

# The Culture of Open Source Computing

By Andrea Holliger, University of Virginia



# The Culture of Open Source Computing

An Annotated Bibliography

By Andrea Holliger, University of Virginia

For the National Center for Women & Information Technology

While women's representation in computing is low, it is even lower in open source computing – only 1.5% of all Open Source Software (OSS) developers are women (Nafus, *et al.* 2006). OSS is collaboratively developed and maintained by volunteers and professionals who communicate primarily through web-based mailing lists and Internet Relay Chat. OSS is distinct from proprietary software because the software source code is available to the public, who may alter it for their personal use. Research suggests that barriers to women's participation may be deeply imbedded in the culture of OSS.

As a first step toward learning more about OSS culture and women's participation in it, this annotated bibliography briefly describes current research organized into five topics: Gender Dimensions, Entry & Internal Advancement, Knowledge Acquisition, Membership and Organization, and Motivations & Intentions to Participate. This bibliography identifies pertinent articles and offers a brief summary of what are, in many cases, extensive research findings, only two of which focus on gender and OSS. The original publications should be consulted for full details. We will evaluate these materials and their implications for further research in a subsequent literature review.

We located articles by first consulting with members of the National Center for Women & Information Technology for known sources of information on women in OSS. Next, with assistance from a research librarian, we searched the Sociological Abstracts, Web of Science, PAIS International, Google Scholar and Dissertation Abstract databases using the search phrases: "open source" + comput\*, and more specifically, "open source" and ((software or computing) and (developers or programmers)). Abstracts were then

consulted to determine whether the article was scholarly in nature and concerned with OSS developers, their community, and culture (as opposed to technical evaluations of OSS, or opinion pieces about the culture). References contained within each relevant article were also evaluated using the same criteria. Finally, using the Web of Science and GoogleScholar databases, articles citing those already selected were viewed and evaluated. These methods yielded the following 26 articles.

## GENDER DIMENSIONS OF OSS

Lin, Y. (2005). *Gender Dimensions of Floss Development*. *Mute Magazine*. Retrieved July, 27, 2007, from <http://www.metamute.org/en/node/5596/print>

Lin's encyclopedic article discusses issues of gender discrimination in Free/Libre Open Source Software (FLOSS) development. Lin uses existing research, including her own previous work, to theorize about the reasons for the gender discrepancy in OSS computing. She posits that women are excluded implicitly and explicitly from becoming FLOSS developers, and that their needs as users are not addressed. Lin theorizes that the long hours necessary for coding, a lack of female role models and mentors, users' discriminatory language on and offline, the prevalence of text-based coding systems (as opposed to graphic coding environments), and the FLOSS community's male-centric competitive world-view are all factors hindering women's participation.

Nafus, D., Leach, J., & Krieger, B. (2006). *Free/Libre and Open Source Software: Policy Support (Integrated Report of Findings)*. Cambridge: FLOSSPOLs.

This detailed report analyzes the Ghosh *et al.* (2002) survey<sup>1</sup> and additional ethnographic information about the gender

<sup>1</sup> Ghosh, R. A., Glott, R., Krieger, B., & Robles, G. (2002). *Free/Libre and Open Source Software: Survey and Study*. FLOSS. Deliverable D18: Final Report, Part IV: Survey of Developers. University of Maastricht, The Netherlands: International Institute of Infonomics. Ghosh *et al.* conducted an extensive survey of 2784 OSS participants, which investigated "fundamental features" and "economic principles" of the OSS community (p. 4). This document contains their findings, which are then analyzed by Nafus *et al.*

<sup>2</sup> Nafus *et al.* (2006) define "flame wars" as extended periods of "inflammatory talk and aggressive posturing" (p. 6).

dimensions of FLOSS. Using data collected in 2004–2005, the authors conclude that women are deterred from entering FLOSS because of the combative hacker ethic, including “flame wars<sup>2</sup>,” and the difficulty of receiving adequate recognition for their contributions. This report also suggests that women are hindered from joining FLOSS communities because they are less likely than men to have the level of computing expertise the FLOSS community expects of new entrants — women generally first engage with computers at a later age or at a less advanced level compared to men.

