for hard to observe characteristics of a programmer such as the productive capacity, the tenacity to convince others of the design, or the ability to motivate others to contribute to the specific project.

8 REFERENCES

Apache. http://www.apache.org/foundation/, Apache Software Foundation, 2001.

Becker, G. S. "Investment in Human Capital: A Theoretical Analysis," *Journal of Political Economy* (70:5, Part 2: Investment in Human Beings), 1962, pp. 9-49.

Blaug, M. "The Empirical Status of Human Capital Theory: A Slightly Jaundiced Survey," *Journal of Economic Literature* (14:3), 1976, pp. 827-855.

Brooks, F. P. *The Mythical Man Month: Essays on Software Engineering*, Silver Anniversary Edition. Reading, MA: Addison-Wesley Pub. Co., 1995.

Clary, E. G., Ridge, R. D., Stukas, A. A., Snyder, M., Copeland, J., Haugen, J., and Miene, P. "Understanding and Assessing the Motivations of Volunteers: A Functional Approach," *Journal of Personality and Social Psychology* (74:6), 1998, pp. 1516-1530.

Fielding, R. "Shared Leadership in the Apache Project," *Communications of the ACM* (42:4), 1999, p. 2.

Fitzgerald, B., and Feller, J. "Open Source Software: Investigating the Software Engineering, Psychosocial and Economic Issues," *Information Systems Journal* (11:4, Open Source Software: Investigating the Software Engineering, Psychosocial and Economic Issues), 2001.

Ghosh, R. A. "Interview with Linus Torvalds: What Motivates Free Software Developers?," *First Monday* (3:3), 1998.

Hars, A., and Ou, S. "Why is Open Source Software Viable? A Study of Intrinsic Motivation, Personal Needs, and Future Returns," in H. M. Chung (ed.), *Proceedings of the Sixth Americas Conference on Information Systems*, Long Beach, CA, 2000, pp. 486-490.

Hars, A., and Ou, S. "Working for Free? Motivations for Participating in Open Source Projects," in *Proceedings of the 34th Hawaii International Conference on System Sciences*. Los Alamitos, CA: IEEE Computer Society Press, 2001.

Lakhani, K., and von Hippel, E. "How Open Source Software Works: 'Free' User-to-User Assistance," MIT Sloan Open Source Project, 2000 (available online at http://opensource.mit.edu/papers/lakhanivonhippelusersupport.pdf; accessed March 2001).

Lerner, J., and Tirole, J. "The Simple Economics of Open Source," NBER Working Paper w7600, The National Bureau of Economic Research, Inc., March 2000 (available online at http://papers.nber.org/papers/W7600; accessed April 4, 2001).

Koch, S., and Schneider, G. "Effort, Cooperation and Coordination in an Open Source Software Project: GNOME," *Information Systems Journal* (12:1, A Further Investigation of Open Source Software: Community, Co-ordination, Code Quality and Security Issues), 2002.

Mauss, M. *The Gift: Forms and Functions of Exchange in Archaic Societies*. New York: Norton, 1967.

Mincer, J. *Schooling, Experience, and Earnings.* New York: Columbia University Press, 1974.

Mockus, A., Fielding R., and Herbsleb, J. "A Case Study of Open Source Software Development: The Apache Server," in *Proceedings of the 22nd International Conference on Software Engineering*. New York: ACM Press, 2000.

Netcraft . *The Netcraft Web Server Survey*. Bath, England: Netcraft, 2001.
O'Reilly, T. "Open Source: The Model for Collaboration in the Age of the Internet," Keynote Presentation to the Conference on Computers, Freedom and Privacy, Toronto, Canada, April 6, 2000 (available online at http://www.oreillynet.com/pub/a/network/2000/04/13/CFPkeynote.html; accessed June 2001).
OSI. "The Open Source Definition," The Open Source Initiative 2001 (available online at http://opensource.org/docs/definition_plain.html).
Raymond, E. "The Cathedral and the Bazaar," 2000a (available online at http://tuxedo.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/; accessed October 15, 2000).
Raymond, E. "Homesteading the Noosphere," 2000b (available online at http://tuxedo.org/~esr/writings/homesteading/homesteading/; accessed April 4, 2001).
Raymond, E. "The Magic Cauldron," 2000c (available online at http://tuxedo.org/~esr/writings/homesteading/magic-cauldron/; accessed April 2001).
Spence, M. "Job Market Signaling," *Quarterly Journal of Economics* (87:3), 1973, pp. 87(3): 355-374.
Spence, M. *Market Signaling: Information Transfer in Hiring and Related Screening Processes.* Cambridge, MA: Harvard University Press, 1974.
Tyler, J. H., Murnane, R. H., ALL AUTHORS. "Estimating the labor market signaling value of the GED," *Quarterly Journal of Economics* (115:2), 2000, pp. 431-468.
Weinberg, G. *The Psychology of Computer Programming*, Silver Anniversary Edition. New York: Van Nostrand Reinhold, 1998.
Weiss, A. "Human-Capital versus Signaling Explanations of Wages," *Journal of Economic Perspectives* (9:4), 1995, pp. 133-154.