



# Beyond Development: A Research Agenda for Investigating Open Source Software User Communities

Leigh Jin

San Francisco State University, USA, [jinlei@sfsu.edu](mailto:jinlei@sfsu.edu)

Daniel Robey

Georgia State University, USA, [drobey@gsu.edu](mailto:drobey@gsu.edu)

Marie-Claude Boudreau

University of Georgia, USA, [mcboudre@terry.uga.edu](mailto:mcboudre@terry.uga.edu)

In the last few years, the open source software (OSS) development movement has captured the attention of both information systems practitioners and researchers. In contrast to proprietary software, OSS is usually developed through public collaboration and its source code is made freely available. In the last five years, OSS development has become a viable alternative to commercial software (Chengalur-Smith and Sidorova, 2003), attracting intense practitioner interest. As a new approach to the production of software, OSS has already begun to revolutionize the software industry as a whole, drastically changing the way software code is produced. Beyond the software industry, OSS has attracted interest for its application of community principles of governance over commercial activities (Markus et al., 2000; von Hippel and von Krogh, 2003). Indeed, by describing OSS development as a "movement," we reflect the broader excitement about the implications of community governance processes in a knowledge economy (Adler, 2001).

Open source has rapidly become a popular area of study within the information systems (IS) research community, as evidenced by the appearance of special tracks for OSS within conferences and special issues of journals. The vast majority of the research conducted so far has focused on the phenomenon of OSS *development* (Fitzgerald and Kenny, 2003). The interest in OSS development reflects a desire to explain the counterintuitive practice of treating commercially valuable products as public goods rather than proprietary products for sale. Likewise, the development and maintenance of complex software products by communities of expert volunteers has piqued interest into the incentives for developers.

As a consequence of the primary focus on OSS development, little research has yet been conducted on OSS *use*. The neglect of OSS use may be attributed to two assumptions about OSS projects. First, it is known that people often become OSS developers because they intend to use the product being developed. To echo Raymond's (2001) frequently quoted expression, OSS developers are users with an "itch to scratch," so they are willing to devote time and expertise to develop software solutions to their own problems as users. Thus, if it is assumed that all OSS developers are users, making a distinction between developers and users becomes unnecessary (Feller and Fitzgerald, 2000). In other words, if OSS use is assumed as the primary motivation for OSS development, research on OSS use is redundant with research on OSS development.

This assumption can be challenged by statistics showing the rapid rise in the number of users, the vast majority of whom have no interest or capability to contribute to modifications of the source code (Fitzgerald

and Kenny, 2003). For widely distributed OSS such as Linux, it makes no sense to assume that all users could possibly be developers (von Hippel and von Krogh, 2003). Rather, it is clear that there are proportionately more users than developers. Moreover, as OSS development becomes increasingly targeted toward productivity and entertainment applications, an increasing number of non-experts are becoming OSS users.

The second assumption discouraging research on OSS use is that the OSS movement is unique solely because of the way software is being developed, but that its use is similar to any other type of software. Given that an abundance of IS research has focused on the adoption and use of software applications, therefore, it might be assumed that no further investigation is necessary for OSS products.

This assumption can be challenged by looking at the differences between OSS and proprietary software. Users of OSS are typically confronted by a fundamentally different type of technical support than in proprietary software. Rather than relying on a vendor's customer support, users of OSS need to find other sources of help for installing, learning, and using their freely acquired software. Perhaps OSS users receive such help through participation in user groups that are supported by community volunteers, similar to the communities supporting development. Given the paucity of research on OSS use, it is important to keep an open mind regarding OSS use and to formulate a program of research rather than to assume that "use is use."

These arguments justify research into OSS use. In this paper, we adopt a community perspective on OSS use, which is explained in the following section. We then present a framework that includes four main areas of investigation: *creation* of OSS user communities, their *characteristics*, their *contributions*, and how they *change*. For each element of the framework we pose several research questions.

