Fedora: A Case Study of Design in a FLOSS Community

Máirín Duffy
Red Hat, Inc.
314 Littleton Rd., Westford MA 01886
duffy@redhat.com
+1 978-392-3908

ABSTRACT
This paper presents a case study of the process of designing free/libre & open source software (FLOSS) concepts and interfaces within an open FLOSS community – the Fedora Project, a popular Linux-based operating system. This paper will review the challenges faced by designers in getting involved in FLOSS with examples from the Fedora community. It will also provide recommendations for getting started in an open source community as a designer, learned from experiences doing so in Fedora.

Author Keywords
FLOSS, open source, free software, development processes, community, software culture, case study

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION
The Free/Libre Open Source Software (FLOSS) community has traditionally been comprised of software developers scratching personal itches or teams of developers working towards a common goal, resulting in a developer-centric and highly-technical culture. A culture of design and attention to more user-centric practices in FLOSS development has not been very strong in this community [32] and designers are not contributing to these projects in great enough numbers. This is despite the benefits available to designers in FLOSS such as potential to reach an audience of millions and effect the usability of computing in a significant way (it is estimated that there are several million users of Fedora alone, for example [17]) and bridging the digital divide, having a publicly-available portfolio unencumbered by non-disclosure agreements, and an active and enthusiastic community of users willing to provide rich feedback on an ongoing basis.

Due to a lack of design emphasis and culture in many FLOSS projects, the less-than-ideal usability, accessibility, and utility of FLOSS [28] may hinder its adoption. FLOSS communities generally acknowledge that a greater emphasis on design is essential for wider adoption, and greater engagement by designers in FLOSS has been recommended in the past [32, 35,1]. Yet it is difficult to engage designers in FLOSS, a hindrance to FLOSS UX noted by Bach, DeLine, and Carroll [1]. How, then, can designers effectively engage FLOSS projects and reap the benefits of involvement in them?

There are myriad challenges that make it difficult for FLOSS projects to attract and retain designers and usability practitioners and build a user-centric culture within FLOSS. Challenges previously identified with respect to usability and the GNOME FLOSS project in particular in 2004 [3], for example, remain unaddressed. Two key long-term goals of a 2002 SIGCHI workshop on usability and FLOSS [20] still need work:

- Increasing attention to user-centered design in FLOSS projects
- Recruiting HCI students and professionals to FLOSS projects.

We will discuss the current challenges to achieving these goals and provide examples from the Fedora Project – a FLOSS operating system that aims to make FLOSS more consumable and easy-to-use. [16]

THE CHALLENGES

Distributed development
FLOSS projects typically involve contributors from around the world who speak different native languages and live in different time zones [20]. For example, the Fedora Project includes contributors from the United States, Canada, Venezuela, Italy, Brazil, Germany, Belgium, Romania, the Czech Republic, India, Japan, Australia, Mexico, Sweden, Ireland, France, Greece, and China, just to name a few – a membership map of just one team within the community is available [10]. The language used for most development is English, but some smaller sub-communities operate using a different language (for example, the Fedora-FR community which operates in French [18].)

This distributed development model poses obvious challenges for design work, as rich interaction and collaboration are difficult between designers, developers, and other contributors that are remote from each other [20]. Some of the regions in which Fedora contributors work in do not have high-bandwidth or reliable internet access, or simply have high costs associated with internet connectivity The quality of these connections limits possibilities for remote rich interaction via high-bandwidth technology – for example, streaming video and audio can be prohibitively difficult and/or costly. Perhaps in part because of these issues, current FLOSS communication mediums are primarily text-based.

Wide contributor dispersal across time zones limits the possibility of real-time communication methods – depending on the distribution of individual FLOSS project team members this may or may not become an issue. For the Fedora Design team, which has contributors in almost every continent (North America, South America, western and eastern Europe, and Asia) it has not been possible to schedule a live weekly status meeting that key team members can reasonably attend, so the group has adapted to operate primarily via non-live and primarily text-based communication mediums such as blogs and email-based mailing lists. This is a challenge as it can be difficult to explain design critique and brainstorm ideas in text; in industry these functions are often performed live in person or over the phone or VOIP using streaming screensharing/whiteboarding collaboration software. An example of a challenge the non-live nature of the medium poses is a design discussion (that might consume 10 minutes in real time) that results in a back-and-forth that spans several days over a mailing list.

