Technology-In-Practice and its Influence on User Involvement in OSS 2.0 Projects

Michelle W. Purcell, Drexel University

Abstract
There is evidence of a lack of non-developer user involvement and usability problems with free and open source software. Open source software (OSS) 2.0 communities where developers are not necessarily the users of the software often have greater commercial and non-profit institutional involvement in the project. As such these types of projects have a greater need to engage non-developer users and are likely to employ more rigorous project software development practices. The study of OSS 2.0 communities provides an opportunity to expand on our understanding of the structures that support involving non-developer users in distributed participatory design. Utilizing the structurational perspective of technology as the main theoretical foundation, this poster presents initial results from a pilot study of one OSS 2.0 community discussing virtual mediation practices and their effect on users' involvement with regard to knowledge transfer and influence over design outcomes.

Keywords: participatory design, free and open source software, social informatics, organizations
Citation: Purcell, M. (2015). Technology-In-Practice and its Influence on User Involvement in OSS 2.0 Projects. In *iConference 2015 Proceedings*.

Research Data: In case you want to publish research data please contact the editor.
Contact: mjw23@drexel.edu

1 Introduction

The free and open source (FOSS) movement grew out of a desire to keep software innovation open (Deek & McHugh, 2008). By distributing code and permitting it to be modified it allowed anyone to build upon the contributions of others as a way to innovate the software. For instance, von Hippel (2001) presents the example of Apache Web Server where innovation occurred through webmasters as users modifying the code and Lin (2004) presents the example of Emacs where the openness of the word processor code enabled many different variants of the tool, e.g., CCAEmacs, XEmacs. In this model user involvement in design took the form of "scratching a developer's personal itch" (Raymond, 1999, p. 25). This means that users who were proficient in the development of code could influence the design of the software through contribution of code to the community. While many suggest that this standard practice represents users, there are studies that point to usability problems with FOSS and a lack of non-developer involvement (Iivari, 2009; Nichols & Twidale, n.d.; Wubishet, Bygstad, & Tsiavos, 2013).

While FOSS development continues to be conducted in large part virtually, there have been major changes since its inception which suggest the possibility of greater non-developer user involvement in certain types of projects. Fitzgerald (2006) makes a distinction between two broad classes of FOSS projects. One he terms as FOSS and the other as OSS 2.0. In FOSS and OSS 2.0 the software development process is different, reflecting differences in characteristics of the product domain, licensing, and business strategy. Such projects lie in product domains where developers are not the users of the software often having greater commercial and non-profit involvement in the project. As such these types of projects have a greater need to engage non-developer users and are likely to employ more rigorous project software development practices (Fitzgerald, 2006).

The study of OSS 2.0 communities provides an opportunity to expand on our understanding of the structures that support involving non-developer users in distributed participatory design. This knowledge has potential to help other open source software communities and distributed software development companies expand the role of non-developer users. This is important since it is believed that while systems development success depends on a host of factors participation among affected stakeholders is positively related to systems development success (Markus & Mao, 2004).

2 Theoretical Framework

Since open source software practices are mainly Internet based this study focuses on the virtual mediation practices of the projects. To that end, the foundation of the theoretical framework is a structurational perspective of technology (Orlikowski, 2000). Building upon the theory of structuration (Giddens, 1984), it acknowledges that while organizational structures may inscribe rules and resources within technical artifacts, human agency "enacts emergent structures" (p. 407) that may or may not reinforce existing organizational practice. This study therefore investigates how rules and resources with regard to the design of the software are constituted through interpretive schemes, norms and facilities enacted through technology-in-practice (Orlikowski, 2000) and how that influences non-developer users' ability to contribute to design outcomes.

The concept of framing will be used as analytical lens for investigating interpretive schemes of participants during the design process. Frames are the perspectives humans bring to the situation based on their own experiences that influence the information used and how it is processed (Goffman, 1974). These perceptions can vary widely across groups and tend to be associated with the interests of specific groups and subgroups in achieving specific outcomes – various frames influence the way technology is used and how it may mediate community interactions and decisions. In addition, different frames among actors may cause different interpretations on the nature of problem, models used for solving the problem, and determination of what is information is considered relevant (Davidson, 2002; Lanzara, 1983).

The design process will be examined from the perspective of distributed cognition among participants from different communities of practice (Hollan, Hutchins, & Kirsh, 2000; Lave & Wenger, 1991). The process is challenging due to the inextricable nature of knowledge and practice meaning that members from different communities of practice will have difficulties sharing knowledge given they have different practices (Brown & Duguid, 2001). "Distinct practices create distinct embedding circumstances. Therefore, to understand where knowledge flows and where it sticks we need to ask where and why practices (and so embedding circumstances) are common, and where and why they are not" (Brown and Duguid, 1991, p. 204). To enable common embedding circumstances requires mediation, technical or otherwise. Therefore, the theoretical framework also draws upon boundary object and boundary spanning theory (Carlile, 2002; Levina & Vaast, 2005; Star, 1989).