## ENTRY AND INTERNAL ADVANCEMENT

von Krogh, G., Spaeth, S., & Lakhani, K. (2003). Community, joining, and specialization in open source software innovation: a case study. *Research Policy*, 32, 1217–1241.

Von Krogh et al. developed an inductive theory of OSS innovation and investigated the process of joining the *Freenet*<sup>3</sup> OSS community. Analyzing data from 13 interviews with eight developers, developer email lists, and the CVS repository<sup>4</sup> makes explicit the unwritten “scripts” or rules that must be followed to join the community successfully. For example, the community requires joiners to demonstrate previous experience and expertise.

Herraiz, I., Robles, G., Amor, J. J., Romera, T., & Gonzalez Barahona, J. M. (2006). *The Processes of Joining in Global Distributed Software Projects*. Paper presented at the Global Software Development for the Practitioner (GSD), Shanghai, China.

Herraiz et al. used the *GNOME CVS*<sup>5</sup> repository to investigate whether hired developers and volunteers follow different paths to become core developers<sup>6</sup>. They conclude that there is not a consistent “joining pattern” for both groups. Instead, volunteers follow the “onion” model<sup>7</sup> and take approximately 30 months to become core developers,

whereas hired developers achieve this status much earlier without following the onion model.

Bird, C., Gourley, A., Devanbu, P., Swaminathan, A., & Hsu, G. (2007). Open Borders? Immigration in Open Source Projects. Paper presented at the Fourth International Workshop on Mining Software Repositories.

In this brief conference paper, Bird et al. consider the length of time between an individual’s first involvement with a project mailing list and their first accepted code contribution. Their analysis of data mined from *Apache*, *Postgres*, and *Python* mailing lists finds that reputation as a programmer positively influences becoming a core developer. Other hypotheses — (1) that the likelihood of becoming a core developer will rise with time, peak, then decline, and (2) that demonstration of skill increases the likelihood of becoming a core developer — were only partially supported.

Stewart, D. (2005). Social Status in an Open-Source Community. *American Sociological Review*, 70, 823-842.

Stewart performs an empirical analysis of the peer-certification system at *Advogato.org*<sup>8</sup> to evaluate how the number and status of certificate-givers affect an individual’s likelihood of receiving additional peer certificates. Stewart uses open-source computing to establish a model of status attainment in society at large. Pertaining specifically to OSS, Stewart concludes that receiving positive certificates does not necessarily correspond to programming skill. Users tend to give certificates consistent with those already received, especially if other certificate-givers have high status. Reciprocity and collaboration were strong predictors that an individual would receive a certificate from another user. Stewart concludes that developers do not necessarily attain status by *being* good developers, but by having others say that they are good developers, though he collected no evidence to suggest that those with a high status did not deserve that status.

---

<sup>3</sup>*Freenet* is an OSS project. Hereafter, the names of OSS projects will appear in italics.

<sup>4</sup>The CVS Repository is a database containing the official source code for a project which catalogues all changes made to that code, including by whom and when they were made.

<sup>5</sup>ibid

<sup>6</sup>A core developer is defined as an individual allowed to change the official source code of an OSS product.

<sup>7</sup>The “onion” model proposes that developers begin as observers, then progress through a series of increasingly central functions (bug reporters, bug fixers, etc.) until they become core developers.

<sup>8</sup>*Advogato.org* is an online community of OSS developers. Any developer may give a peer-recognition certificate to any other developer by using a pull-down menu on the user’s *Advogato* page and selecting from four hierarchical ranks.

Jensen, C., & Scacchi, W. (2007). *Role Migration and Advancement Processes in OSSD Projects: A Comparative Case Study*. Paper presented at the 29th International Conference on Software Engineering (ICSE).

Jensen & Scacchi conduct a comparative case study of *Mozilla*, *Apache*, and *NetBeans*, and perform an empirical analysis of user movement from observer to core developer. They find multiple methods of initial involvement past the observation stage, including submitting bug reports and source code. Like Herraiz *et al.*, they determine that there are different paths to core developer status, suggesting that while OSS community structure is hierarchical, it is also fluid. Analysis also finds that the majority of core developers are full-time paid employees of corporate or non-profit organizations, not volunteers as is supposed in much OSS literature.

