Open Source Software Development Experiences on the Students' Resumes: Do They Count?
- Insights from the Employers' Perspectives

Ju Long
Texas State University-San Marcos, San Marcos, TX, USA

julong@txstate.edu

Executive Summary

Open Source Software (OSS) is a major force in today's Information Technology (IT) landscape. Companies are increasingly using OSS in mission-critical applications. The transparency of the OSS technology itself with openly available source codes makes it ideal for students to participate in the OSS project development. OSS can provide unique opportunities for students to learn and develop their software project development skills. From an educator's perspective, we argue that it is beneficial for the students' career prospects if OSS experiences are included in the students' resumes. This research is to test these hypotheses from the potential employers' perspective. We set out to examine the following research questions: First, do the potential employers value the students' OSS development experience? Second, what are the factors that potential employers look for when assessing students' OSS experience? To answer these questions, we surveyed a large sample of CIO/CTOs, project managers, and developers from various technology companies. Most of our respondents had been involved in the corporate hiring process and most were familiar with OSS. Based on the analysis, we found that OSS related experience could have a positive effect on students' career development. Our respondents rated OSS-related experience as being as valuable as experience gained through proprietary software projects. Crucial factors employers look for include whether the OSS projects are supported or compatible with the hiring companies' practices. In addition, experience focusing on programming and architecting were regarded as valuable. Our research is potentially of value to IT educators when they advise students on career development. It also offers insight to IT practitioners when evaluating student applicants with OSS experience.

Keyword: Open Source Software, Career Development, IT Education, Software Development Process, Evaluation and Promotion

Introduction

Open Source Software (OSS) is a major force in today's Information Technology (IT) landscape.

Guided by the Bazaar model first advocated by Eric Raymond (1999), geographically-distributed software developers develop and release software for free (Sowe & Stamelos, 2007). Since the late 1990s, OSS has grown explosively when successful OSS such as Linux, Apache, MySQL and PHP/Perl (LAMP) revolutionized enterprise computing in the Internet age. According to research conducted by Forrester Consulting

(2007), the majority of the companies that have considered OSS ended up adopting it, and more than half of them have adopted OSS for mission-critical applications. Apache, an Open Source web server program, runs on 49.5% of the 176 million websites surveyed by NetCraft in August 2008 (NetCraft, 2008). In addition to existing OSS, there is an abundance of new and innovative OSS emerging from the development community each day. The zero-cost licensing structure of most Open Source projects has opened up the acceptance of these products into a number of previously untapped markets, including the enterprise computing market.

There is no doubt that the zero initial licensing fee of OSS plays an important role in its rapid adoption. But the quality and feature sets of OSS also compare favorably against their Closed Source counter parts. As OSS is adopted by more and more corporations, OSS development and management skills are growing increasingly important in today's IT work places.

Given the transparency of the technology itself with openly available source code (Feller & Fitzgerald, 2002; Perens, 1999), OSS development provides students who major in Information Technology (IT) a unique opportunity to participate in software development, to interact with software developers and project managers, and to gain valuable experiences in the real world. Researchers have proposed that IT courses should include a significant "real-world" experience that is necessary to learn software engineering skills and concepts (Sowe & Stamelos, 2007). However, IT students still graduate without getting enough experiences in realistic and long-term software engineering projects (Liu, 2005). This is partly because most of these projects are Closed Source proprietary software projects, which may not be available for the students to participate.

By contrast, OSS projects provide students with many advantages that proprietary software could not. First, unlike proprietary software projects, OSS projects do not require the contributors to be employees of the company. All the source code is open and free to access by anyone who is interested in the OSS development. Students can join in any OSS project that fits their interests and skills. Second, the peer review process of OSS allows the students to learn through critiques from fellow developers. Students can inspect the code and modify it, learning from and building upon the work of prior programmers (Watson, Boudreau, York, Greiner, & Wynn, 2008a). Third, most of the OSS projects allow developers to contribute to every stage of the development process. Thus, by participating in OSS projects, students can gain a valuable holistic view of the entire project development process. Fourth, the OSS community has a wide range of projects. Students can gain experience in a variety of programming languages, platforms and Integrated Development Environment (IDEs). Lastly, through interacting with the developers and other stakeholders of OSS projects, students can improve their team work skills and communication skills.

Clearly, from an IT educator's perspective, OSS projects provide students with a unique opportunity to enhance their skill sets and broaden their career perspectives. More and more corporations value the skills to make truly innovative products (Schrage, 2000). Students learn how to innovate in OSS projects to enhance their career possibilities. Extant research has already proposed that software engineering educators may utilize OSS to extend the methodology by which we learn, apply, and teach software engineering courses (Sowe & Stamelos, 2007). However, before we encourage students to participate in OSS projects, some questions may still remain to be explored: (1) Do potential employers value OSS related skills and experiences in their hiring process? In other words, if students participate in the OSS projects, do such experiences count? (2) Do employers regard the OSS experiences as valuable as the experiences gained in proprietary software projects? How does OSS experience compare to the other sorts of skill credentials, such as IS certificates and degrees? (3) What criteria do employers use when evaluating the students' OSS experiences? This study addresses these questions.

The rest of the article is organized as follows: in the second section, we discuss the related literature and set the theoretical foundation for the study. Then in the third section, we report the data collection process and demonstrate the data analysis results. In the last section, we discuss the research implications and conclude the article.

