Opening Up Open Source

Michelle Levesque and Jason Montojo
University of Toronto

As Free/Libre/Open Source Software (FLOSS) becomes an increasingly popular alternative to commercial software, its user base has extended from software developers to the general public. However, many FLOSS projects still suffer from usability issues such as nonintuitive interfaces and poor documentation. Many of these problems stem from the technical elite's general impatience towards new users. This negative attitude causes less effort to be spent on making it easier for new users to use the software. Thus, many of the non-technical aspects of the development cycle such as documentation and interface design are neglected. This neglect is further emphasized by the fact that many developers would rather work on the code than documentation and interface design. These social and usability issues must be resolved for FLOSS to become a much more viable alternative for both technical and non-technical users alike.

Free/Libre/Open Source Software (FLOSS) has become a competitive alternative to commercial software in almost all areas of computing. It has become more and more common to find FLOSS software in schools, hospitals, governments, businesses, and homes. The Linux operating system, the Firefox Web browser, and the Apache Web server are just some examples of FLOSS software that is being used on an increasing basis. There are many obvious advantages to FLOSS: the freedom to tinker with the code, the ability to observe exactly what the software is doing, and the lack of dependence on a commercial provider. As well, FLOSS has a strong international community of programmers who volunteer countless hours to produce this software.

With so many obvious advantages, the problems with FLOSS are often overlooked. Though the FLOSS movement claims to be open to everyone, many users and developers alike feel that the community is not as welcoming as it claims to be. This exclusion creates a gap between users and FLOSS developers, and this gap fosters software usability problems.

It is becoming increasingly common for corporations to participate in FLOSS projects; however, in this article we are focusing on issues that arise from projects produced primarily by self-governed volunteers. This is both because corporate software development is already well researched, and because it is the strong volunteer community that makes FLOSS so unique.

Open Community

One of FLOSS' greatest strengths is its openness: Anyone is free to contribute. However, more than just the capacity for contribution is necessary to create an open community. FLOSS communities are built upon geek and hacker cultures, and these cultures are not known for their friendliness towards new users or those with different opinions. This stubbornness often causes new users or non-hackers to feel unwelcome by the community. Though this exclusion is a social issue, the nature of FLOSS causes it to affect more than just social relationships. It also impacts the code.

Because FLOSS programmers tend to contribute code during their spare time, it is natural that they would make contributions that interest them. However, not all software design tasks are viewed as being equal. For example, there is much less geek prestige to be earned for interface design, user testing, or documentation. Therefore these tasks are often neglected by the volunteer coders who can receive more *geek cred*[1] by focusing on other elements of software design. An open source usability study states:

> Indeed, there may be a certain pride in the creation of a sophisticated product with a powerful, but challenging to learn interface. Mastery of such a product is difficult and so legitimates membership of an elite who can then distinguish itself from so-called *lusers*. [1]

In contrast, software companies hire employees to specifically perform tasks like user testing, documentation, and interface design. Usability experts, graphic artists, tech writers, and others all have a place in commercial software development. So where are these participants in FLOSS development? The hard geek culture behind FLOSS may be useful in creating powerful software, but it also drives away these other essential members of the software team [1].

When users criticize FLOSS' usability, their suggestions are often ignored or flamed, rather than analyzed and used to improve the software: "Geeks tend to treat others who disagree with loud, obvious disdain. These behaviors are harmful both to the disagreer and to the community as a whole" [2]. Many users are told that they have no right to complain about FLOSS software unless they are willing to fix it themselves [3]. This attitude chases off many users who might otherwise have become firm FLOSS supporters. In turn, it reduces the number of users who are available to report bugs, participate in user testing, and help out with basic documentation. Thus the gap between user and developer widens.

The distributed nature of FLOSS also means that not all developers have the same goals in mind. Some developers participate in order to create superior software. Others simply want to tinker with some code. Increasing the software's accessibility is not a priority for the second group. They are creating software for their own use, or simply coding for coding's sake, rather than creating software for a general audience. As one programmer on Slashdot <www.slashdot.org> said:

> You'd better believe I'm designing it for ME. It's not fun to design programs for other people. That's a job. I wouldn't do that for free. If you would like to PAY me to make it work for you,

I would be happy to. [4]

This attitude draws effort away from software usability and often causes users to believe that FLOSS is intended for hackers alone.

Usability

The engineer's fallacy is the belief that if something is easy for the designer, it will be easy for the average user, too. Usability does not occur naturally in software, it is something that must be consciously planned. But most software written outside the industry tends to be written by programmers for their own use, so they tend to focus on the technical aspect of their program such as the language it is written in or the algorithms being used. Some of them try to show off their mastery of the technology to other hackers, oftentimes at the expense of usability [1]. Although they "can be very good at designing interfaces for other hackers, they tend to be poor at modeling the thought processes of the other 95 percent of the population" [5].

When interfaces fail at being intuitive, users have two main options for getting help: developer-provided documentation, and community-provided documentation such as forums and mailing lists. The problem with developer-provided documentation is that it is sometimes non-existent, especially in FLOSS projects. When it is there, it is typically written with the assumption that the reader already has some technical background about the software and what it does. Information on forums and mailing lists is often just as technical as developer-provided documentation since the main code contributors are usually the ones fielding the questions.

