Issues with designer participation in FLOSS communities

Paula M. Bach, The Pennsylvania State University

Pascal Finette, Mozilla Labs

Abstract In this position paper we describe the following challenges to bringing designers to open source communities: motivation, socio-technical platform, incentives, quality, and partnerships. We argue that designers are important for continued innovation in FLOSS communities and that more research is needed to understand these problems further in order to begin offering solutions.

Introduction

Designers include people with skills such as user researchers, interaction designers, usability engineers, information architects, and other job functions that support user-centered design. User-centered design refers to various methods and practices in software development focusing on user needs. Particularly in communities that build domain specific software, or applications that are widely used by many user types, designer participation in free/libre/open source (FLOSS) communities is essential. The design roles in FLOSS range from paid professionals, to students, to volunteers. In this position paper we argue that in order for FLOSS communities to sustain innovation the user-centered design role must be supported. The innovation and evolution of good software includes well-designed software, and well-designed software can be used by many people.

Challenges for building a sustainable open source community that includes designers

Building a community of designers around an open source project has many challenges. These challenges include mainly grassroots initiatives to bring design innovation to help sustain open source communities. Challenges surrounding bringing designers to open source communities include motivation, socio-technical platform, incentive, quality, and partnerships.

Motivation

While motivation for open source developers has been studied previously, motivation of designer participation in FLOSS projects has not been studied, mainly because designer participation is relatively rare. Developers are motivated by both intrinsic and extrinsic drivers (Lakhani & Wolf, 2005). Intrinsic drivers include creativity, fun, and intellectual stimulation, and extrinsic motivators include career advancement and better jobs. We have seen that some open source projects, such as Mozilla and OpenOffice.org, pay user experience designers and thus are driven by extrinsic motivators. We conjecture that designers would be motivated by intrinsic factors much like open source developers. For example, design is fundamentally a creative activity and therefore exploring creativity motivates designers. So while designers may have more creative freedom than developers in their day jobs, they also are bound by organizational constraints that might not exist in open source environments and thus designers who are bound by such constraints could feel more creative in the absence of the constraints. On the other hand, socio-technical barriers may discourage even the most highly motivated designers.

Socio-technical platform

Even if we could gather the most highly motivated designers, socio-technical barriers can potentially discourage participation. The open source environments are largely based on merit and trust. Developers who write good code are valued and trusted. Many designers have little or no coding skills and in the open source environments developers have no way to assign merit to design skills. Furthermore, developer work is different from designer work and currently the technical infrastructure in open source communities does not support designer work very well. If design work occurs in open source projects, it does so over email lists or discussion forums. The best environment for designers to work is in a 'war room' where multidisciplinary teams consisting of multiple stakeholders convene in a room and work together identifying and solving design problems using white boards, sketches, and other face-to-face mechanisms. A successful distributed, asynchronous version of best design practices has not yet been discovered (Olson & Olson, 2000).

A socio-technical platform would incorporate the FLOSS principles of "use, study, modify, and copy" to design. This can be enormously empowering for the individual (e.g. a design student has access to the source files of a complex design solution (e.g. a layered PSD file or a vector drawing) and can use them freely in his or her own projects, iterate on them and mash them up with other works) as well as increase "social welfare". For example a large design library akin to something like wikipedia, *designpedia,* would motivate designers to use the library and participate in open design. This can prove to be a strong motivator with wide-reaching implications for amassing design as social good.

Incentives

Even if we could build a socio-technical environment where many designers from across the globe could work together on design problems, we need a way to convince them that submitting their ideas and expertise in an open environment is fundamentally a good idea. This idea of collecting people of similar interest and bringing their ideas into the community to spark innovation has been recently used as a business model in a company called Cambrian House and has been called *crowdsourcing* (Howe, 2006). The Cambrian House crowdsourcing business model provides different incentives for idea submissions such as points for community participation that can eventually be turned into money if the ideas turn to products. Venture capital funds are also available for project ideas that are deemed important to the community. While the Cambrian House business model provides traditional monetary incentives, we are thinking of other incentives such as design awards and overall popularity of a given design where winning designs would be displayed in a top design gallery.

Quality

Quality is an ethereal term that can be demonstrated in this context by asking the question, where do the best ideas come from? Creating a successful incentive schema that brings many ideas into the community has challenges. The community solves part of the quality question. The community is able to drop bad ideas out of the bottom and let the good ideas bubble to the top. The nature of the community impacts quality. For example, a highly engaged community with passionate users might not let the most innovative ideas bubble to the top for fear of change. We conjecture that the best ideas come from passionate, engaged designers and lead users (Von Hippel, 2005).

Partnerships

Open source software development is a developer's world. Partially because of this, schools with strong design programs and software design firms don't think to have their students participate in open source projects or even establish partnerships and create new business models. Some programs, for example, Season of Usability, partner design students with open source projects to help work on the software's user experience. While this brings new ideas in terms of improving the project's usability, it does little to inspire new innovations from designers. Other partnerships with open source projects that bring design thinking and processes to open source projects or to start new projects from new ideas have not yet gained momentum. As a start a socio-technical infrastructure for collaborative design is important for when partnerships are established because designers would be unfamiliar with the open source developer environment and practices and thus have a difficult time establishing design innovation in an environment that does not support design activities.

Conclusion

Building a sustainable open source community requires strategies for innovation and designer and user participation. Designers facilitate ideas from the user community and through the process of iteration can build ideas into working prototypes that can be tested and commented on by the community, and a community of open source developers can build good designs. Throughout the process, a community of designers can drive innovation and change that helps sustain the community by producing superior, innovative software.

References

Howe, J. (2006) The Rise of Crowdsourcing. *Wired, 14*, 06.

Lakhani, K. R., & 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. A. Hissam & K. R. Lakhani (Eds.), *Perspectives on Free and Open Source Software*. Cambridge: MIT Press.

Olson, G. M., & J. S. Olson. (2000). Distance matters. *Human Computer Interaction* 15 139–179.

Von Hippel, E. (2005). *Democratizing Innovation*. Cambridge: MIT Press.