Review of Related Theories

Open Source Software and IT Education

The use of OSS in IT education has been emphasized in recent years (Attwell, 2005; González-Barahona, Heras-Quirós, Centeno-González, Matellán-Olivera, & Ballesteros-Cámara, 2000; Liu, 2003). Research proposes that OSS is both an alternative teaching methodology and an educational model (Faber, 2002). Many universities have started teaching OSS courses (German, 2005; Megias, Serra, & Macau, 2005; Ozel, Cilingir, & Erkan, 2006). OSS projects (for example, Edukalibre) have also been launched to study the transfer of OSS practices to the production of educational resources (Barahona, Tebb, Dimitrova, Chaparro, & Romera, 2005), as well as to create e-learning portals (Marengo, Baldassarre, & Pagano, 2008). Research has also been done to create an open source learning object repository (Koohang, Floyd, Santiago, Greene, & Harman, 2008).

There are several successful examples of open source applications in the classroom setting, including PHPWebSite (http://phpwebsite.appstate.edu/), which is a very popular open source web content management system originally developed and maintained by the Web Technology Group at Appalachian State University (which is comprised of upper level and graduate students in Computer Science) (Watson et al., 2008a).

The availability of the source code for OSS provides a unique opportunity for the educators to experiment (Kamthan, 2007). Students get experience in dealing with realistic software systems with large quantities of code written by other people (Carrington & Kim, 2003). Such open technologies further encourage open communication and open learning (Watson et al., 2008a). The typical classroom or corporate development setting often focuses on preselected, preinstalled, and preconfigured components. With more technologies with open standards such as OSS being adopted at the enterprise level, more opportunities are available for students to learn and create knowledge (Watson et al., 2008a).

Research evidence also suggests that the communities in various OSS projects provide peer-to-peer help and critique (Lakhani & von Hippel, 2003) and problem solving for respondents. Joining an open source project is an easy way to interact with and learn from highly respected professionals. As the students begin to contribute code to the project, these colleagues send feedback (Spinellis, 2006). Such a peer-review process facilitates effective learning and sharing of knowledge (Holtgrewe, 2004; Sowe, Stamelos, & Deligiannis, 2006).

Furthermore, research shows that developing OSS could help students in their future career paths (Cusumano, 2004). OSS provides students with a larger picture of the software development process (Spinellis, 2006). Students get to participate in all aspects of software development by working as system administrator, database administrator, and other roles. All of these skills are valuable in today's marketplace.

Researchers have found over and over again that engaging students is critical to deep learning (Guzdial & Soloway, 2002). Researchers argue that the use of open source as metaphor for e-learning should be further explored in three areas: instructional practices, instructional platforms, and instructional philosophy (Koohang & Harman, 2005). OSS provides such opportunities to expose students to intriguing subjects while allowing them to immediately apply concepts to

solve today's problems (Akbulut & Looney, 2007). When participating in the OSS projects, students almost immediately recognize that they are learning something new that is cutting edge. Anyone with programming experience, including educators and students, can contribute to the collection of available open source software (Surran, 2003). Unlike proprietary software, OSS allows extensive customization of software to support the needs of users better. This also allows the users to participate more proactively in the development and implementation process (Lin & Zini, 2008).

Unique Skills Gained through OSS Experiences

OSS is different from traditional proprietary software in many important aspects. It provides unique opportunities for students to learn many skills that may not be derived from developing traditional software. In particular, OSS is unique in its choices of programming languages, platforms, tools, community dynamics, and development processes:

- Programming languages and platforms: OSS developers tend to stay away from proprietary and expensive programming environments such as VB, .Net, Oracle, WebSphere, and SAP. Rather, they either use common general purpose programming languages such as C/C++, SQL, and Java, or use programming languages and platforms that have evolved from OSS. Examples of the latter include Perl, PHP, Python, Ruby, GTK, Linux, Apache, and many open source frameworks in Java.

- Tools: OSS developers are less likely to use proprietary IDE tools such as the Microsoft Visual Studio and Adobe Dreamweaver. Instead, they often opt to use OSS IDEs such as Eclipse and NetBeans. For PHP / Perl / Python / Ruby developers, it is common among developers to use just simple text editors to code. OSS developers rely heavily on open and distributed source code version control tools such as CVS and SVN, while proprietary software developers typically use closed commercial solutions for version control inside a company.

- Community: The OSS development is all about the online community and consists of a group of individuals who are informally bound to collaborate on a shared task (Koohang & Harman, 2005). The software is developed by the community, promoted and marketed by the community, and used by the community. Such a community of practice forms an open, loosely networked, and "virtual" ecosystem. This ecosystem is dynamic and "organic" as OSS developers are loosely related and move across projects and coalesce (Weiss, Moroiu, & Zhao, 2006). For companies that use OSS, fostering good community relationships is crucial for getting timely technical support. Managing OSS community relationship requires a good understanding of what motivates OSS developers to contribute, as well as building personal trust with the project's lead developers outside of the company.

- Development process: The OSS development process differs significantly from traditional commercial software development process. The requirement of OSS is typically collected from the community, as opposed to being handled by product managers. The focus on design is informal, and there is much attention on implementation and in some cases on testing (Kamthan, 2007). Compared with traditional software, OSS has a much more frequent release cycle and lets the user community do much of the quality assurance work. Many OSS projects employ some of the agile practices such as openness to change and parallel development (Warsta & Abrahamsson, 2003).

