

A Survey of Usability Practices in Free/Libre/Open Source Software

Celeste Lyn Paul

KDE

celeste@kde.org
<http://www.kde.org>

Abstract. A review of case studies about usability in eight Free/Libre/Open Source Software (FLOSS) projects showed that an important issue regarding a usability initiative in the project was the lack of user research. User research is a key component in the user-centered design (UCD) process and a necessary step for creating usable products. Reasons why FLOSS projects suffered from a lack of user research included poor or unclear project leadership, cultural differences between developer and designers, and a lack of usability engineers. By identifying these critical issues, the FLOSS usability community can begin addressing problems in the efficacy of usability activities and work towards creating more usable FLOSS products.

1 Introduction

The term open source in Free/Libre/Open Source Software (FLOSS) is a software licensing philosophy in which the human readable source code of a software is available for the public to freely install, modify, or redistribute [18]. FLOSS can also refer to the community and development practices of thousands of developers who subscribe to this philosophy and license their software under one of the many available software licenses. In this paper, “FLOSS” refers to FLOSS as the community and development practices of contributors who also subscribe to the “open source” philosophy.

Raymond [23] was one of the first to describe some of the idiosyncrasies of the social economics characterized in FLOSS development including the practice to “release early and release often”, and the idea that developers are “scratching an itch” by writing software for themselves. These have been cited as the most notable reasons why FLOSS is criticized for being developer-centric and why they have poor usability [12, 13, 15, 22].

The concept of usability has multiple definitions including the ISO 9241-11:1998 standard and Nielsen’s heuristics standard [16]. In this paper, the term usability is used as a general term to mean the aforementioned definitions as well as the engineering methods and practices involved in this quality.

Shortly after Raymond’s article, FLOSS usability projects formed – as a response to his article or as a natural awareness of the FLOSS community is unclear. There are several usability projects [4, 7, 9, 10, 19, 20] whose goals are to improve FLOSS

usability by educating developers about good usability practices and conducting usability studies on FLOSS. Also, many FLOSS-centered companies such as Canonical, Novell, Red Hat, and Sun Microsystems have hired designers and usability engineers to help improve the usability of their software.

FLOSS usability projects experience interesting challenges compared to their industry counterparts. Many FLOSS projects are written by unpaid volunteers, and volunteers are distributed around the world [8]. One of the greatest incentives for developers to participate in a FLOSS project is, as Raymond describes, to fulfill their own expectations. Fixing usability issues, especially when the project owner/developer does not experience them, are not always a priority.

Nearly 10 years since the founding of the first FLOSS usability projects, contributors are still struggling to get basic usability practices integrated in to development [5, 24]. Progress towards better FLOSS usability is being made, but it is very slow. This paper discusses a review of case studies of usability in eight FLOSS projects which aimed to discover if there were commonalities among the reported challenges or failures. A discussion follows of what were found to be the greatest issues in FLOSS usability projects, why these issues have such an impact on usability, and how to overcome these issues and begin improving FLOSS usability.

2 Method

There have been many usability reports published by FLOSS usability projects describing the testing and evaluation of FLOSS [4, 5, 9, 10]. Recently, the most notable scholarly discussion about FLOSS usability has been by Twidale and Nichols [26]. However, they primarily focus on defects in FLOSS and the workflow surrounding bug reporting. There have been few discussions on why these defects occur or how they can be better prevented.

Case studies are a pragmatic reporting format since they focus on practice, organization, and processes more than data disconnected from context. However, there are very few case studies describing successes and failures in usability of FLOSS projects. This may be due to the need to “report early and often” [24], which is discussed later in this paper. Though they are few, the available case studies are important because they help FLOSS usability engineers identify organizational and procedural problems which affect the effectiveness of a usability initiative.

The eight FLOSS projects reviewed were from case studies published in conference proceedings and journals where experienced FLOSS usability engineers wrote about their experiences in the FLOSS project. Although they were written by multiple authors and published in varying venues, all of the case studies followed a similar format which described the condition of the problem with conclusions and summary thoughts. These “lessons learned” issues were explicitly enumerated either through bullet points or section headers, providing the basis for data collection.

2.1 FLOSS Projects Reviewed

The FLOSS projects written about in the case studies have active development and user communities.