## KNOWLEDGE ACQUISITION

Hemetsberger, A., & Reinhardt, C. (2006). Learning and Knowledge-building in Open-source Communities: A Social-experiential Approach. *Management Learning*, 37, 187-214.

Hemetsberger and Reinhardt present empirical evidence based on an interpretive investigation of individual and community learning processes in the *K Desktop Environment* community. They find that the primary method of learning is “re-experience” — archives of code changes and mailing lists allow new users to re-experience what others have previously learned. These archives create a “virtual transactive group memory” that exists independently of individual users, and therefore facilitates building and perpetuating community knowledge.

## MEMBERSHIP & ORGANIZATION

Xu, J., Gao, Y., Christley, S., & Madey, G. (2005). *A Topological Analysis of the Open Source Software Development Community*. Paper presented at the 38th Hawaii International Conference on System Sciences.

Xu *et al.* perform a quantitative analysis of the entire community of OSS developers at *SourceForge*<sup>9</sup>. They

find that approximately 65% of *SourceForge* users are “passive,” making no discernable contributions to project development. They also find a “small world phenomenon” suggesting that all developers can be linked to each other by their connections with other developers, and that large and small projects have different community distributions. For example, small projects have a greater percentage of core developers.

Krishnamurthy, S. (2002, June). Cave or Community? An Empirical Examination of 100 Mature Open Source Projects. *First Monday*, 7.

This paper briefly reports on Krishnamurthy’s investigation into whether OSS developers tend to work singly or in communities. His study of 100 successful projects at *SourceForge* finds that most OSS projects are conducted in very small groups, even singly (the average was 4 and the mode was 1). Krishnamurthy additionally finds that products with more developers are viewed and downloaded more often, that larger projects have a smaller percentage of project administrators, and that the age of the project is not related to the number of developers.

Crowston, K., & Howison, J. (2005). The Social Structure of Free and Open Source Software Development. *First Monday*, 10.

Crowston and Howison test the validity of generalizing about communication style between OSS projects by examining the bug reports of 120 projects hosted on *SourceForge*. Their statistical analysis finds that not all projects’ communications are equally centralized. They find that small projects are more centralized than large, but additionally suggest that large projects may function as a network of smaller projects. They conclude that one cannot assume OSS projects will inherit the characteristics found in other OSS case studies and research.

---

<sup>9</sup>*SourceForge.net* is a server that hosts and provides tools for multiple OSS projects.

## MOTIVATIONS AND INTENTIONS TO PARTICIPATE

Raymond, E. S. (1998). *Homesteading the Noosphere*. Retrieved 8/10/2007, 2007, from [http://www.firstmonday.org/issues/issue3\\_10/raymond/index.html](http://www.firstmonday.org/issues/issue3_10/raymond/index.html)

Raymond uses his involvement with the OSS community to theoretically examine the customs regulating ownership and control of OSS contributions. He famously proposes that OSS communities are “gift cultures”, i.e. that developers make a gift of their source code, expecting reciprocation by others’ code contributions.

Lerner, J., & Tirole, J. (2000). *The Simple Economics of Open Source*. Unpublished manuscript, Cambridge, MA.

Lerner and Tirole challenge Raymond’s suggestion that OSS developers are motivated primarily by altruism (the “gift culture”). They theorize that developers are externally motivated by *career concerns* and *ego gratification*. They then group these motivating factors under the heading signaling incentives because in both cases developers use their OSS work to signify their talents in the hopes of career advancement and peer recognition, respectively.

Hars, A., & Ou, S. (2001). *Working for Free? Motivations of Participating in Open Source Projects*. Paper presented at the 34th Hawaii International Conference on System Sciences.

Hars and Ou’s survey of 81 OSS project participants (21% response rate) finds that motivations differ among participants, but that external motivations (career advancement, self-marketing, etc.) have greater weight than internal motivations (altruism, community identification, etc.). They furthermore find that these motivations correspond to different types of developers: Students and hobbyists are more internally motivated, whereas salaried programmers are more externally motivated.

Hann, I.-H., Roberts, J., Slaughter, S., & Fielding, R. T. (2002). *Why Do Developers Contribute to Open Source Projects? First Evidence of Economic Incentives*. Paper presented at the Second Workshop on Open-Source Software Engineering, Orlando, FL.