Because of these differences, a student participating in OSS development can learn new development tools and methodologies that are usually not available through developing traditional proprietary software.

Career Concern and OSS

Open Source Software development exemplifies not only a viable software development approach, but also a model for the creation of self-learning (Sowe, Karoulis, Stamelos, & Bleris, 2004) and self-organizing communities (Sowe & Stamelos, 2007). A central question to answer is what motivates developers to make unpaid contributions to OSS. Past research in this area has indicated that one of the most important motivations is "career concern" (Freeman, 2007; Lerner & Tirole, 2002; Shah, 2006). The career concern incentive refers to future job offers, shares in commercial open source-based companies or future access to the venture capital market (Lerner & Tirole, 2002). Based on the "career concern" literature in labor economics, researchers argue that developers contribute to Open Source development so that they can build up their resumes and get a better career opportunity later. In other words, by participating in OSS projects, software professionals gain a new and valuable way to obtain significant experience in a wide range of areas and to advance professionally (Spinellis, 2006).

In this research, we argue that the rise of OSS, and the career concern motivator in particular, has made it important for higher education institutions to put more emphasis on OSS in IS education curriculum. Students must continuously adapt their skills to meet the ever-changing market demands (Watson, Boudreau, York, Greiner, & Wynn, 2008b). In order to leverage and manage OSS in the real world, MIS/CIS graduates need to be familiar with OSS technologies, communities, and methodologies. And in return, graduates with OSS skills will enjoy an added benefit of the career concern factor and get a head-start in the hiring process.

Results

Demographic Data

To understand how corporate employers evaluate OSS experiences, we conducted a survey on a sample of IT professionals consisting of software developers, system administrators, and project / product managers from software companies and IT departments of other firms. The survey instrument was developed and then pre-tested and revised based on a focus group discussion conducted on 10 participants from a major OSS company. (A copy of the survey is in the Appendix.) The major revision was rewording the instrument to clarify some confusing sentences. The revised instrument was sent out to randomly selected IT professionals listed in the directory of a national association of IT professionals. We chose this directory because this is a professional organization that consists of professionals working in the information technology industry. We wanted to ensure that the sample included a mixture of both the proprietary and open source software companies, and respondents with different work experiences, positions, and from organizations of different sizes. We sent out a total of 300 surveys by mail with stamped return envelope included. We received 100 valid responses in return, with a response rate of 30%.

The demographic data analysis results are as follows: the mean work experience was 15 years, ranging from 3 to 35 years. About 46% were developers; 40% were Chief Information Officers (CIOs) or Chief Technology Officers (CTOs) in a company; and 11% were product managers.

A majority of the respondents (68%) work for proprietary companies. Twenty percent of the respondents worked for companies that use both proprietary and Open Source software. Another 6% worked for companies that only use Open Source software.

To ensure the validity of the respondents' inputs, we also examined the respondents' actual experiences in the corporate hiring process. Of all the respondents, 90% have participated in the hiring process and 91% have participated in the performance and ability evaluation process.

Below are the results from our data analyses.

Familiarity of Open Source Software

We first use a 3 point Likert-type scale to examine whether the respondents are familiar with OSS, including not familiar, somewhat familiar, and very familiar. Among the respondents, 89% of them are very familiar with OSS. Another 11% are somewhat familiar with OSS. We did not have any respondent who is unfamiliar with OSS. We concluded that our respondents are familiar with OSS and could give us valid inputs on OSS development.

We also measured whether the respondents had worked with developers with OSS experiences. Eighty-five percent of the respondents had worked with OSS developers. This result further shows the respondents are familiar with the OSS development.

We also ask our respondents whether they have used OSS development experiences in evaluating developers' abilities and qualifications. Fifty percent of the respondents have used OSS project experiences in evaluating the developers' abilities and qualifications, which suggested that employers take OSS experience into account in the hiring process.

OSS Experience's Impact in Hiring and Promotion Processes

We found that our respondents regarded Open Source experiences as valuable during the hiring and promotion process. More than 88% of the respondents agree that the OSS experience could have a positive influence in the hiring process. .

