

Moving Open Source Communities: How Foundations Onboard Communities

Research-in-Progress

Georg J.P. Link
University of Nebraska at Omaha
glink@unomaha.edu

Abstract

Open source foundations onboard open source communities to provide professional community support especially as corporate interest increases. Extant literature explored how open source communities evolve and what influence open source foundations and corporate interests have. This study will shed light on the little-understood process that open source communities go through when joining an open source foundation. I will follow a phenomenological approach to learn how members of open source foundations understand the process of onboarding a new open source community. This extended abstract includes the justification for this research and suggests a viable method.

Keywords: Open Source Community, Open Source Foundation, Evolution

Introduction

Open source foundations onboard open source communities. For example, the Git project was initiated to support the development of the Linux Kernel but as an independent project joined the Software Freedom Conservancy (“Git and Software Freedom Conservancy” n.d.). This case is not unique. As of writing, the Software Freedom Conservancy lists 39 “member projects” on its website (Software Freedom Conservancy n.d.). Similarly, the Linux Foundation lists 47 “Projects”. Some of these open source communities have existed before the foundation was incorporated and were onboarded in recent years.

Open source foundations were established to serve as neutral brokers to the diverse community members, facilitate between their different interests, protect the respective community and its intellectual property, and provide administrative support (Germonprez, Allen, et al. 2013; Link and Germonprez 2016; Riehle 2010). Consequently, practices and norms in these open source communities have stabilized (Germonprez, Kendall, et al. 2013). It is important to understand how this stabilization is facilitated by foundations. Therefore, the purpose of this study is to understand the process by which open source communities join a foundation and stabilize structures, roles, and rules of the communities.

To my knowledge, no research exists on the evolution of open source community governance during the process of joining an open source foundation. Research in this area could benefit many open source communities that consider joining an open source foundation to better prepare for the process and master it with fewer road blocks. Open source foundations could benefit from understanding the process better to then provide better guidance to open source communities and potentially improve the process. In this study, I am interested in the perspective of the open source foundations’ employees because they help communities through the process, which leads to my research questions:

RQ1: How do open source foundation employees perceive the process through which their foundation onboards open source communities?

RQ2: How is open source governance affected when a foundation onboards an open source community?

RQ3: What influence, if any, does existing open source governance have on open source communities that are onboarded at a foundations?

The Emergence of Governance Structures

Traditional software development and open source software development have been compared to the cathedral and the bazaar. Raymond (2000) based this distinction on his personal experience and observation. The great cathedrals in Europe were built based on a plan and the masons and workers were assigned small parts in the process but their contributions remained anonymous. This compares to the traditional software development where an organization hires software developers but their individual contributions are never communicated. In contrast, the methods of open source communities are based on individual contributions and software developers are recognized for their contributions which is comparable to a bazaar. The direction of an open source project is not predetermined but depends on the contributions that developers choose to make.

The community ideal of fully equal members never existed in open source communities. Naturally, the initiator of an open source project would be able to make changes to the software and decline changes from other developers (Cornford et al. 2010; Jergensen et al. 2011). New developers have to prove themselves through reliable and quality contributions to earn trust before being allowed to make changes directly. However, the challenge of coordinating the development increases with more developers.

Developing software is no simple task as dependencies exist throughout the project. Dependencies exist inside the software between modules as well as in the process of creating software. Within small open source communities, community members can manage these dependencies informally by knowing and trusting each other, being aware of each other’s specific skills and strengths, and through frequent communication (Mockus et al. 2002).

Because of these challenges, governance structures, roles, and rules emerged in open source communities (O’Mahony and Ferraro 2007). These help organize and coordinate the efforts of the community in creating

software and elect a leader that represents the community when needed. The Debian project, for example, established a democratic system where all members could vote on a leader who has limited authority over technical matters but is charged with representing and coordinating the community (O'Mahony and Ferraro 2007). O'Mahony and Ferraro observed that developers were more likely to emerge when their technical contribution were widely used by other members and when they helped in the coordination of the community. This form of governance is referred to as meritocracy, where community members gain authority based on their technical and organizational contributions. Meritocracy is a common (if not default) governance model used by open source communities (Bacon 2012; Kelty 2008; O'Mahony and Ferraro 2007). The case of the Debian community showed that meritocracy is sometimes combined with democratic legitimacy as those who gain authority are officially elected for the role of the leader.