## A COMMUNITY PERSPECTIVE ON OSS USE

The term "community" was introduced into the English language in the 14<sup>th</sup> century from Latin to refer to a group of people living in a common geographical location. Only between the 17<sup>th</sup> and 19<sup>th</sup> centuries was the meaning of community extended to describe people who share common characteristics, interests or identities — even if they are not geographically close. As the 21<sup>st</sup> century begins, people have grown more accustomed to participating in "virtual communities" that are enabled by Internet technology and the World Wide Web (Shumar and Renninger, 2002). Virtual communities differ from co-located communities by offering a wider range of options for participation and by allowing

community size to grow, unconstrained by physical space. Individual members may tailor their virtual communities to satisfy personal preferences (Wellman, 2001).

OSS development has certainly depended in large part upon the ability of developers to contribute as members of virtual communities. Given that much of the OSS development transpires in computer mediated online communities, we anticipate that OSS use would also rely on virtual communities for software acquisition, implementation, maintenance, and support. For example, Lakhani and von Hippel (2003) suggest that successful open source projects are capable of delivering high quality “field support” – mundane but necessary tasks — to users through voluntary effort. Field support primarily involves experienced users answering questions posted by new users through the archived mailing list. Indeed, a highly organized system of OSS user groups has sprung up around the major OSS products. Taking Linux user groups (LUGs) as an example, there are currently 829 registered LUGs in 105 countries, including 274 in the United States<sup>1</sup>.

Despite the importance of electronically mediated interaction within OSS communities, we do not assume that OSS user communities are exclusively virtual. Indeed, one of the naïve assertions about OSS development is that software can be developed by a community of complete strangers who interact only through electronic media. To the contrary, experienced OSS participants have opportunities to attend conferences as well as regular meetings held in physical places. For example, O'Reilly Open Source Convention and LinuxWorld Conferences are popular places for open source developers to meet and exchange ideas. The Silicon Valley Linux User Group, which claims to be the oldest and one of the largest LUG in the world, holds face-to-face at least monthly, inviting distinguished or interesting speakers. Besides regular meetings, LUGs also organize InstallFests, where new users can bring in their computers and allow experienced volunteers to install Linux, diagnose problems, and repair configurations. The LUG of Davis runs a Linux Emergency Relief Team, staffed with competent Linux experts who even travel to new users' homes to give their Linux computers personal attention. According to Moen (2003), LUGs are vital to the Linux movement, taking on many of the same roles that a regional office does for a large organization:

*LUGs' role in Linux advocacy cannot be overestimated, especially since wide-scale commercial acceptance of Linux is only newly underway. While it is certainly beneficial to the Linux movement each and every time a computer journalist writes a positive review of Linux, it is also beneficial every time satisfied Linux users brief their friends, colleagues, employees, or employers.*

The research agenda that follows focuses on questions about the creation, characteristics, contributions and change in OSS user communities. The agenda is proposed at a high level due to the novelty of the phenomenon and the paucity of existing research efforts. By restricting our attention to a community perspective, we purposefully omit consideration of individual and organizational influences on OSS use. However, we believe that a community perspective on OSS research is valuable because it has played a prominent role in research on OSS development.

## A RESEARCH AGENDA

Figure 1 identifies the four main areas where research into OSS user communities should be undertaken: creation (C1), characteristics (C2), contributions (C3), and change (C4). As shown, these areas are related sequentially, beginning with creation. It is also shown that change not only completes the cycle but begins a new cycle.

### C1: Creation of OSS User Communities

The creation of an OSS user community requires potential members to learn about OSS as an alternative to proprietary software and in-house

development, actually engage in community activities, and respond to incentives for starting or joining a user community. The research questions below address these three important issues.