Fifty-six percent (56%) of the respondents agree that OSS experience could have a positive influence in the promotion process, fewer than those who believed it was important to the hiring process. This may be because OSS experience is a useful indicator of skills when hiring but has less relevance for promotion within an organization where it is not directly relevant to the specific tasks or skills used in-house (see Figure 1).
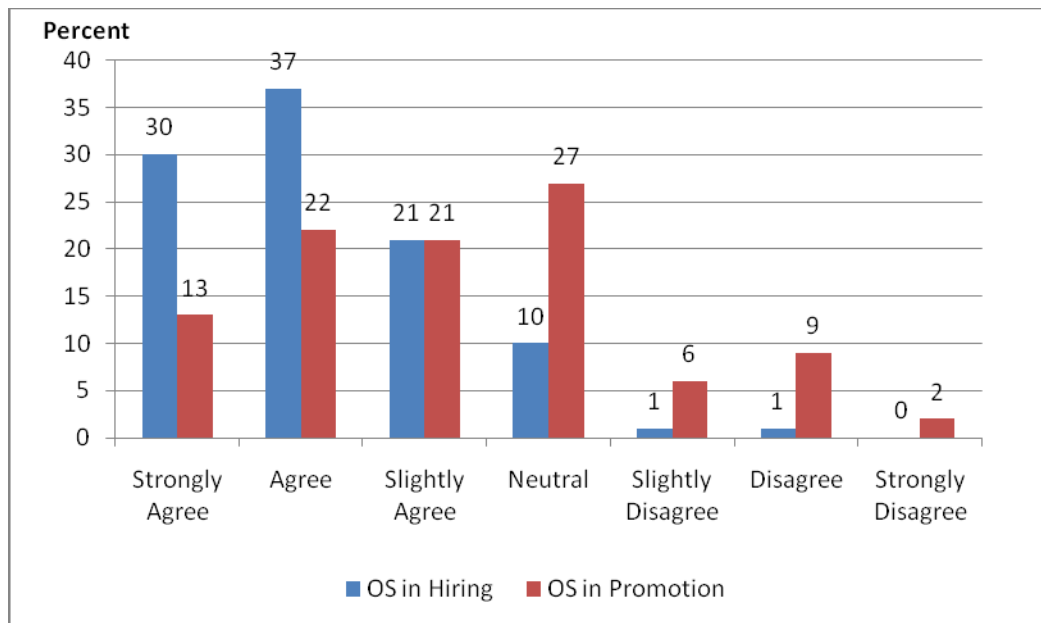

Figure 1. Percent of respondents who agree or disagree that
OSS Experience Has Difference Impact in Hiring and in Promotion

OSS Experience as an Indicator of Developers' Abilities

One of our research objectives was to determine whether OSS experience was considered a valid indicator of the developers' abilities. Eighty percent (80%) of the respondents thought that per-

formance in Open Source Software projects was a good indicator of the developers' abilities, with around 60% of them selected strongly agree or agree (see Figure 2).
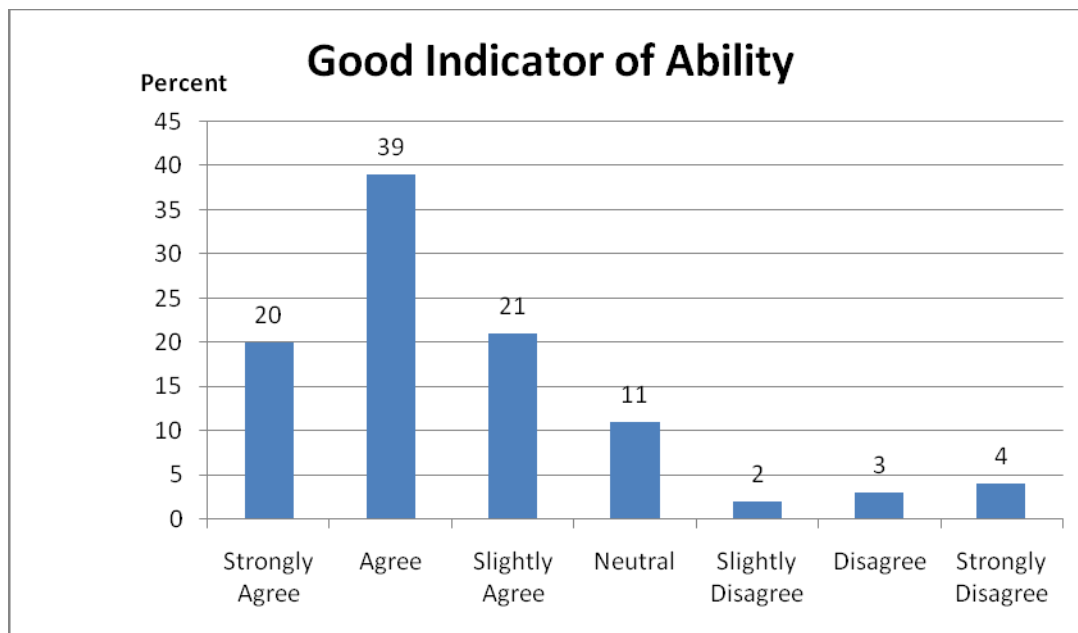

Figure 2. Percent of respondents who agree or disagree that
OSS experiences as a Good Indicator of Ability

OSS Experiences Compared to Other Credentials and Experiences

Another research objective was to examine the value of OSS experience compared with other types of credentials and experiences. The respondents compared the value of developer's OSS experience with experience in developing proprietary software projects and degrees and diplomas.