Table 1 provides a list of the FLOSS projects reviewed.

Table 1. Provides a list of the FLOSS projects reviewed

FLOSS Project	Description of Project
GIMP [24]	A powerful open source image editing and manipulation tool
GNOME [3]	A popular open source desktop environment known for its ease of use
KDE [24]	A popular open source desktop environment known for its customization and features
Mozilla [25]	A suite of web communication applications, most notably the Firefox web browser
NetBeans [3]	A full-featured Java integrated development environment (IDE)
OpenOffice [14]	A popular office productivity suite
TV Browser [24]	A cross-platform television programming guide
TYPO3 [17]	An enterprise-level web content management system

3 Results

Among the 8 FLOSS projects, reported issues fell into one of 8 identified topics:

1. Business Practices, which involve organizational strategy and decisions that have an affect on the business aspect of the project (if it exists)
2. Communication, including the content, context, and medium for discussing usability issues
3. Culture, these are idiosyncrasies of the practices and attitudes found in FLOSS contributors and communities
4. Leadership, the decision makers and authority figures of a project who oversee and promote project goals
5. Methods, the usability methods, activities, and processes practiced such as surveys, user profiling, and subscribing to the UCD method
6. Ownership, those who have domain over goals, user interface, and functionality
7. Usability Engineers, the recruitment and involvement of experiences usability and design practitioners
8. User Research, the understanding of user needs and goals and how they relate to the functional and technical requirements and business goals of the product

Several FLOSS projects were the only to report on a certain issue (e.g. Mozilla was the only case study to discuss business practices), or they reported a similar issue more than once (e.g. TV Browser cites two issues regarding User Research). Final analysis considered if an issue topic occurred one or more times in a project case study rather than the total number of occurrences.

The most common topic, and only topic reported by all eight FLOSS projects, was related to User Research. Methods was also a frequently cited topic (seven projects) and is closely related to the issues presented in the User Research topic. Communication (four projects) was also a highly cited issue topic.

Table 2 provides a summary of the most commonly occurring topics.

Table 2. Summary of the most commonly reported topics that affected usability in the surveyed FLOSS projects

Category	FLOSS Project			
	GIMP	GNOME	KDE	Mozilla
User Research	X	X	X	X
Methods	X	X	X	X
Communication		X		X
Culture		X	X	
Ownership				X
Leadership			X	X
Usability Engineers				X
Business Practices				X

Category	FLOSS Project			
	NetBeans	OpenOffice	TV Browser	TYPO3
User Research	X	X	X	X
Methods		X	X	X
Communication	X	X		
Culture				X
Ownership	X			
Leadership				
Usability Engineers		X		
Business Practices				

3.1 User Research

Every project reported that the lack of user research was an issue in achieving better usability. The GIMP, TV Browser, and TYPO3 case studies provided the most descriptive accounts of the effects of user research on their projects.

The GIMP project failed to identify clear project goals and a user focus early on. As a result, developers focused on user feature requests which resulted in irrelevant “feature creep”. This increased the complexity of the tool without providing an increased overall value. The project was forced to re-engineer the entire application and redefined the purpose and scope of the application.

TV Browser reported not knowing enough about their users in order to make informed design decisions. This problem was identified early in the project due to

their proactive inclusion of designers. A series of user surveys were conducted to help fill in gaps of information about the users.

TYPO3 experienced the effect of a lack of community interest and participation. The usability team was organized and attempted to gather user research data through stake holder surveys to help create user groups and personas. They received poor participation from the community and were unable to complete the activity and so design efforts continued without valuable user research. They speculated that reasons for a lack of participation from the community were that the culture of getting work done may have surpassed the need for understanding their users needs and goals.