**C1-1: How do new users, especially technically disadvantaged users, learn about OSS alternatives to proprietary software?** This issue is interesting because, compared to proprietary software, OSS projects lack specialized teams to market the product and to promote it through mass media. Although the notion of a “gift culture” is well established within the OSS development community, users are more likely to be suspicious of OSS or simply not learn about it. Although studies of established user communities may focus on the incentives for community participation, users must first see OSS as a viable option.

**C1-2: How are OSS user groups created?** Traditional software user groups are often sponsored by software's vendors. Generally, proprietary software user groups maintain constructive liaison with their vendors, which can provide considerable benefits to both parties (Buckner, 1996). On the one hand, vendors receive a low cost marketing opportunity and get feed-back on the usability of their product. On the other hand, users may benefit from discounted prices negotiated with the vendors (in the short term) and the chance to make suggestions for improvement in the software's next release (in the long term).

Because no software vendor is involved in OSS user groups, their creation may depend on a group of like-minded enthusiasts sharing the same passion for the software. These enthusiasts may be among the original developers, motivated by increasing the software's user base. Or, these enthusiasts may be pure users, interested in maximizing their return on using the software. Large companies may also be the principal instigators of OSS user groups. If an OSS product is used extensively within the company, creation of a user group for that product provides “free” education opportunities for their employees.

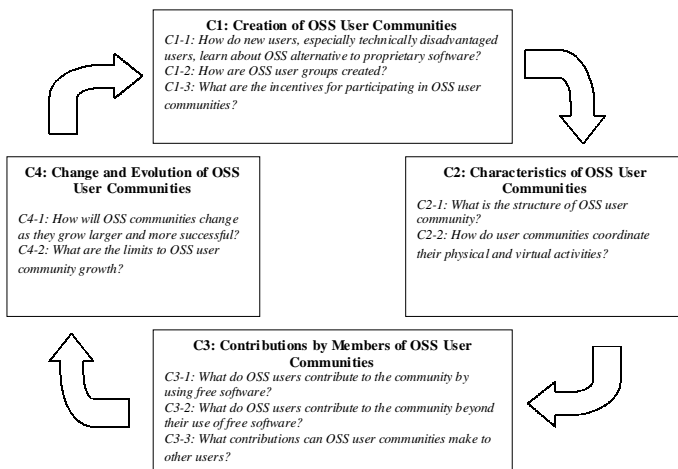
**C1-3: What are the incentives for participating in OSS user communities?** Because users may obtain OSS freely with no obligation to contribute to development, their use of the software is likely to be based primarily on cost and quality considerations (Fitzgerald and Kenney, 2003). The incentives for *community participation*, however, differ from the incentives for using OSS. For developers who incur substantial private costs by investing their own resources into development, incentives include the ability to use the software but also benefits related to reputation and learning. It is conceivable that users may also obtain such benefits, gaining reputations as skilled implementers who are helpful to novice users.

It is argued that the true benefit for using open source software is beyond low initial cost but rather “long-term control over IT”. The users of proprietary software who become dependent on software that they are not allowed to see inside, let alone change, have lost control of IT by subjecting themselves to a monopoly relationship with their vendor (Moen, 2000).

It is also conceivable that OSS user groups offer more to users than user groups organized by vendors. Vendor user groups sometimes charge fees for membership and use their meetings as opportunities to promote new products. These commercial interests do not exist in OSS user groups. For the more technically inclined, OSS user groups offer solutions in the form of code modifications that address specific problems of individual users. By contrast, vendors avoid short-term solutions involving code modification and focus on software configuration or settings, which may or may not solve the user's problems. As indicated by Moen (2003):

*Traditional groups must closely monitor what software users redistribute at meetings. While illegal copying of restricted proprietary software certainly occurred, it was officially discouraged — for good reason. At LUG meetings, however, that entire mindset simply does not apply: Far from being forbidden, unrestricted copying of Linux should be among a LUG's primary goals.*

Figure 1. Research Areas



## C2: Characteristics of OSS User Communities

As communities, OSS user groups are likely to be differentiated by the roles that different members of the community play. It is also important to recognize the relationships between user and developer communities. Indeed, it may be desirable to consider users and developers as sub-communities within an open source project. The questions below address these and other issues related to the characteristics of OSS user communities.