Differences in Governance Structures

In addition to meritocracy, other governance structures were found in open source communities. Germonprez et al. (2014) identified a total of four governance structures: meritocracy, bureaucracy, family/republic, and adhocracy. Adhocracy is a governance structure without structures where any member can make decisions and act on them. In an adhocracy, members often organize in specialized groups that work on different aspects of a shared goal. The family/republic governance works with elected leaders, like in a republic, who assume responsibility for the community just like a parent would in a family. Bureaucracy with its rigid rules and control structures was not found in the open source community under study, supposedly because it limited the community's freedom. The interesting finding is that the community under study used a mix of elements of meritocracy, adhocracy, and family/republic in its governance structure, which are all freedom oriented in nature (Germonprez et al. 2014).

These freedom oriented governance structures face challenges to maintain a healthy community when corporate interests are involved. This is critical since Riehle et al. (2014) found that corporate influence is paramount with about 50% of all open source code is developed by paid developers. Corporations engage in open source communities using a variety of strategies: by simply using the software, by actually providing feedback to the community, or ultimately by actively participating in the development process (Dahlander and Magnusson 2005). The latter strategy has direct impact on the community governance. Compared to volunteer developers, paid employees have been found to communicate more actively with a wider variety of community members (Dahlander and Wallin 2006). Additionally, corporate employees could utilize corporate resources that volunteer developers do not have access to and quickly become a central actor in the community. Corporations can strategically install an employee inside an open source foundation and gain influence over the direction of the community (Dahlander and Wallin 2006). When an open source community appears to be controlled by a corporation, volunteer developers might abandon the community (Spaeth et al. 2015) or fork the project and start a new community (Gamalielsson and Lundell 2014).

Tension not only exists between the open source community and corporations but also between corporations who engage in the same community to develop non-differentiating software that they can mutually benefit from (Germonprez, Allen, et al. 2013). Open source foundations were established as neutral facilitators to relieve the tensions. A "foundation serves as the steward of the [communities] under its responsibility. It provides financial backing, legal certainty, making the survival of the software less dependent on the individuals who initially started it" (Riehle 2010). Foundations are often non-profit organizations and as such have legal requirements for their management and governance. The foundation adds a managerial layer to the open source community that on the one hand solves the discussed tensions but on the other hand introduces a new layer of governance. Consequently, an open source community that joins an open source foundation evolves its governance structures, roles, and rules.

Extant research observes evolution of the community, the software, or the co-evolution of the two (Syeed et al. 2013). The community evolves because member join, leave, or change their roles within the community (Gutsche 2005; Nakakoji et al. 2002). Open source software typically grows in size and features as it matures (Godfrey and Tu 2000; Koch 2004), the structures of the software changes over time (Capiluppi et al. 2004). Changes in the community or software coincide with the other as they depend and impact each other (Nakakoji et al. 2002; Ye et al. 2005). The reported cases of evolution are either in communities that are independent of a foundation or belong to a foundation.

Institutional Theory

This study will use institutional theory as its theoretical lens. Institutional theory is found in many research disciplines, such as sociology, economy, politics, or organizational studies (Powell and DiMaggio 1991), but the latter is most applicable for this study, which “highlights cultural influences on decision making and formal structures” (Barley and Tolbert 1997, p. 93). Derived from the organizational studies, four streams of institutional theory research are found in the field of information systems: The first stream examines how institutions affect organizational decisions to adopt IT, the second stream examines the interaction between IT and institutions, the third stream examines the IT institutionalization process, and the fourth stream examines the discourse in the institutional processes (Nielsen et al. 2014).

Institutions in organizations are the structures, roles, and rules that constrain and shape human interaction and only become evident in the actions of individuals (Barley and Tolbert 1997). Therefore, the existence of an institution, which influences the interaction, cannot be assumed but must be evidenced in the interaction between individuals.

The establishment and change of institutions is Institutionalization. “Institutionalization [is] a process in which constraining relations with local constituents evolve over time” (Powell and DiMaggio 1991, p. 14). Institutionalization has five phases: innovation, theorization, diffusion, full institutionalization, and beginning of deinstitutionalization (Mignerat and Rivard 2009). This process is never complete, as institutions are subject to change over time (Luna-Reyes and Gil-Garcia 2013), especially during times of major organizational change (Miller and Friesen 1980) such as when open source communities join foundations. The changes in institutions can be observed over time.