This conference paper briefly discusses the preliminary results of an investigation of *Apache* project archives and a targeted survey of *Apache* participants. Preliminary results indicate that employers do not reward participants for their OSS experiences. They do find, however, that those with a higher rank in the project average higher salaries, suggesting that OSS project rank may signal a developer as an above-average programmer.

Lakhani, K., & von Hippel, E. (2003). How Open Source Software Works: “Free” User-to-User Assistance. *Research Policy*, 32, 923-943.

Lakhani and von Hippel build upon Lakhani’s 1999 research into the *Apache* user-to-user assistance system to empirically investigate the motivations to provide this assistance for free. Researchers examine interviews with several assistance-providers, as well as a four-year period of user-to-user assistance postings. They determine that users must spend a considerable time scanning the user-to-user help archives in order to find questions they can answer, but that 98% of this time is spent reading previously answered questions and deriving direct learning benefits from them. Therefore, they state that 98% of the effort expended in providing assistance returns a direct learning benefit, and that this learning benefit is the primary motivation behind providing user-to-user assistance.

Hertel, G., Niedner, S., & Herrmann, S. (2003). Motivation of Software Developers in Open Source Projects: an Internet-based Survey of Contributors to the Linux Kernel. *Research Policy*, 32, 1159-1177.

This article examines the motivations of 141 *Linux* developers predicated on an internet-based survey and utilizing two pre-existing motivational models: 1) voluntary action for social movements and 2) individuals who work in small teams. They find that *Linux* developers are motivated by their identification with the *Linux* community and pragmatic desires for software improvement. They also

find that developers are more likely to participate if they evaluate the team's goal highly and perceive themselves as indispensable to the project. These motivations and determinants are considered similar to motivational processes in other social communities and can be understood using the existing models.

**Zeitlyn, D. (2003). Gift Economies in the Development of Open Source Software: Anthropological Reflections. *Research Policy*, 32, 1287–1291.**

In this brief article and call for further research, Zeitlyn builds upon Eric Raymond's work and discusses the motivation of OSS developers, theorizing that OSS communities resemble kinship groups (extended families) and develop kinship-amity (defined as family-like affection for those one is not actually related to). He suggests that this model refines Raymond's theory of a "gift culture," by accounting for the fact that gift-giving is not always directly reciprocated in OSS communities.

**Lakhani, K., & Wolf, R. G. (2005). Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. In J. Feller, B. Fitzgerald, S. Hissam & K. Lakhani (Eds.), *Perspectives on Free and Open Source Software* (pp. 3-21). Cambridge: MIT Press.**

The results of Lakhani *et al.*'s web-based survey of 684 OSS developers finds that enjoyment-based intrinsic motivation (feeling creative and the intellectual stimulation of writing code) is the strongest and most pervasive motivator. This motivation is followed by user need and improving programming skills. Results also find, however, that no one motivator has more than 50% importance, suggesting that the OSS community is heterogeneously motivated by an interplay of extrinsic and intrinsic motivations. They find that the motivations suggested by Raymond (2001), Lerner, and Tirole (2002), such as community reputation, professional status, and antagonism toward closed source software, rank relatively low.

**Ratto, M. (2005). "Don't Fear the Penguins": Negotiating the Trans-local Space of Linux Development. *Current Anthropology*, 46(5), 827-834.**

This anthropological investigation suggests that *Linux* developers create a multi-national, dispersed, but strong community based upon their relationship to *Linux* itself. This relationship is facilitated by "Tux" the *Linux* mascot, which operates as a locus of identification and affection.

**Bagozzi, R. P., & Dholakia, U. M. (2006). Open Source Software User Communities: A Study of Participation in Linux User Groups. *Management Science*, 52(7), 1099–1115.**

Bagozzi and Dholaki take a social-psychological perspective and perform statistical analysis of data from an internet-based survey conducted with 402 active members of international LUGs (*Linux* user groups). They find that intentions to participate in LUGs are positively associated with attitudes toward LUG participation, how difficult or easy participation is viewed to be, negative anticipated emotion from not being able to participate, identification as "belonging" to the LUG group, and identification with the Open Source movement. They also find that social identity and joint interactions with other LUG members have an increasing impact over time, with these factors contributing more for experienced members.