**C2-1: What is the structure of OSS user community?** An overall OSS community is built around a specific OSS project, with shared interests of improving and using the software. As more people get interested in using the software, the community grows and differentiates into various roles. For example, Ye et al. (2002) identified eight roles in OSS communities:

1. Project leader: project initiator, responsible for the overall direction of the project.
2. Core members: responsible for guiding and coordinating the development of an OSS project.
3. Active developers: regular contributors of new features and bug fixes.
4. Peripheral developers: occasional contributors of new functionality.
5. Bug fixers: fix bugs that either they discovered or that have been reported by bug reporters.
6. Bug reporters: discover and report bugs, assuming a role comparable to testers in traditional software development.
7. Readers: users who also try to understand how the system works by reading the source code.
8. Passive users: users who use the system without showing interest in how the system is constructed.

Of these roles, the last three would be involved primarily as users rather than developers. However, all roles include people likely to be members of the user community. An important issue deserving research attention is the way members assume these roles and the relationships among the various roles. Such research could contribute to more effective designs for community structure.

**C2-2: How do user communities coordinate their physical and virtual activities?** Given that OSS development communities operate both physically and virtually, it is an important question to understand how they use these different arenas of community life. Theories of virtual organizing have pointed to the possibility for virtual and physical activities to reinforce, complement, compensate and produce synergies with each other (Robey, Schwaig and Jin, 2003). On the one hand, OSS virtual activities allow developers to access and modify source code, to access necessary archives, and to

interact through a mailing list. On the other hand, OSS physical activities are important for different reasons. For example, Moen (2003) emphasize that LUGs physical “socializing” perspective is the most effective way of Linux acculturation:

*By “socializing”, here I mean primarily sharing experiences, forming friendships, and mutually-shared admiration and respect. In other words, acculturation turns you from “one of them” to “one of us”... LUGs are often much more efficient at this task than are mailing lists or newsgroups, precisely because of the former’s greater interactivity and personal focus.*

## C3: Contributions by Members of OSS User Communities

Much is made in the OSS development literature of the voluntary gifts donated by skilled designers to the creation of a public good (Fitzgerald and Kenney, 2003). Indeed, communities are likely to fail if such contributions are not made. The following questions are posed with the same issue in mind for the OSS user community.

**C3-1: What do OSS users contribute to the community by using free software?** If contributions are not made by users, OSS users assume the status of “free riders” who simply take from the community without paying back. Interestingly, free ridership by OSS users is actively encouraged rather than discouraged. Because the number of OSS users is a measure of a project’s success, users do not have the same pressures to contribute that developers do (Von Hippel and Von Krogh, 2003). Indeed, their most important contribution may simply be their use of the OSS product.

**C3-2: What do OSS users contribute to the community beyond their use of free software?** As Raymond (2001) pointed out, some of the most successful OSS projects are created by the most talented software developers. Because the OSS community tends to attract people with extensive technical backgrounds, there is a risk that resulting products would reflect the “geek” culture and be less useful to ordinary users. For example, the user interfaces of OSS tend to be command line driven, making even their installation and configuration very demanding technically. It is conceivable that less technical users might contribute to development by making OSS projects more user friendly: easier to install, configure, use and maintain. Whether such participation would be welcomed by core developers remains uncertain, so the issue presents a good research opportunity to study the impact of less talented users on the development process.