Method

A suitable method for exploring the process through which open source foundations onboards communities, must consider the steps and challenges. Participant observation (van Maanen 1988) provides the means of learning from communities. Within participant observation, Netnography provides tools and a framework to conduct meaningful engagement with online communities (Kozinets 2015). Interviews with members of open source communities and foundations provide the main source of data which will be supplemented with archived mailing list conversations, documents provided by the community and foundation members, and personally created field notes. I will segment the process of onboarding into transition periods which will provide points in time between which changes in structures, roles, and rules are observable (Miller and Friesen 1980).

The context of my study is an open source community that originated from an academic research project. An open source foundation has agreed to onboard this community to ensure its sustainability, to attract more collaborators, and to help establish an open source standard. I have a unique opportunity to study the onboarding process as member of this community from a front-and-center perspective. This engagement provides access to community and foundation members and the chance to generate data required to answer the research questions (Dourish 2014).

References

- Bacon, J. 2012. *The art of community: building the new age of participation* (2 edition.), Sebastopol, CA: O'Reilly Media.
- Barley, S. R., and Tolbert, P. S. 1997. "Institutionalization and structuration: studying the links between action and institution," *Organization Studies* (18:1), pp. 93–117 (doi: 10.1177/017084069701800106).
- Capiluppi, A., Morisio, M., and Ramil, J. F. 2004. "Structural evolution of an open source system: a case study," in *Program Comprehension, 2004. Proceedings. 12th IEEE International Workshop on*, IEEE, pp. 172–182.
- Cornford, T., Shaikh, M., and Ciborra, C. 2010. "Hierarchy, Laboratory and Collective: Unveiling Linux as Innovation, Machination and Constitution," *Journal of the Association for Information Systems* (11:12).
- Dahlander, L., and Magnusson, M. G. 2005. "Relationships between open source software companies and communities: observations from Nordic firms," *Research Policy* (34:4), pp. 481–493 (doi: 10.1016/j.respol.2005.02.003).
- Dahlander, L., and Wallin, M. W. 2006. "A man on the inside: unlocking communities as complementary assets," *Research Policy* (35:8), pp. 1243–1259 (doi: 10.1016/j.respol.2006.09.011).
- Dourish, P. 2014. "Reading and Interpreting Ethnography," in *Ways of knowing in HCI* J. S. Olson and W. Kellogg (eds.), New York: Springer, pp. 1–24.
- Gamalielsson, J., and Lundell, B. 2014. "Sustainability of Open Source software communities beyond a fork: How and why has the LibreOffice project evolved?," *Journal of Systems and Software* (89), pp. 128–145 (doi: 10.1016/j.jss.2013.11.1077).
- Germonprez, M., Allen, J. P., Warner, B., Hill, J., and McClements, G. 2013. "Open Source Communities of Competitors," *ACM Interactions* (20:6), pp. 54–59 (doi: 10.1145/2527191).
- Germonprez, M., Kendall, J. E., Kendall, K. E., and Young, B. 2014. "Collectivism, creativity, competition, and control in open source software development: reflections on the emergent governance of the SPDXTtextregistered working group," *International Journal of Information Systems and Management* (1:1/2), pp. 125–145 (doi: 10.1504/ijisam.2014.062290).
- Germonprez, M., Kendall, K., Kendall, J., Young, B., Warner, B., and Mathiassen, L. 2013. "The Domestication of Open Source," in *DIGIT 2013 Proceedings*.
- "Git and Software Freedom Conservancy,," (n.d.). (available at <https://git-scm.com/sfc>; retrieved October 3, 2016).
- Godfrey, M. W., and Tu, Q. 2000. "Evolution in Open Source Software: A Case Study," in *International Conference on Software Maintenance, 2000. Proceedings.*, pp. 131–142 (doi: 10.1109/ICSM.2000.883030).
- Gutsche, J. 2005. "The Evolution of Open Source Communities,," *B.E. Journal of Economic Analysis & Policy: Topics in Economic Analysis & Policy* (5:1), pp. 1–18.
- Jergensen, C., Sarma, A., and Wagstrom, P. 2011. "The Onion Patch: Migration in Open Source Ecosystems," in *Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering ESEC/FSE '11*, Szeged, Hungary: ACM, pp. 70–80 (doi: 10.1145/2025113.2025127).
- Kelty, C. M. 2008. *Two Bits: The Cultural Significance of Free Software*, Durham: Duke University Press.
- Koch, S. 2004. "Profiling an Open Source Project Ecology and Its Programmers," *Electronic Markets* (14:2), pp. 77–88 (doi: 10.1080/10196780410001675031).
- Kozinets, R. V. 2015. *Netnography: Redefined*, SAGE Publications.
- Link, G., and Germonprez, M. 2016. "Understanding Open Source Communities as Complex Adaptive Systems: A Case of the R Project Community," *AMCIS 2016 Proceedings*.
- Luna-Reyes, L. F., and Gil-Garcia, J. R. 2013. "Understanding the Co-evolution of Institutions, Technology, and Organizations: The Enactment of the State Government Portal of Puebla," in *Proceedings of the 14th Annual International Conference on Digital Government Research dg.o '13*, New York, NY, USA: ACM, pp. 214–223 (doi: 10.1145/2479724.2479755).
- van Maanen, J. 1988. *Tales of the Field: On Writing Ethnography*, The University of Chicago Press.
- Mignerat, M., and Rivard, S. 2009. "Positioning the institutional perspective in information systems research," *Journal of Information Technology* (24:4), pp. 369–391 (doi: 10.1057/jit.2009.13).
- Miller, D., and Friesen, P. 1980. "Archetypes of Organizational Transition," *Administrative Science Quarterly* (25:2), pp. 268–299 (doi: 10.2307/2392455).