3 Research Design and Methods

The study is a grounded qualitative analysis of four OSS 2.0-type projects' enacted technology-in-practice and how it enables and constrains non-developer users' ability to contribute to design outcomes. It aims to answer:

1. How do facilities, norms and interpretive schemes enacted through technology-in-practice enable and constrain knowledge transfer among non-developer users and members from other communities of practice?
2. How do facilities, norms, and interpretive schemes enacted through technology-in-practice enable and constrain non-developer users' ability to influence design decisions within the community?

Data will include collection of online artifacts such as bug reports, mailing list communications, new feature requests and semi-structured interviews with participants. This grounded qualitative study will use discourse analysis as the method for conducting a dynamic analysis of participants' frames during the distributed cognition process. In addition, analysis will be conducted on the external artifacts such as boundary objects and their role in coordinating knowledge. Outcomes from the study will describe how characteristics of OSS 2.0 communities' enacted technology-in-practice may facilitate and constrain the process of distributed cognition and non-developer users' ability to contribute to design outcomes.

This poster will present initial results from a pilot study of one OSS 2.0 community.

References

Brown, J. S., & Duguid, P. (2001). Knowledge and organization: A social-practice perspective. *Organization Science*, *12*(2), 198–213.

Carlile, P. R. (2002). A pragmatic view of knowledge and boundaries: Boundary objects in new product development. *Organization Science*, *13*(4), 442–455.

Davidson, E. J. (2002). Technology frames and framing: A socio-cognitive investigation of requirements determination. *MIS Quarterly*, *26*(4), 329–258.

Deek, F. P., & McHugh, J. A. (2008). *Open Source*. Cambridge University Press, Cambridge.

Fitzgerald, B. (2006). The transformation of open source software. *MIS Quarterly*, *30*(3), 587–598.

Giddens, A. (1984). *The constitution of society: Outline of the theory of structuration*. Cambridge, UK: Polity Press.

Goffman, E. (1974). *Frame analysis: An essay on the organization of experience.* Harvard University Press.

Hollan, J., Hutchins, E., & Kirsh, D. (2000). Distributed cognition: toward a new foundation for human-computer interaction research. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *7*(2), 174–196.

Iivari, N. (2009). " Constructing the users" in open source software development: An interpretive case study of user participation. *Information Technology & People*, *22*(2), 132–156.

Lanzara, G. F. (1983). The design process: Frames, metaphors, and games. In U. Briefs, C. Ciborra, & L. Schneider (Eds.), *Systems Design For, With, and By the Users*. North-Holland Publishing Company.

Lave, J., & Wenger, E. (1991). *Situated learning : legitimate peripheral participation*. Cambridge [England]; New York: Cambridge University Press.

Levina, N., & Vaast, E. (2005). The emergence of boundary spanning competence in practice: Implications for implementation and use of information systems. *MIS Quarterly*, 335–363.

Lin, Y. (2004). Epistemologically multiple actor-centered systems: or, EMACS at work! *Ubiquity*, *2004*(February), 1–1.

Markus, M. L., & Mao, J.Y. (2004). Participation in development and implementation-updating an old, tired concept for today's IS contexts. *Journal of the Association for Information Systems*, *5*(11), 14.

Nichols, D., & Twidale, M. (n.d.). The usability of open source software | Nichols | First Monday. Retrieved July 7, 2014, from http://firstmonday.org/article/view/1018/939

Orlikowski, W. J. (2000). Using technology and constituting structures: A practice lens for studying technology in organizations. *Organization Science*, 404–428.

Raymond, E. (1999). The cathedral and the bazaar. *Knowledge, Technology & Policy*, *12*(3), 23–49.

Star, S. L. (1989). The structure of ill-structured solutions: heterogeneous problem-solving, boundary objects and distributed artificial intelligence. In M. Huhns (Ed.), *Distributed Artificial Intelligence* (Vol. 2, pp. 37–54). San Francisco, CA: Morgan Kaufmann Publishers, Inc.

Von Hippel, E. (2001). Learning from open-source software. *MIT Sloan Management Review*, *42*(4), 82–86.

Wubishet, Z. S., Bygstad, B., & Tsiavos, P. (2013). A participation paradox: Seeking the missing link between free/open source software and participatory Design. *Journal of Advances in Information Technology*, *4*(4), 181–193.