

Originally published at:

http://www.upassoc.org/upa_publications/upa_voice/volumes/2008/february/usability-open-source.html

Available at:

http://web.archive.org/web/20080515210843/http://www.upassoc.org/upa_publications/upa_voice/volumes/2008/february/usability-open-source.html



Usability in Open Source Software

By Celeste Lyn Paul

Celeste Lyn Paul is Senior Interaction Architect for User-Centered Design, Inc., organizes the KDE Usability Project, and works with open source projects such as KDE, Linux Foundation, OpenUsability, and Ubuntu.

Free as in free speech.

Open source is a software licensing philosophy which believes the human readable code source of a software should be available for the public to freely install, modify, or redistribute. The term "open source" can also refer to the community and development practices of thousands of free/libre/open source software (FLOSS) projects who subscribe to this philosophy and license their software under one of the many available software licenses.

Even if you have never heard of "open source" before, there is a good chance you have used open source software. For example, the second most popular browser on the web, Firefox, is open source software. If you have ever flown with Virgin America, their in-flight entertainment system runs on the open source operating system Red Hat Linux. The popular web publishing systems WordPress and more recently MovableType are also open source.

Open source has been hot news in the design world as well. [Robert Watkins' 2003 article](#) in the UPA Voice first introduced open source to usability professionals. The Sugar user interface for the One Laptop Per Child project was a long running [discussion](#) on the Interaction Design Association's mailing list. Recently, open source was the featured topic in the [Summer 2007 issue of BCS's Interfaces magazine](#) and the [November/December 2007 issue of ACM's Interactions magazine](#).

The developers were running the asylum.

Open source is fundamentally a user-centered community. Projects rely on bug reports, wish lists, and functionality requests submitted by users to shape future development goals. Open source software is often criticized for having poor usability and being developer-centric, but it is important to remember that it began as a software movement for developers by developers. The

primary users were traditionally developers, making the software essentially user-centric.

Over time open source software grew in popularity, attracting a diverse population of users beyond developers. When new users who were less technically savvy began using developer-centered software, it became the tipping point at which open source software went from having an "advanced user interface" to a "bad user interface". However, software development continued to focus on functionality and features instead of improving the user interface.

A driving factor for pushing usability forward in open source was centered around the GNU Linux operating system. As Linux became more refined and ready for mainstream use, the concern for the quality of its software grew. Bad press about poor software usability correlating with a group of usability-dedicated developers and designers help pushed usability in a direction where a growing number of projects wanted it.

We've come a long way.

Being a designer in an open source project is not easy. For many years, usability specialists were disregarded, under appreciated, and ignored. This attitude probably sounds familiar to many of you because it was not too long ago that professional usability specialists encountered the same resistance.

Over time, developers in the open source community became more knowledgeable about usability and design due to good case studies, developer-designer education, and increased exposure through industry software practices. It took several years before many open source projects embraced usability as part of the development process and not as an afterthought.

There are a few of organizations dedicated to promoting and practicing usability in open source software. [**Open Usability**](#) is a not-for-profit organization which aims to bring designers and developers together to produce more usable software and get more designers involved in open source. One of its major outreach programs (in partnership with FLOSS Usability and Aspiration), the [**OpenUsability Season of Usability**](#), is modeled after Google's developer-oriented Summer of Code program. The program offers several sponsorships for design students to work with an experienced designer and a software developer to help improve the usability or design of an open source project.

Some open source software projects have managed to successfully integrate usability into their development cycle. Although a small project, KDE's HCI working group consists of artists, usability specialists, translators, and accessibility specialists to help improve the user experience of the K Desktop Environment. It is common practice for usability issues to be reported as bugs, which has elevated the status of these issues to as important as other bugs in the software. Designers often participate in developer meetings to discuss user interface design updates due to functionality or feature changes. The KDE project has also supported [**usability testing**](#) to help gather user feedback and improve its software.

The Linux Foundation's [**OpenPrinting working group**](#) aims to establish a set of standards so that printing across applications, desktop environments, and operating systems is consistent. One of the working group's projects includes developing a set of standards for consistent printing interfaces. They sponsor interaction designers and usability specialists so that they can attend meetings, conduct user research, design interface prototypes, and help write the specification.

Why get involved?

As usability professionals, we can learn a lot from working with open source software. The short fix-and-release cycles can help us develop practices we can apply to clients with an iterative- or short- development cycle (e.g. Agile development). The closer interaction with developers (often abstracted by project managers) can help us develop better documentation and deliverables for development teams. Many open source projects are highly innovative which provide unique challenges that can help us improve as designers.

Many of us enjoy participating in communities. Otherwise, we would not have the hundreds of UPA chapters, special interest groups, and mini-conferences which require dedicated volunteers. Communities offer the opportunity to sharpen and further develop our professional skills, network with colleagues and friends, or contribute to the greater good. The next time you are looking for a fun and innovative community to join, consider an open source project.