**Shah, S. (2006). Motivation, Governance, and the Viability of Hybrid Forms in Open Source Software Development. *Management Science*, 52(7), 1000–1014.**

Shah analyzes interviews and publicly available information to determine how differences in governance affect developers' motivations and the quality of their contributions. Shah finds that there are two types of developers: need-driven and hobbyist. Need-driven participants are motivated by software needs and feelings of reciprocity. Hobbyists participate because they enjoy creating code. Shah also finds that outcomes of OSS participation do not always match the initial motivations for joining. For example, Shah cites developers who have advanced their careers by OSS participation were not initially motivated by this factor.

Roberts, J. A., Hann, I.-H., & Slaughter, S. (2006). Understanding the Motivations, Participation, and Performance of Open Source Software Developers: A Longitudinal Study of the Apache Projects. *Management Science*, 52(7), 984–999.

Roberts *et al.* develop a theoretical model and evaluate it with survey and archival data from a longitudinal field study of developers in *Apache*. They find that contributors have multiple motivations, some complementary, others not; and find no evidence that extrinsic motivations preclude strictly intrinsic motivations. They instead find that extrinsic status motivations *enhance* intrinsic motivations, and that paid contribution positively relates to status motivations. They therefore conclude that paid developers are a positive influence upon OSS communities because developers with higher status motivations are more substantive contributors. They also find that feedback systems increase status motivations, and confirm that the *Apache* meritocracy is operating effectively: promotions within the community are based upon actual contributions.

Li, Y., Tan, C.-H., Teo, H.-H., & Mattar, A. T. (2006, April 13-15). *Motivating Open Source Software Developers: Influence of Transformational and Transactional Leaderships*. Paper presented at the Special Interest Group on Computer Personnel Research Annual Conference, Claremont, CA.

*Li et al.* develop a research model and test it on a survey of 118 OSS developers on *SourceForge*. They seek to determine how project leaders affect developers' motivations to continue. They find that transformational leadership (the use of idealized behavior, inspirational motivation, individualized consideration and intellectual stimulation) influences intrinsic motivation, while transactional leadership (the use of active or passive management by exception, and clear reward and punishment structures) affects extrinsic motivations.

Bitzer, J., Schrettl, W., & Schroder, P. J. H. (2007). Intrinsic Motivation in Open Source Software Development. *Journal of Comparative Economics*, 35, 160–169.

Bitzer *et al.* examine OSS using an adapted "private-provision-of-public-goods model." Using existing empirical evidence from other researchers, they determine that intrinsic motives like software needs, the fun of coding, and a gift culture are incorporated simultaneously. They further find that OSS developers are more patient, younger, and more efficient; derive a high value from playing with software and mastering challenges; and derive a higher gain from using OSS software solutions than the general population.

Wu, C.-G., Gerlach, J. H., & Young, C. E. (2007). An Empirical Analysis of Open Source Software Developers' Motivations and Continuance Intentions. *Information & Management*, 44, 253–262.

Following an empirically validated research model, researchers use data from a field survey of 148 OSS participants to evaluate developers' motivations for contributing and their intentions to continue involvement. The data show that participants' intentions to continue are most strongly influenced by satisfaction with OSS participation, followed by their desire to enhance personal skills and capabilities (human capital), and the satisfaction of personal software needs. This research supports the theory of OSS as a gift culture. The researchers also find that career advancement opportunities derived from OSS development enhance satisfaction, and indirectly influence the intention to continue participating.



National Center for Women & Information Technology (NCWIT)  
University of Colorado  
Campus Box 322  
Boulder, CO 80309-0322  
[www.ncwit.org](http://www.ncwit.org)  
[info@ncwit.org](mailto:info@ncwit.org)

We thank our investment partners: the National Science Foundation, Avaya, Microsoft, and Pfizer

This bibliography was produced with assistance from Joanne McGrath Cohoon at the University of Virginia, Debra Richardson at the University of California at Irvine, the NCWIT Leadership Team, and NCWIT staff social scientists.

© 2007 by the National Center for Women & Information Technology (NCWIT)  
This booklet may not be reproduced in any form without prior written permission of the National Center for Women & Information Technology.