The other projects also discuss the needs or effects of user research and usability. The GNOME, KDE and OpenOffice.org projects reported missing a shared vision of project goals or audience. The Mozilla case study stresses the importance of user

Table 3. Summary of reported User Research issues and their effect on the FLOSS project (if reported)

Project	Reported Issue	Effect on Project
GIMP	Failed to define target users and user goals	Suffered from “feature creep” as additional unnecessary functionality and complexity was added to the software
GNOME	No documentation of target users	None reported.
KDE	No defined and agreed upon vision and target users for the project due to lack of communication and leadership	Usability engineer had insufficient knowledge about the product and its users and had difficulty addressing usability issues
Mozilla	Encourages identifying target users because of a previous bad design experience	Wasted design resources by redesigning failed user interface elements
NetBeans	Concerned if mailing list users were their target users, and if not, who their target users were	None reported.
OpenOffice	No definition of target users	Concerned about the effect the missing data might have on design and development
TV Browser	Insufficient and incorrect knowledge about their target users and user expectations	Conducted user surveys to learn more about their users
TYPO3	Failed attempt (due to lack of community participation) to better understand their target users by conducting a survey to collect user information data	Could not create valuable user personas because of lack of research data

research. The project had failed to gather both functional and user requirements which resulted in wasted resources. NetBeans questions if mailing lists are representative of their true user population, and if not, who their primary users really are.

Table 3 provides a summary of reported User Research issues and the effect the issue had on the project.

3.2 Methods

Methods include what types of activities should be conducted, when, and how often. Seven of the eight FLOSS projects discuss methods in the case studies. This issue category is closely related to the User Research category, as many of the methods discussed were for the purpose of gathering data from users and learning more about them.

A variety of methods were discussed between the projects. GIMP and TV Browser suggested using surveys or task observations as a way to gather data for understanding their users. GNOME and Mozilla both stressed the importance of “design first, code later”. OpenOffice suggested that face-to-face or remote workshops could be effective for getting design ideas down on paper. TYPO3 attempted to develop heuristics to act as guidelines for developers working on user interfaces. KDE simply recommended using the most appropriate method to achieve an identified goal.

FLOSS is known for its highly iterative development cycle which complements the iterative design cycles of user-centered design (UCD) [13]. GNOME described how the GNOME Usability Project ought to act as a resource for developers to get early usability feedback; however, the authors admitted that developers never think to ask for design advice early in the development process. KDE reported punctual design feedback was the most valued because it allowed developers to address easy fixes immediately. Unfortunately this does not support larger usability activities such as usability testing and surveys, which require a greater time investment to prepare, conduct, and report. Mozilla described productivity loss due to time spent fielding bugs and fixing the UI after release, all of which could have been avoided if they had designed and refined it from the beginning.

Table 4 provides a summary and description of the methods reported by the FLOSS projects.

Table 4. Summary and description of Methods reported by FLOSS projects

Method Type	Description	FLOSS Projects
Early and Often <i>Design is involved early and is iterates throughout the development process</i>	<ul style="list-style-type: none"> • Design before coding • Find and fix design issues earlier 	GNOME KDE Mozilla
Participatory Methods <i>Methods which directly involve users</i>	<ul style="list-style-type: none"> • Surveys • User and task observations • User-based testing 	GIMP OpenOffice TV Browser
Non-participatory Methods <i>Methods which do not directly involve users</i>	<ul style="list-style-type: none"> • Competitive analysis • Design workshops • User interface guidelines • User interface heuristics 	GNOME OpenOffice TYPO3

3.3 Communication Channels

FLOSS development is often distributed and most communication occurs over mailing lists, forums, and Internet relay chat (IRC). Design over these mediums is difficult because of the iterative and visual nature of design and the textual nature of the communication protocols.

GNOME was dubious as to how useful mailing list and IRC discussions could be. Also, they thought that these communication methods, in addition to a complex bug database, may intimidate non-technical users (such as usability engineers). This can be especially true when developers begin discussing technical details and implementation. Mozilla experienced community and content problems with having a too-open (open-to-public) or too-closed (invite-only) mailing lists. NetBeans reported the problem of fragmented discussions between bug reports and mailing lists, which can also happen across different mailing lists.

OpenOffice made a recommendation of setting up a “collaborative, visual space” to help developers and designers more easily communicate visual and interactive ideas.

4 Discussion

User research is an important step for creating usable products and is essential to the UCD process [6]. Previous experience in FLOSS usability fueled the anticipation of user research to be a frequently reported issue in FLOSS usability, but it was still surprising that it was reported by each FLOSS project surveyed. It is significant that each FLOSS project experienced issues linked to user research so severe or valued the information provided by user research enough to indicate it in the case study.