Experience in OSS was rated as effective as experience in proprietary software projects. However, when compared with a diploma or a certificate, respondents do not agree that the experience in OSS could replace a degree or diploma (see Figure 3). The results suggest that formal education and training are considered irreplaceable; however, OSS experience can serve as a valuable complementary component to formal college education. As argued earlier, with OSS experience students gain unique experiences that can be complementary to the formal college education, including how to interact well with clients, understand their requirements, abstract these into technical designs, and deliver workable solutions. They can enhance more skills than most typical classroom experiences can provide (Watson, et al, 2008a).
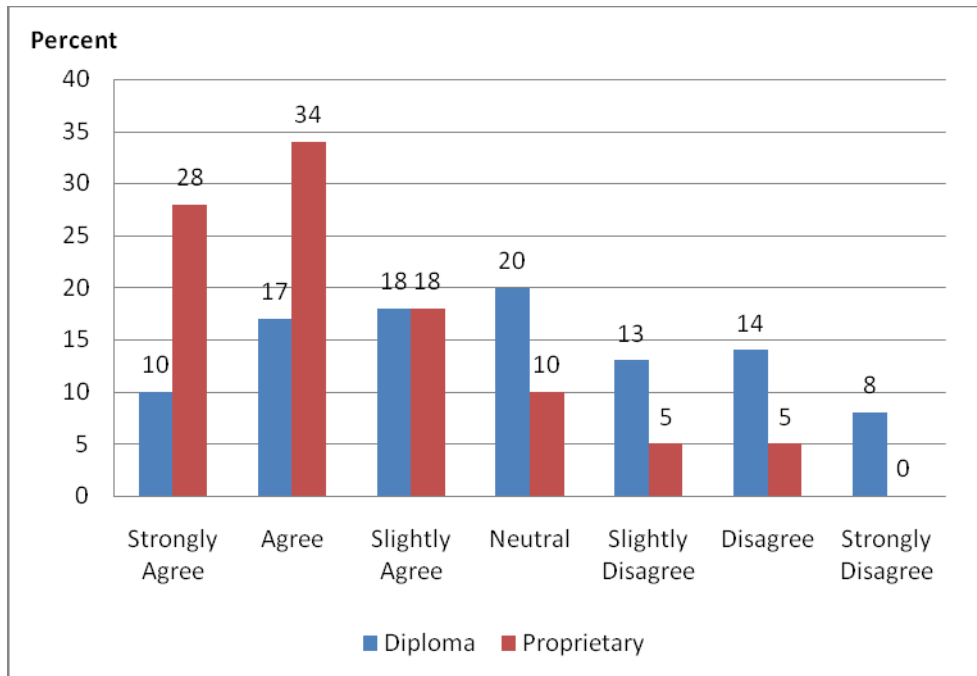
Figure 3. Percent of respondents who agree or disagree that
OSS experiences as effective as Proprietary software experiences but not as effective as diploma

Important Factors to Consider in the Hiring Process

Our respondents also rated factors to consider when evaluating the developers' Open Source experiences. Of all the factors, the most important and the highest rated one was whether the skills the developers learn in the OSS experiences are supported or compatible with the company's existing or future projects. The second most important factor was the programming language the developers use in the OSS projects. By contrast, years of experience and the type of operation systems were not considered as important and so were ranked lower (see Figure 4).
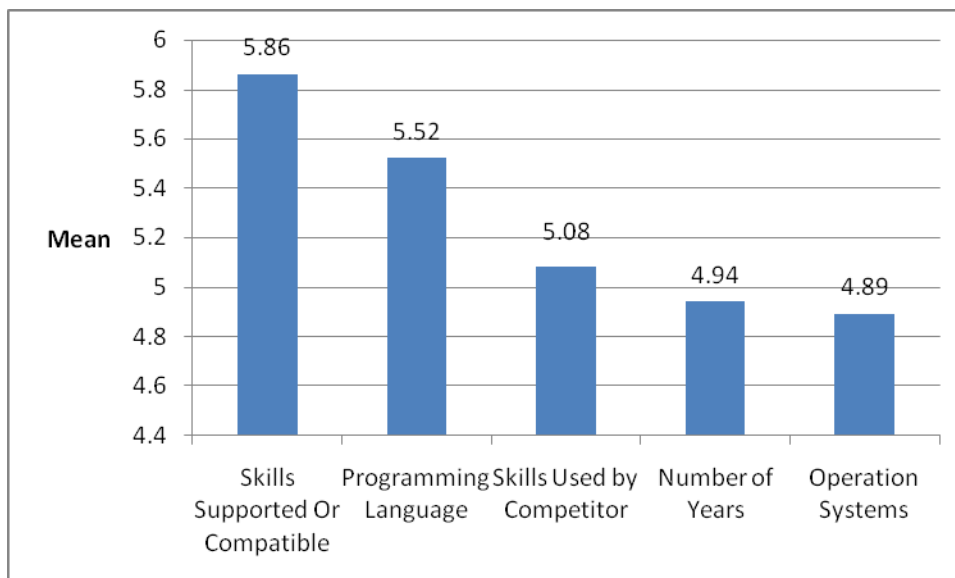

Figure 4. The mean rate of importance of the factors related to OSS experiences

OSS Activities as Indicators of the Developer's Abilities

The extensive peer collaboration in OSS development allows project participants to write code, debug, test, and integrate software (Sowe & Stamelos, 2007). We surveyed our respondents' opinions on the importance of all OSS development activities.