Regional cultural differences also arise from time to time and pose additional challenges in communication and understanding. At times, this can be a strength, though – one example is when the Fedora Design Team mocked up a desktop wallpaper that, to a contributor, replicated the imperialistic Japanese flag of the 1940's. The contributor helped steer the team towards a more culturally-sensitive design.

Crowston, Li, Wei, Eseryel, & Howison provide more general discussion of the challenges posed by distributed development in FLOSS [8].

Lack of central organization

Unlike typical design projects sponsored by or designed within a corporation, FLOSS projects are typically self-organized [8], even those backed by corporate sponsors such as Fedora which is backed by Red Hat, Inc. FLOSS projects follow a “bazaar” organization, a term coined by Eric S. Raymond [29]. There is usually no or very little management or organizational charting – the Fedora project is a self-described meritocracy [13] in which those contributors who put the time and effort into furthering the project gain status and responsibility within a loose organizational structure.

Level of Professionalism

FLOSS design discussions take place openly in public, requiring skills beyond that of a typical corporate setting as noted by Benson [2]. As the vast majority of FLOSS contributors are volunteers [4], they are typically unpaid for their work and outside of social and legal expectations are free to express themselves in any manner they choose, including manners that would be unacceptable in a professional environment. The challenge for designers involved in FLOSS is to develop a tolerance for such behavior, especially since discussions incite much controversy from time to time in FLOSS, especially regarding highly-visible changes such as design. Contributors to FLOSS may be motivated by a desire to build skills which they have not yet developed (Ghosh found this was the most important reason to join FLOSS according to FLOSS participants [21]), meaning designers in FLOSS also need to be cognizant and tolerant of the skill levels which may vary quite a bit more than in a professional environment.

Stakeholder Identification and Power

The meritocracy model poses challenges to designers wishing to get involved in FLOSS projects: without the job titles and organizational charts of a traditional organization or design consultancy, it is especially difficult to identify the position and status of community members in order to determine the appropriate contacts necessary to create and follow a design through to successful completion. It takes patience and careful observation to fully understand the organization and hierarchy in the community**;** to be an effective designer may require significant time investment in the community. For example, a designer may complete a redesign of an existing UI, but in order to get that redesign implemented, the designer must first find the correct code maintainers, then work to gain their buy-in on the redesign work.

'Upstream' vs. 'Downstream'

Consider that the successful implementation of a complete user experience design may require design modifications to multiple FLOSS projects, some of which may not be part of the primary community a designer associates with. For example, within the Fedora community, there are parts of the desktop experience that are developed within the Fedora community, and there are parts that are inherited from the GNOME Desktop community [22]. The GNOME community is a separate FLOSS community referred to as an 'upstream' to Fedora since Fedora uses FLOSS designed and developed by the GNOME community. Fedora is considered a 'downstream' project with respect to GNOME as it is a consumer of FLOSS produced by the GNOME community.

The lines between 'upstream' and 'downstream' projects may be fluid and difficult to discern. For example, many

Fedora community contributors are also GNOME community contributors. The relationships may be complex too – Fedora is a downstream of GNOME, and in some cases GNOME is a downstream of the X.Org FLOSS community [37], which works on displaying graphics on the screen. Any given FLOSS project that serves as an upstream for another project may be an upstream for many downstream projects. In addition, there may be FLOSS projects that exist to unite 'sibling' FLOSS projects such as the GNOME and KDE desktop [25] projects, which are different desktop environments that work to share some systems to help ensure portability for users between desktops. The Freedesktop.org project serves to unite these two projects [19].

Complex relationships between projects can cause confusion for designers. This is especially true when designers approach FLOSS by identifying a problem they would like to help solve and find themselves unable to identify the correct community to approach in order to offer design help because of these complex inter-project relationships. Scacchi discusses some of the mechanics of this ecosystem [31].

Ownership and Licensing

The lack of a central organization also brings to light challenges regarding design ownership and intellectual property rights. One defining attribute of the Fedora Project is that code and content produced by and for the community is free now and forever, and that includes the freedom to redistribute [13, 16]. The requirement to allow free distribution of Fedora Project output limits the licenses under which contributed code and content may be accepted [6]. For designers who are accustomed to working for a single entity and who are rightfully very concerned about the license under which their work is distributed, a misunderstanding of FLOSS license requirements can mean a FLOSS project may not be legally able to make use of design work produced for it. The Fedora Design Team has been affected by such misunderstandings several times in the past, in some cases necessitating the re-creation of a design from the ground up in order to meet licensing requirements to complete a project – for example, a design submission that included inappropriately licensed content. A license must be stated – an ambiguous or missing license statement generally results in unusable design work.

