Why Do Developers Contribute to Open Source Projects? First Evidence of Economic Incentives

Il-Horn Hann, Jeff Roberts, Sandra Slaughter
Graduate School of Industrial Administration
Carnegie Mellon University
Pittsburgh, PA 15213
(412) 268-4637
hann@andrew.cmu.edu

Roy Fielding
Chairman
Apache Software Foundation
fielding@apache.org

ABSTRACT

The availability of commercial quality, free software products such as the Apache HTTP (web) server or the Linux operating system has focused significant attention on the open source development process by which these products were created. One of the more perplexing aspects of open source software projects is why developers freely devote their time and energy to these projects. While many open source participants cite idealistic motives for participation, 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. Preliminary results of our empirical investigation suggest that greater open source participation per se, as measured in contributions made, does not lead to wage increases. However, a higher status in a merit-based ranking within the Apache Project does lead to 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 2000). Others liken the open source community to a gift culture where the status of a participant depends on ‘what he gives away’ (Raymond 2000) . Alternatively, Lerner and Tirole suggest that open source participation may in part be explained by existing theories of labor economics.

Understanding the incentive structure is a critical first step in evaluating open source as a viable development model for commercial software engineering endeavors. Raymond makes the point that the vast majority of software written has no sale value and does not provide any competitive advantage to the firm. Raymond (2000) contends that for these software projects, open source development should be considered as a viable alternative. Before businesses can rely on open source to develop and maintain large systems; however, the fundamental question of the contributor’s incentive must be well understood.

2. Explaining Open Source Participation

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 protect 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 2000). As a second explanation, Raymond (2000) 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 (Mauss 1967; Clary, Ridge et al. 1998).

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 career concern incentives. However, as noted by Lerner & 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 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” (Spence 1974).

Borrowing further from 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. 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 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 by either employers to screen potential employees or by applicants to signal these desirable traits.

3. 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% 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 2000).

3.1 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 over 9 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.

3.2 The Apache "Career"

A common characteristic of open source projects is presence of a strong project leader (Raymond 2000). 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, project management committee member, ASF 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, Murnane 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. PMC members are nominated and elected by the project committers, and then formally appointed by the ASF board. Project committee members are responsible for all aspects of managing an Apache subproject including project plans and roadmaps, release schedules, etc. ASF members are nominated and elected by the existing ASF members based on their contributions and their ability to work within a collaborative community. ASF membership carries with it a certain prestige in the Apache community and is required to be nominated for the ASF Board of Directors. The ASF Board of Directors, elected by ASF members, makes decisions regarding corporate governance as well as decisions regarding the addition of new projects under the ASF organizational umbrella.

3.3 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 email 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.

4. Preliminary Results and Interpretation

Our preliminary results indicate that contributions, as measured by number of patches submitted, per se do not increase wages. On the other hand, the wage of contributors with rank committer or above is on average about 29% higher than that of developers after controlling for education, programming experience, work experience, job switch, and firm characteristics. These results suggests 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.

5. ACKNOWLEDGMENTS

We would like to thank Rebecca Tsui for her comments and the numerous open source programmers who have contributed to this study.

6. REFERENCES

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

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

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

Clary, E. G., R. D. Ridge, et al. (1998). "Understanding and Assessing the Motivations of Volunteers: A Functional Approach." Journal of Personality and Social Psychology **74**(6): 1516-1530.

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

Ghosh, R. A. (1998). "Interview with Linus Torvalds: What motivates free software developers?" First Monday **3**(3).

Lerner, J. and J. Tirole (2000). The Simple Economics of Open Source, The National Bureau of Economic Research, Inc. (Accessed: April 4, 2001). http://papers.nber.org/papers/W7600

Mauss, M. (1967). The gift; forms and functions of exchange in archaic societies. New York, Norton.

Mincer, J. (1974). Schooling, experience, and earnings. New York, Columbia University Press.

Mockus (2000). A case study of open source software development: the Apache server. Proceedings of the 22nd international conference on on Software engineering, Limerick Ireland, ACM.

Netcraft (2001). The Netcraft Web Server Survey. Bath, England, Netcraft.

O'Reilly, T. (2000). Open Source: The Model for Collaboration in the Age of the Internet. Computers, Freedom and Privacy, Toronto, Canada.

OSI (2001). The Open Source Definition, The Open Source Initiative. (Accessed: May, 2001). http://opensource.org/docs/definition_plain.html

Raymond, E. (2000). The cathedral and the bazaar, Eric Steven Raymond. (Accessed: October 15, 2000). http://tuxedo.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/

Raymond, E. (2000). Homesteading the Noosphere, Eric Steven Raymond. (Accessed: April 4, 2001). http://tuxedo.org/~esr/writings/homesteading/homesteading/

Spence, M. (1973). “Job Market Signaling.” Quarterly Journal of Economics **87**(3): 355-374.

Spence, M. (1974). Market signaling: Information transfer in hiring and related screening processes. Cambridge, MA, Harvard University Press.

Tyler, J. H., R. J. Murnane, et al. (2000). “Estimating the labor market signaling value of the GED.” Quarterly Journal of Economics **115**(2): 431-468.

Weinberg, G. (1998). The psychology of computer programming. Silver Anniversary Edition. New York, Van Nostrand Reinhold.

Weiss, A. (1995). “Human-Capital Vs Signaling Explanations of Wages.” Journal of Economic Perspectives 9(4): 133-154.