The results show that programming and architecting were considered to be the most imporant activities and have the most value to the developers (see Figure 5). By contrast, other activies, such as promotion and advocation and recruiting, are considered less important.
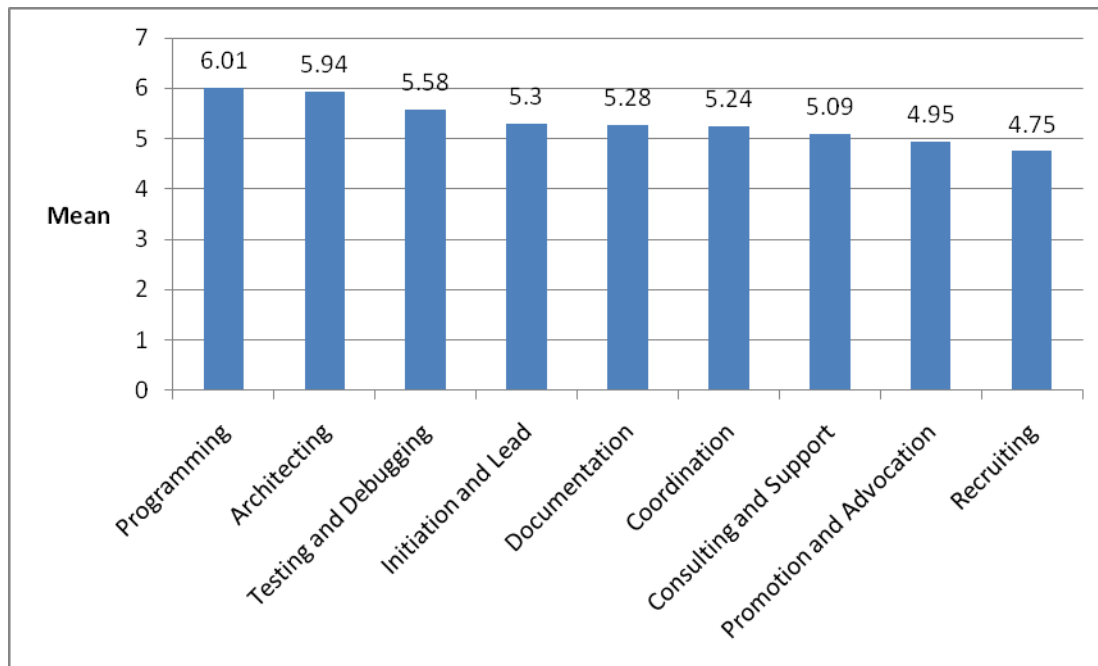

Figure 5. Rank of importance of OSS activities on a seven-point scale

Influences of the OSS Projects' Own Characteristics

Which OSS projects the students participate in may also have an influence in the hiring process. For instance, do the respondents regard the popularity of the projects as important on the hiring decision? Is the compatibility of the projects with the hiring companies' own projects considered more important? We asked the respondents to evaluate what project characteristics affected their hiring decision. As indicated in Figure 6, the most important charateristic was whether the OSS projects fitted the hiring company's own projects. Compatibility between the OSS projects and the hiring company's customer base is rated as the second most important. On the other hand, we found that the popularity of the project was not rated as high as the previous two factors. It implied that the students do not have to work for the most popular project.
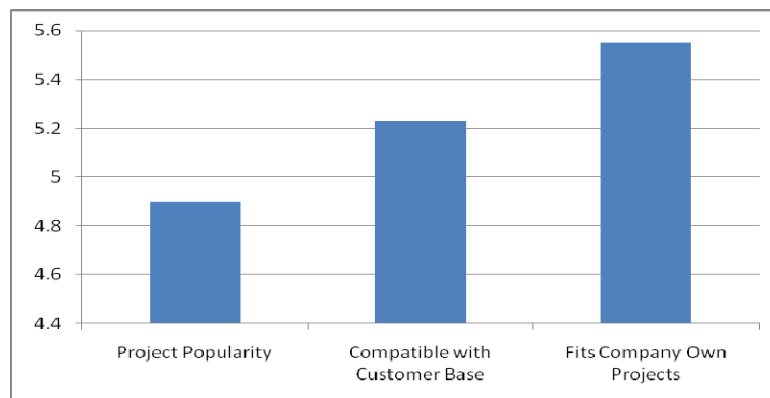
Figure 6. Means of the ratings of OSS project characteristics

Research Contribution and Conclusion

Our research is one of the first studies that examine OSS experiences and its potential impact on IT students' career development and education from the employers' perspective. Open Source Software has been increasingly used in the industry for mission-critical tasks. The career benefits from participating in OSS development could provide students with the motivation to self-manage their continued learning throughout their careers (Watson et al., 2008a)

Respondents in our study were all real world practitioners with a mean of 15 years of work experience. They came from a variety of companies, including companies that used proprietary software, open source software, or both. More than 90% of the respondents had been involved in the hiring process and promotion process. The results based on these respondents shed light on how potential employers evaluate the students with OSS development experience.

Our study suggests that participating in Open Source Software development is beneficial to the students' career development from the employers' perspectives. The majority of our respondents strongly agree that OSS development experience is as valuable as any experience gained in proprietary software development projects. It is also worth noting that although OSS development experience is considered as valuable to the employers, it cannot replace diplomas and formal college education. Our study also examines how potential employers evaluate OSS related experience. We concluded that the most important factor is whether the skills students learn through the OSS experienceis supported or compatible with the hiring company's projects and its customer base. These findings give IS students and educators valuable information when selecting which OSS projects to join. We also found that the among all the OSS activities, employers value programming and architecting the most. Among all the OSS projects' characteristics, the fit between the hiring company's own projects and the OSS projects should be the single most important consideration when selecting which projects to join. These findings show what the employers consider important when evaluating the OSS experiences, thus giving IT students and educators more detail on how to select OSS projects. Our study could also be valuable to potential employers as an aid when hiring.

References

Akbulut, A. Y., & Looney, C. A. (2007). Inspiring students to pursue computing degrees. *Communications of the ACM, 50* (10), 67-71.