Licenses that are acceptable typically include allowances for redistribution and modification, such as the Creative Commons Attribution and Creative Commons Attribution ShareAlike licenses [7]. The use of these licenses not only allows for design work to incorporated in FLOSS – it also ensures the creation of a body of design work that may be repurposed, studied, used as learning material, and built upon for all time. The open licensing of design works increases their potential reach and impact in FLOSS and proprietary software, correspondingly increasing the visibility of the designers involved.

Shared Vision

Finally, the lack of central organization inherent in a FLOSS project means there may not be a single driving vision for a project. Contributors in FLOSS projects, among whom the majority are volunteers, come to the project with different and sometimes conflicting visions and goals [26]. The absence of a central driving vision is challenging to a designer. It is necessary in design practice to balance the needs and requirements of different stakeholders, but it can be substantially more difficult to do so in a FLOSS project where the very goals of the entire project itself may not be agreed upon. An example of this is the Fedora Project website redesign, a project still in process and somewhat hindered by differing opinions across the community of contributors as to what Fedora itself is – an operating system, a platform for FLOSS development, a development community, among other definitions. Who is the target audience for the project? What kinds of users is the project wishing to attract? Decisions such as this within a corporate environment are usually made or commissioned by mid- to upper-level management, but within a FLOSS community such decisions are up for debate from individual contributors right up through project leadership. Depending on the situation, the designers themselves may need to define, propose, and act on a vision for a FLOSS community – a heavy responsibility, but a great opportunity to affect positive change.

The oft-conflicting and numerous voices and ideas in FLOSS communities results in an expansive and challenging amount of noise that designers must sift through in order to effectively determine the priorities of the project, the most crucial design tasks within those priorities, and the contexts which the actual design work needs to accommodate.

Cultural differences

A commitment to freedom and transparency is highly valued if not required in working within the FLOSS community. As Eric S. Raymond describes the development process of Linux, one of the most significant FLOSS projects to date, in *The Cathedral & the Bazaar* [29], “release early and often, delegate everything you can, be open to the point of promiscuity.” This culture of freedom and transparency is not limited to the processes under which successful design work in FLOSS must occur, but also encompasses the tools and formats used in artifact creation, the methods to share those artifacts, and the simple economics of working in FLOSS.

Designers encounter particular challenges in ramping up on FLOSS projects in addition to those faced by developer 'immigrants' (new developers ramping up on unfamiliar software projects) as identified by Sim and Holt [33].

Tools

In FLOSS culture, the most elegant technical solution is worthless if it is patent-encumbered; the most usable and useful piece of software is often eschewed entirely if the source is not available. In the latter case, if the source is available but under an inappropriate license, it may still be shunned within a FLOSS community. The utility of a tool in FLOSS culture, then, is not the only nor the primary attribute in determining its value – FLOSS tools are typically preferred in the creation of FLOSS itself. The Fedora Design team, for example, strongly encourages the use of FLOSS creative tools such as Gimp, Inkscape, and Scribus instead of proprietary tools from vendors such Adobe, Microsoft, Omnigroup, and Google, although in some cases the FLOSS tools are not as feature-rich as their proprietary counterparts. (In many cases they are perfectly sutitable.)

There are several pragmatic concerns that have lead to the adoption of a culture of using FLOSS software in FLOSS communities:

- **'Dogfooding'** – FLOSS communities typically practice 'dogfooding' which means you use your own product/project in your daily work. Pragmatically this means understanding in depth the strengths and limitations of the FLOSS your project is producing, and when you run into a bug or other problem with the software, it is more likely to be reported and fixed in a timely manner. To this end, many teams within the Fedora Project including the Design, Documentation Infrastructure, and various development teams use a 100% FLOSS toolchain and workflow which has led directly to patches and improvements to the tools on many occasions. In addition, dogfooding FLOSS tools that aren't necessarily part of a design workflow is of benefit to designers in gaining familiarity and domain knowledge of the FLOSS projects available. Building a base of familiarity of openly-licensed software effectively builds a designer's library of patterns they may freely draw from in their own design work. It also increases the designer's understanding of how various components interact and can lead to insights in user experience vision and design that span and integrate multiple FLOSS projects to meet a single common user need.