FLOSS' community-provided documentation also assumes a certain level of technical skill. Unlike commercial software, which usually provides a hotline to satisfy this need, FLOSS projects typically use online communication channels like e-mail, newsgroups, chat rooms, online forums, and mailing lists [6]. Although these are invaluable resources, they are not easily accessible to the average user [7] who may not know how to use them or even that they exist.

Providing adequate documentation and easily accessible assistance is not an easy task, and many FLOSS developers would rather focus on their work than deal with these issues: "We all know that some projects, probably most, need better documentation and could use some more refactoring. But until you open up your wallet, get off our backs" [8]. In an environment where only hackers feel comfortable, yet do not want to do certain tasks, these tasks quickly become neglected.

To make matters worse, the distributed nature of FLOSS means that the different contributors may have different ideas about where they want their project to go. Such conflicts end up producing "15 different editors, several different Web browsers, several different desktops, and so on" often leaving the end-user with more choice, and just as much confusion [3]. For every choice that the user must make, the FLOSS

> ## "Though the FLOSS movement claims to be open to everyone, many users and developers alike feel that the community is not as welcoming as it claims to be."

---

learning curve becomes that much more difficult. Though there are many advantages to the variety of forks available in FLOSS, it just adds another layer of complexity for the users.

Conclusion

None of the problems that we describe above are that impossible to resolve.

> The existence of a problem does not necessarily mean that all OSS [open source software] interfaces are bad or that OSS is doomed to have hard-to-use interfaces, just a recognition that the interfaces ought to be and can be made better. [1]

Usability is an issue that has not been solved in proprietary software either. However, the FLOSS community has to actively acknowledge that this problem exists before an effective resolution can be implemented.

There are already some initiatives in place to try to solve some of these problems. An example is GrokDoc <www.grokdoc.net>, a usability study that strives to create documentation for GNU/Linux. Unlike most documentation, which is created by having developers explain how to perform various tasks, GrokDoc is based on having new users demonstrate exactly what they find difficult.

Efforts are also being developed to deal with the unwelcoming environment that many users and developers feel exists in FLOSS communities. The most pronounced of these efforts are the support mechanisms being built to try to encourage women to participate in FLOSS activities. FLOSSpols <www.flosspols.org> is a study currently in progress to try to understand the wide gender gap in FLOSS and to offer concrete recommendations on how to solve this gap. Other efforts include WOWEM, a gender equity and FLOSS research and education project, and LinuxChix, a community for supporting women in Linux. Despite these efforts, there is still a strong belief that most geek-saturated communities like Slashdot are often unwelcoming and hostile environments.

It is important to remember that volunteers do most FLOSS programming. It would be unreasonable to ask these volunteers to contribute in ways that they find boring, tedious, or work-like. However this does not mean that there will always be tasks that are neglected in FLOSS development.

We believe that if the FLOSS community makes the social adjustments necessary to create a more open setting, then non-hackers will become more inclined to participate. This includes graphic designers, teachers, writers, more developers, and just normal users. It is the inclusion of all of these groups in the development process that will make FLOSS stronger, more usable, and truly open to all.◆

References
1. Nichols, David, and Michael Twidale. "The Usability of Open Source Software." First Monday 8.1 (2003) <www.firstmonday.dk/issues/issue8_1/nichols>.
2. Lester, Andy. "Geek Culture Considered Harmful to Perl." Lightening Talks at Yet Another Perl Conference, St. Louis, MO, 20 June 2002 <www.petdance.com/perl/geek-culture>.
3. Gunton, Neil. "Open Source Myths." 25 July 2004 <www.neil

gunton.com/open_source_myths>.

4. Slashdot. "Five Fundamental Problems With Open Source." Comment By Bill Shooter of Bul. 13 Apr. 2004 <http://ask.slashdot.org/ask slashdot/04/04/12/1757244.shtml>.

5. Dibona, C., et al. Open Sources: Voices From the Open Source Revolution. 1st ed. O'Reilly, Jan. 1999 <www.oreilly.com/catalog/open sources/book/raymond2.html>.

6. Cubranic, Davor. "Open-Source Software Development." University of British Columbia, Nov. 1999 <http://sern.ucalgary.ca/~maurer/ICSE99WS/Submissions/Cubranic/Cubranic.html>.

7. Trudelle, Peter. "Bugzilla Bug 89907 - Need to make it easier for users to make us their default browser." Mozilla, 2 Jan. 2002 <http://bugzilla.mozilla.org/show_bug.cgi?id= 89907#c14>.

8. Countryman, Dan. "Sometimes great things have too [sic] be absorbed and thrown way [sic]." Object Country, 25 Apr. 2004 <http://jroller.com/page/object country/20040425>.

Note

1. Credibility among young fashionable urban individuals.

About the Authors

Michelle Levesque is currently involved with the Citizen Lab where she designs and implements programs to enumerate and circumvent state-imposed Internet content filtering. She is working towards a degree in software engineering at the University of Toronto.

E-mail: ml@cs.toronto.edu

Jason Montojo is currently developing bioinformatics Web application software for the Blueprint Initiative in Toronto, Canada. He also has worked on the Eclipse Open Source project as a member of the platform development team at Object Technology International. Montojo is working towards a degree in software engineering at the University of Toronto.

E-mail: j.montojo@utoronto.ca