Attwell, G. (2005). What is the significance of open source software for the education and training community? *Proceedings of the First International Conference on Open Source Systems (OSS 2005),* Genova, Italy, July 11-15, 2005.

Barahona, J. M., Tebb, C., Dimitrova, V., Chaparro, D., & Romera, T. (2005). Transferring libre software development practices to the production of educational resources: The edukalibre project. *Proceedings of the First International Conference on Open Source Systems (OSS 2005)*, Genova, Italy, 11th-15th July, 2005, pp. 341-348.

Carrington, D., & Kim, S. (2003). Teaching software engineering design with Open Source Software. *Proceedings of 33rd ASEE/IEEE Frontiers in Education Conference*, Nov. 5-8, Boulder, CO.

Cusumano, M. A. (2004). Reflections on free and open software. *Communications of the ACM, 47*(10), 25-27.

Faber, B. D. (2002). Educational models and open source: Resisting the proprietary university. *Proceedings of the 20th Annual international Conference on Computer Documentation*, SIGDOC '02, ACM Press, pp. 31-38.

Feller, J., & Fitzgerald, B. (2002). *Understanding open source software development.* Boston, MA: Addison-Wesley.

Forrester Consulting. (2007). Open Source Software's expanding role in the enterprise. Retrieved December 5, 2007 from http://www.unisys.com/eprise/main/admin/corporate/doc/Forrester_research-open_source_buying_behaviors.pdf

Freeman, S. (2007). The material and social dynamics of motivation: Contributions to Open Source language technology development. *Science Studies*, *20* (2), 55-77.

German, M. D. (2005). Experience teaching a graduate course in Open Source Software Engineering. *Proceedings of the First International Conference on Open Source Systems (OSS 2005)*, Genova, Italy, 11th-15th July 2005, pp. 326-328.

González-Barahona, J. M., Heras-Quirós, P. D. L., Centeno-González, J., Matellán-Olivera, V., & Ballesteros-Cámara, F. (2000). Libre software for computer science classes. *IEEE Software, 17*(3), 76-79.

Guzdial, M., & Soloway, E. (2002). Teaching the Nintendo generation to program. *Communications of the ACM, 45*(4), 17-21.

Holtgrewe, U. (2004). Articulating the speed(s) of the Internet: The case of Open Source/Free Software. *Time & Society, 13*(1), 129-146.

Kamthan, P. (2007). On the prospects and concerns of integrating Open Source Software environment in software engineering education. *Journal of Information Technology Education, 6*, 45-64. Retrieved from http://jite.org/documents/Vol6/JITEv6p045-064Kamt214.pdf

Koohang, A., Floyd, K., Santiago, J., Greene, S., & Harman, K. (2008). Design, development, and implementation of an Open Source learning object repository (OSLOR). *Issues in Informing Science and Information Technology, 5*, 487-498. Retrieved from http://proceedings.informingscience.org/InSITE2008/IISITv5p487-498Koohang462.pdf

Koohang, A., & Harman, K. (2005). Open source: A metaphor for e-learning. *Informing Science: The International Journal of an Emerging Transdiscipline, 8*, 75-86. Retrieved from http://inform.nu/Articles/Vol8/v8p075-086Kooh.pdf

Lakhani, K. R., & von Hippel, E. (2003). How open source software works: "Free" user-to-user assistance. *Research Policy, 32*(6), 923-943.

Lerner, J., & Tirole, J. (2002). Some simple economics of Open Source. *The Journal of Industrial Economics*, *L*(2), 197-234.

Lin, Y., & Zini, E. (2008). Free/libre open source software implementation in schools: Evidence from the field and implications for the future. *Computers & Education, 50*, 1092–1102.

Liu, C. (2003). Adopting open-source software engineering in computer science education. *The Third Workshop on Open Source Software Engineering*, Portland, Oregon, USA, May 3, 2003.

Liu, C. (2005). Enriching software engineering courses with service-learning projects and the open-source approach. *Proceedings of the 27th international Conference on Software Engineering, ICSE '05*. ACM Press, pp. 613-614.

Marengo, A., Baldassarre, M., & Pagano, A. (2008). An Open Source LMS, vision, and development: The development of innovative web technologies for e-learning: the eLearning portal project. *Proceedings of the Informing Science & IT Education Conference (InSITE) 2008*. Varna, Bulgaria, pp 390-401. Retrieved from http://proceedings.informingscience.org/InSITE2008/InSITE08p389-401Marengo487.pdf

Megias, D., Serra, J., & Macau, R. (2005). An international master programme in Free Software in the European higher education space. *Proceedings of the First International Conference on Open Source Systems (OSS 2005)*, Genova, Italy, 11th-15th July 2005, pp. 349-352.

NetCraft. (2008). *Web server survey*. Retrieved on August 28, 2008 from http://news.netcraft.com/archives/web_server_survey.html

Ozel, B., Cilingir, B., & Erkan, K. (Eds). (2006). Towards Open Source Software adoption and dissemination. *Proceedings of OSS 2006 tOSSad Workshop*, Como, Italy, pp.79-88

Perens, B. (1999). The open source definition. In C. DiBona, S. Ockman, & M. Stone (Eds.), *Open sources: Voices from the open source revolution*. Sebastopol, CA: O'Reilly

Raymond, S. E. (1999). *The cathedral and the bazaar: Musings on Linux and Open Source by an accidental revolutionary*. Sebastopol, CA, USA: O'Reilly.