These findings support the general consensus among FLOSS usability leaders that the lack of user research has had a significant impact on FLOSS usability [2, 4, 5, 13, 24]. The lack of user research affects a project’s ability to improve usability through appropriate methods. There seemed to be a strong connection between reported User Research issues and Methods issues, as many of the Methods issues were directly related to methods used for gathering user information for user research. Other issues described by the case studies also suggest a link between user research and other issues such as leadership, culture, and involvement of usability professionals.

4.1 Why Is User Research So Difficult?

In order to solve the user research problem, the reasons why user research is so difficult to accomplish in FLOSS must first be understood. This issue seems to be directly related to other reported issues found in the FLOSS usability case studies and has been discussed independently by other FLOSS usability project leaders.

4.1.1 Leadership

Defining project goals and target users are decisions that contributors to an entire FLOSS project must agree to. Without a shared vision, the project will not be able to move in a cohesive direction. Projects which have no clear leadership structure cannot make decisions since it does not ensure that all project members will follow the new vision [8]. A lack of leadership can make acquiring a decision more complicated.

Decisions must be made by relevant developers instead of the single project leader [24]. Projects who have strong leadership can define these goals, make executive decisions about features, and make usability and user research a priority [1].

4.1.2 Culture

FLOSS contributors are primarily developers, and it is easy for developers to assess the skill and competency of another developer through experience. It is difficult for usability engineers to prove themselves to developers in the same way since developers do not always understand the processes or results of usability activities [13]. As a result, there is often tension and communication problems when usability engineers and developers try to work together. Trust must be established between the usability engineer and developer in order to have a successful usability relationship [8, 13].

There is also a cultural issue of what it means to “do” and what activities are considered useful contributions. The TYPO3 project reported a failure in conducting a user research activity and speculated that the “culture of doing” was stronger than the “abstract notion” of understanding their users (which is a key component in creating usable products). Developers, who can count “doing” by lines of code or number of commits, simply did not see the value of abstract activities such as discussion and research.

4.1.3 Usability Engineers

It can be difficult to get started with user research and the ability to ask the right questions and draw useful conclusions requires experience. Defining user groups and personas takes guidance and experience. Usability engineers are necessary to help conduct user research studies such as surveys and interviews, and to help make sense of the data by summarizing it for developers in user groups and personas.

However, participation by usability engineers is rare [14, 24]. There are few incentives for usability engineers to get involved if they are aware of FLOSS at all [15]. Organizations such as OpenUsability were founded to help address these issues by acting as a meeting place where usability engineers and developers can meet on equal ground [13, 20].

5 Summary

The lack of user research was the most reported issue in the usability case studies of eight FLOSS projects reviewed. Several case study authors went in to detail of how not having a complete understanding of the project goals and users affected the usability of the FLOSS project they were working with. The identification of this problem is an important step for working towards improved FLOSS usability and more successful FLOSS usability projects.

There are several reasons why user research could be so difficult to accomplish in FLOSS projects. Leadership, cultural, and resource issues seem to have the greatest impact in whether user research is done or not. Clear and strong leadership is required to define a project vision and target users. The culture of the FLOSS project must support usability engineers as contributors and support their methods. Finally, more

usability engineers are needed to conduct user research studies and report the data in a way that can be understood by developers.

Addressing the three challenges of leadership, culture, and participation will make it easier to accomplish user research tasks in FLOSS projects. If these projects establish project goals and user focus, they will be able to more easily achieve good usability and improve the overall state of FLOSS usability.

6 Future Work

By understanding what is missing in FLOSS usability, we can begin to address these problems. A number of usability initiatives in FLOSS projects have actively been working towards including more user research in their projects.

A recent initiative by the KDE Usability Project are the KDE User Research Templates [11]. This template provides an outline to document user research data such as a project vision, user types, primary functionality, and use cases. The goal is to help developers document important user information and make it accessible to project members with minimal reliance on usability engineers. Contributors to the KDE Usability Project then review the developer-generated documents and provide feedback.

The OpenUsability Season of Usability [21] is an internship program that provides design students an opportunity to work with a FLOSS project. Project mentors encourage their students to conduct user research as a part of the project. These activities help students understand the goals and users, begin the documentation process, and begin institutionalizing the idea of user research in the project they are working with.