- Mockus, A., Fielding, R. T., and Herbsleb, J. D. 2002. "Two Case Studies of Open Source Software Development: Apache and Mozilla," *ACM Trans. Softw. Eng. Methodol.* (11:3), pp. 309–346 (doi: 10.1145/567793.567795).
- Nakakoji, K., Yamamoto, Y., Nishinaka, Y., Kishida, K., and Ye, Y. 2002. "Evolution Patterns of Open-source Software Systems and Communities," in *Proceedings of the International Workshop on Principles of Software Evolution IWPSE '02*, New York, NY, USA: ACM, pp. 76–85 (doi: 10.1145/512035.512055).
- Nielsen, J. A., Mathiassen, L., and Newell, S. 2014. "Theorization and Translation in Information Technology Institutionalization: Evidence from Danish Home Care," *MIS Quarterly* (38:1), pp. 165–A7.
- O'Mahony, S., and Ferraro, F. 2007. "The Emergence of Governance in an Open Source Community," *Academy of Management Journal* (50:5), pp. 1079–1106 (doi: 10.5465/AMJ.2007.27169153).
- Powell, W. W., and DiMaggio, P. 1991. *The New institutionalism in organizational analysis*, Chicago: University of Chicago Press.
- Raymond, E. S. 2000. "The Cathedral and the Bazaar," (available at <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/>).
- Riehle, D. 2010. "The Economic Case for Open Source Foundations," *Computer* (43:1), pp. 86–90 (doi: 10.1109/MC.2010.24).
- Riehle, D., Riemer, P., Kolassa, C., and Schmidt, M. 2014. "Paid vs. Volunteer Work in Open Source," in *System Sciences (HICSS), 2014 47th Hawaii International Conference on*, , January, pp. 3286–3295 (doi: 10.1109/HICSS.2014.407).
- Software Freedom Conservancy. (n.d.). "Current Member Projects," (available at <https://sfconservancy.org/members/current/>; retrieved October 3, 2016).
- Spaeth, S., Krogh, G. von, and He, F. 2015. "Research Note—Perceived Firm Attributes and Intrinsic Motivation in Sponsored Open Source Software Projects," *Information Systems Research* (26:1), pp. 224–237 (doi: 10.1287/isre.2014.0539).
- Syeed, M., Hammouda, I., and Syatä, T. 2013. "Evolution of Open Source Software Projects: A Systematic Literature Review," *Journal of Software* (8:11) (doi: 10.4304/jsw.8.11.2815-2829).
- Ye, Y., Nakakoji, K., Yamamoto, Y., and Kishida, K. 2005. "The co-evolution of systems and communities in free and open source software development," *Free/open source software development*, pp. 59–82.