Schrage, M. (2000). *Serious play: How the world's best companies simulate to innovate*. Boston, USA: Harvard Business School Press.

Shah, S. (2006), Motivation, governance, and the viability of hybrid forms of open source development, *Management Science, 52*(7), 1000-1014.

Sowe, S. K., Karoulis, A., Stamelos, I., & Bleris, G. L. (2004). Free/Open Source Software learning community and web-based technologies. *IEEE Learning Technology Newsletter*, *6*(1), 26-29.

Sowe, S. K., & Stamelos, I. G. (2007). Involving software engineering students in Open Source Software projects: Experiences from a pilot Study. *Journal of Information Systems Education, 18*(4), 425-436.

Sowe, S. K., Stamelos, I., & Deligiannis, I. (2006). Framework for teaching software testing using F/OSS methodology. In E. Damiani, B. Fitzerald, W. Scacchi, M. Scott, & G. Succi, (Eds.), *IFIP International Federation for Information Processing, Open Source Systems* (Vol. 203, pp. 261-266). Boston: Springer.

Spinellis, D. (2006). Open Source and professional advancement. *IEEE Software*, September|October, 70-71.

Surran, M. (2003). Making the switch to Open Source Software. *T H E Journal*, *31*(2), 36-40.

Warsta, J., & Abrahamsson, P. (2003). Is open source software development essentially an agile method? *The Third Workshop on Open Source Software Engineering*, Portland, Oregon, USA, May 3, 2003.

Watson, R., Boudreau, M., York, P., Greiner, M., & Wynn, D. (2008a). Opening the classroom. *Journal of Information Systems Education*, *19*(1), 75-85.

Watson, R., Boudreau, M., York, P., Greiner, M., & Wynn, D. (2008b). The business of Open Source. *Communications of ACM*, *51*(4), 41-46.

Weiss, M., Moroiu, G., & Zhao, P. (2006). Evolution of open source communities. *The Second International Conference on Open Source Systems (OSS 2006)*. Como, Italy. June 8-10, 2006.

Appendix: Survey Instrument

Examine Open Source Software (OSS) Developer's Career Concern From a Managerial Perspective

Survey Questions:

1. Your experiences with Open Source Software (OSS) developers and OSS software

Are you familiar with the term Open Source Software (OSS)?
(1) Not familiar (2) Somewhat familiar   (3) Very familiar

Have you ever worked with any developer with some OSS experiences?
(1) Yes (2) No

Have you ever used a developer's OSS experiences to evaluate his/her abilities and qualifications?
(1) Yes (2) No

2. On a scale 1-7 (strongly disagree-strongly agree), how much do you agree or disagree with the following statement with regard to the developer's OSS experiences in influencing the hiring and promotion process?

1-Strongly Disagree   2-Disagree   3- Slightly Disagree   4- Neither disagree nor agree
5-Slightly Agree   6-Agree   7-Strongly Agree

(1) OSS experience could have a positive influence in the hiring decision
(2) OSS experience could have a positive influence in the promotion decision
(3) OSS experience could be a good indicator of the developer's ability
(4) To show a developer's ability, OSS experience can be as effective as the experiences in the proprietary software development
(5) In evaluating a developer's ability, OSS experience can be as effective as a diploma or certificate

3. On a scale 1-7 (strongly disagree-strongly agree), how much do you agree or disagree that the following factors are important to consider during the hiring process:
(1) How many years of the developer's OSS experience
(2) The programming language used by the developer in the OSS project
(3) The operating systems used by the developer in the OSS projects
(4) The skills the developer learned in the OSS experiences are supported or compatible with your company's existing or future projects
(5) The skills the developer learned in the OSS experiences are used by your company's current and/or potential competitor

4. On a scale 1-7 (strongly disagree-strongly agree), how much do you agree or disagree that the following OSS activities are important indicators of the developer's abilities?
(1) Programming and code writing
(2) Architecting the projects
(3) Project documentation
(4) Project testing and debugging

(5) Initiating and leading the project development
(6) Coordinating development tasks among developers
(7) Promoting and advocating projects in the community
(8) Consulting and supporting the clients
(9) Recruiting more developers

5. On a scale 1-7 (strongly disagree-strongly agree), how much do you agree or disagree that the following OSS project characteristics could have positive influence on the evaluation and hiring process of an OSS project developer?
(1) The OSS project's popularity
(2) The project's customer base fits the customer base of your company
(3) The project's topic fits those of your company's projects

6. Demographic information
(1) How long have you worked in the industry? _____ Years.

(2) What is your position in the company, please choose one:
(a) Architect/Developer   (b) CIO or CTO   (c) Product Manager
(d) Researcher of R&D   (e) Other, please specify_____

(3) Do you participate in the developer hiring process?
(a) Yes   (b) No

(4) Do you participate in the process of evaluating a developer's ability?
(a) Yes   (b) No

(5) Does your company produce:
(a) Proprietary software   (b) Open source software   (c) Shareware
(d) Mixed   (e) None of the above

Biography

Ju Long is an assistant professor of Computer Information Systems at the McCoy College of Business at Texas State University. She has a Ph.D. degree from the University of Texas at Austin. Her research interests include reward system in information technology innovation; open source software development, wireless computing and information technology and its impact on health care industry.