Feedback from these projects, as well as further investigation in to the usability methods used in FLOSS, can provide a better understanding beyond the surveyed case studies as to why user research is so difficult, and how it can be achieved.

Acknowledgements

Special thanks to Ellen Reitmayr of OpenUsability for her input to early discussions of the user research problem, as well as all of the usability engineers who contribute to FLOSS usability.

References

1. Asay, M.: Usability, a question of (open source) leadership. CNET (March 25, 2008), http://www.cnet.com/8301-13505_1-9903080-16.html
2. Benson, C.: Meeting the challenge of open source usability. BCS Interfaces Magazine 60, 9–12 (2004)
3. Benson, C., Müller-Prove, M., Mzourek, J.: Professional Usability in Open Source Projects: GNOME, OpenOffice.org, NetBeans. In: Proc. of ACM CHI 2004, pp. 1083–1084 (2004)
4. Better Desktop, <http://www.betterdesktop.org/> (last accessed September 2008)

5. Çetin, G., Gorturk, M.: Usability in Open Source: Community. *ACM Interactions Magazine* 14(6), 38–40 (2007)
6. Cooper, A.: *The Inmates are Running the Asylum*. Sams (1999)
7. FLOSS Usability, <http://www.flossusability.org/> (last accessed September 2008)
8. Frishberg, N., Dirks, A.M., Benson, C., Nickell, S., Smith, S.: Getting to Know You: Open Source Development Meets Usability. In: *Workshop in Proc. of ACM CHI 2002*, pp. 932–933 (2002)
9. GNOME Usability, <http://developer.gnome.org/projects/gup/> (last accessed September 2008)
10. KDE Usability Project, <http://usability.kde.org/> (last accessed September 2008)
11. KDE Project User Research Profile Template, http://techbase.kde.org/index.php?title=Projects/Usability/Project_User_Research_Template (last accessed September 2008)
12. Mühlig, J., Horstmann, J., Bruscherseifter, E., Ackermann, R.: *Linux Usability Study* (English Translation). Relevante AG technical report (2003)
13. Mühlig, J., Paul, C.: *OpenUsability.org: Usability and Open Source*. *BCS Interfaces Magazine* 66, 18–20 (2006)
14. Müller-Prove, M.: *User Experience for OpenOffice.org*. *ACM Interactions Magazine* 14(6), 47–48 (2007)
15. Nichols, D.M., Twidale, M.B.: *The Usability of Open Source Software*. *First Monday* 8(1) (2003), http://firstmonday.org/Issues/issue8_1/nichols/
16. Nielsen, J.: *Usability Engineering*. Morgan Kaufmann, San Francisco (1995)
17. Nielsen, L., Bodker, M.: To do or not to do: Usability in open source development. *BCS Interfaces Magazine* 71, 10–11 (2007)
18. Open Source Definition, <http://www.opensource.org/docs/osd> (last accessed September 2008)
19. Open Office User Experience Project, <http://ux.openoffice.org/> (last accessed September 2008)
20. Open Usability, <http://www.openusability.org/> (last accessed September 2008)
21. Open Usability Season of Usability, <http://season.openusability.org/> (last accessed September 2008)
22. Paul, C.L.: *Usability in Open Source*. *UPA Voice* 10(1) (2008), http://www.upassoc.org/upa_publications/upa_voice/volumes/2008/february/
23. Raymond, E.S.: *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, revised edn. O'Reilly & Associates, Inc., Sebastopol (2001); Raymond's article was first published in 1998 by *First Monday*
24. Reitmayr, E., Balazs, B., Mühlig, J.: Integrating Usability with Open Source Software Development: Case Studies from the Initiative OpenUsability. In: *Proc. of tOSSad 2006*, pp. 65–72 (2006)
25. Trudelle, P.: Ten Lessons Learned from Netscape's Flirtation with Open Source UI Development (Response to Open Source Development Meets Usability). In: *Workshop in the Proc. of ACM CHI 2002*, pp. 932–933 (2002)
26. Twidale, M.B., Nichols, D.M.: Exploring Usability Discussions in Open Source Development. In: *Proc. of HICSS 2005*, pp. 198–207 (2005)