**C3-3: What contributions can OSS user communities make to other users?** Fitzgerald and Kenney (2003) report an interesting case of an Irish hospital using OSS software for a number of internal operations. Although the hospital’s IT staff had no intention of ever contributing modifications to the software’s source code, they had begun to offer the applications, which they had tailored for themselves, to other health care organizations, free of charge. In this manifestation of community spirit, one user was “giving back” to the community of other users. The study suggests that users may add further value by making OSS programs fit specific industry needs. While these contributions may not earn great reputations, they may provide value for the user community. Future research is warranted on the practice of users making vertical applications more useful for other users, in contrast to the traditional focus in OSS development on horizontal infrastructure systems (Fitzgerald and Kenny, 2003).

## C4: Change and Evolution of OSS User Communities

It is clear that OSS communities are new phenomena that have only become significant economically in the last half decade. We expect the nature of OSS communities to change, perhaps rapidly, as software development and use practices continue to evolve. This, in turn, will affect the creation of new communities, as our cyclical representation in Figure 1 suggests. The following questions address the evolution and change of OSS user communities.

**C4-1: How will OSS communities change as they grow larger and more successful?** Although core developers initiate and contribute the major portion of source code (e.g., 80% in case of the APACHE project (Mockus, Fielding et al., 2002)), the largest growth in community size comes from supporting roles like bug fixers, bug reporters and end users. Core developers may even recede in importance once a project stabilizes and does not require major revision. Thus, users may assume more prominent stature in mature communities, partly due to their endorsement of a particular OSS project.

Communities are sensitive in the longer run to the free ridership phenomenon because attempts to control free ridership involve increased monitoring costs that eventually outweigh the rewards from contributions (Von Hippel and Von Krogh, 2003, p. 217). Given that free ridership is viewed positively in OSS development communities, it would be useful to study whether growth will simply strengthen the community or whether growth can ultimately erode community feelings. Will the trust that characterizes community governance (Adler, 2001) disappear as community size increases?

**C4-2: What are the limits to OSS user community growth?** Assuming that OSS use can be described by the typical s-curve for diffusion, at what point does the curve flatten out? Some OSS projects appear to have almost unlimited growth possibilities. For example, Linux began with one user and increased to 18 millions users.

## CONCLUSION

The framework offered in this paper is designed to stimulate a new direction in OSS research studies that focuses primarily on use rather than development. Although development has attracted the bulk of research interest to date, many important issues pertain to OSS use. OSS users far outnumber OSS developers, and as OSS products become more popular, the number of OSS users will continue to increase. We have identified many of the issues that make OSS different than the use of proprietary and in-house developed software and posed our research questions accordingly. Despite the desire to distinguish this new direction from prior research, our research agenda emphasizes the community perspective that has attracted such interest in OSS research so far. We believe that many valuable insights can be generated by a focus on OSS user communities.

## REFERENCES

Adler, P. (2001). "Market, Hierarchy, and Trust: The Knowledge Economy and the Future of Capitalism." *Organization Science* 12(2): 215-234.

- Buckner K. (1996). "Computer user groups: The advantage of successful partnership." *International Journal of Information Management*, 16(3), 195-204.
- Chengalur-Smith, S. and Sidorova, A. (2003). "Survival of Open-Source Projects: A Population Ecology Perspective," International Conference on Information Systems (ICIS), Seattle, Washington.
- Feller, J. and B. Fitzgerald (2000). "A framework analysis of the open source software development paradigm." *International Conference of Information Systems*, Brisbane, Australia.
- Fitzgerald, B. and T. Kenny (2003). "Open Source Software the Trenches: Lessons from a Large-Scale OSS Implementation." International Conference on Information Systems (ICIS), Seattle, Washington.
- Lakhani, K. R. and E. von Hippel (2003). "How open source software works: "Free" user-to-user assistance." *Research Policy* 32(6): 923.
- Markus, M. L., B. Manville, et al. (2000). "What makes a virtual organization work?" *Sloan Management Review*, 42: 13-26.
- Mockus, A., R. T. Fielding, et al. (2002). "Two Case Studies of Open Source Software Development: Apache and Mozilla." *ACM Transactions on Software Engineering and Methodology* 11(3): 309-346.
- Moen, R. (2000). "Eric Raymond's tips for effective open source advocacy." <http://www.itworld.com/AppDev/344/LWD000913expo00/pfindex.html>.
- Moen, R. (2003). "Linux User Group HOWTO." <http://www.linux.org/docs/ldp/howto/User-Group-HOWTO.html>.
- Raymond, E. (2001). *The Cathedral and the Bazaar, Musings on Linux and Open Source by an Accidental Revolutionary*. Sebastopol, CA, O'Reilly & Associates.
- Robey, D., K. Schwaig, and L. Jin (2003) "Intertwining Material and Virtual Work." *Information and Organization*, 13(2), 111-129.
- Shumar, W. and Renninger, A. K (2002) "On Conceptualizing Community", in *Building virtual communities: learning and change in cyberspace*, edited by Renninger and Shumar. Cambridge: Cambridge University Press.
- von Hippel, E. and von Krogh, G. (2003) "Open Source Software and the 'Private-Collective' Innovation Model: Issues for Organization Science," *Organization Science*, 14 (2), 209-223.
- Wellman, B. (2001). Physical place and cyberplace: The rise of personalized networking. *International Journal of Urban and Regional Research*, 25(2), 227-252.
- Ye, Y., K. Kishida, et al. (2002). Creating and Maintaining Sustainable Open Source Software Communities. *International Symposium on Future Software Technology*, Wuhan, China.



0 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/proceeding-paper/beyond-development-research-agenda-investigating/32682](http://www.igi-global.com/proceeding-paper/beyond-development-research-agenda-investigating/32682)

## Related Content

---

### On-Line Credit and Debit Card Processing and Fraud Prevention for E-Business

James G. Williams (2018). *Encyclopedia of Information Science and Technology, Fourth Edition* (pp. 2707-2722).

[www.irma-international.org/chapter/on-line-credit-and-debit-card-processing-and-fraud-prevention-for-e-business/183982](http://www.irma-international.org/chapter/on-line-credit-and-debit-card-processing-and-fraud-prevention-for-e-business/183982)

### A CSP-Based Approach for Managing the Dynamic Reconfiguration of Software Architecture

Abdelfetah Saadi, Youcef Hammal and Mourad Chabane Oussalah (2021). *International Journal of Information Technologies and Systems Approach* (pp. 156-173).

[www.irma-international.org/article/a-csp-based-approach-for-managing-the-dynamic-reconfiguration-of-software-architecture/272764](http://www.irma-international.org/article/a-csp-based-approach-for-managing-the-dynamic-reconfiguration-of-software-architecture/272764)

### Parallel and Distributed Pattern Mining

Ishak H.A Meddah and Nour El Houda REMIL (2019). *International Journal of Rough Sets and Data Analysis* (pp. 1-17).

[www.irma-international.org/article/parallel-and-distributed-pattern-mining/251898](http://www.irma-international.org/article/parallel-and-distributed-pattern-mining/251898)

### Temperature Measurement Method and Simulation of Power Cable Based on Edge Computing and RFID

Runmin Guan, Huan Chen, Jian Shang and Li Pan (2024). *International Journal of Information Technologies and Systems Approach* (pp. 1-20).

[www.irma-international.org/article/temperature-measurement-method-and-simulation-of-power-cable-based-on-edge-computing-and-rfid/341789](http://www.irma-international.org/article/temperature-measurement-method-and-simulation-of-power-cable-based-on-edge-computing-and-rfid/341789)

### Secure Mechanisms for Key Shares in Cloud Computing

Amar Buchade and Rajesh Ingle (2018). *International Journal of Rough Sets and Data Analysis* (pp. 21-41).

[www.irma-international.org/article/secure-mechanisms-for-key-shares-in-cloud-computing/206875](http://www.irma-international.org/article/secure-mechanisms-for-key-shares-in-cloud-computing/206875)