- **Economics** - Just as FLOSS community members are dispersed geographically, they also come from varying economic backgrounds. One of the stated reasons the Fedora Design Team prefers design work produced in FLOSS is that proprietary design tools are prohibitively expensive for many members of the design team itself (which includes high school students and international contributors.) Standardizing on the same tools reaps benefits in terms of documenting design processes, creating re-usable templates for design work, and using consistent file formats that any team member can view and edit.

- **Avoiding 'Bit rot'** - Using FLOSS file formats means that design work is less susceptible to 'bit rot' – the state at which a digital work is no longer viewable or modifiable because the tools used to create the work are not standardized and are no longer maintained or documented such that they can be easily reverse-engineered. FLOSS design tools tend to use file formats that are well-documented, open, unencumbered by patents, and standardized. For example, Inkscape is a FLOSS design tool that uses the SVG format, a vector graphics format standardized by the World Wide Web Consortium (W3C.)

- **Building a Showcase -** Using FLOSS tools to produce design work over time results in a portfolio that demonstrates what FLOSS tools are capable of. The existence of such a portfolio can be an effective tool in promoting the adoption of FLOSS tools, a chief aim of FLOSS projects including the Fedora Project.

- **Maintaining Appropriate Licensing -** FLOSS design tools typically include templates, patterns, brushes, fonts, palettes, icons, and other supplementary content useful in creating designs that has been reviewed and vetted from a licensing perspective as to be compatible with general FLOSS licensing requirements for freedom and redistribution. This means design work produced using this supplemental content will not inherit the baggage of incompatible licensing as it may with proprietary supplementary content.

- **Democratization of Production -** The free and redistributable nature of FLOSS design tools democratizes the production of designs. Typically in corporate environments, usage licenses for proprietary design software are purchased only for those who are employed as designers – other team members such as developers who may interface with designers may not have the benefit of opening up a design file and making modifications in order to illustrate an idea in communication with designers. This results in a relationship between designer and developer in which the designer has more control over design artifacts. This is not so in FLOSS communities, where a developer can easily download and install the same design tools. That a developer can modify design artifacts themselves can lead to a richer designer-developer interaction.

For designers who already have a preferred set of proprietary design tools and formats to use for their design work, the FLOSS culture and values may conflict and create a challenging environment in which to work in. This is especially true in the creation of interactive media involving technologies such as Adobe Flash(TM), Adobe Air(TM), and Microsoft Silverlight(TM) as there are no existing FLOSS tools capable of producing such media effectively. (As an aside, rapid progress in FLOSS does make it likely this situation can improve if more designers participating in the community take a stand and point out the need, driving the priority for such projects higher.) A key challenge for designers is to develop sensitivity to these cultural issues and to work out compromises with fellow FLOSS community members to avoid opposition to a design that, aside from the tools and formats used, would otherwise greatly benefit FLOSS. McLuhan's 'the medium is the message' absolutely applies here – the medium in which a design is presented will affect its adoption and general perception in FLOSS communities.

Effective participation in a FLOSS community may require a designer to learn a new set of tools – that are FLOSS – as well as administrative tasks such as account creation that may pose a significant barrier in terms of the time and resources required to learn and frustration involved. That the participants in this on-boarding process to date have overwhelmingly code-centric and technical, holes and potential improvements in the process that might particularly affect designers remain unaddressed. (One potentially impactful opportunity is a redefinition of the FLOSS on-boarding experience.) Sometimes FLOSS tools require using a FLOSS operating system (although many FLOSS design tools are cross-platform), the adoption of which can be very disorientating and pose a steep learning curve for a designer accustomed to proprietary operating systems. The frustration involved in the configuration and environment setup required to participate as a new contributor to a software project on the part of technical developers documented by Sim and Holt [32] may be a factor in designers ramping up on FLOSS projects, although projects such as live bootable Linux environments that do not require installation to a computer's hard drive may ease this somewhat.

It is not only the tools used in production of FLOSS design that can pose challenges for designers new to FLOSS – the tools used to communicate, share, and archive work may also be similarly foreign and are often more technical than their proprietary counterparts. What follows is a sampling of such communication and sharing tools:

- **Real-time communication –** real-time communication typically occurs in text-based chat using the free and open IRC and Jabber communication protocols. A number of different FLOSS exists to enable communication across these protocols, including cross-platform tools. There are subtle and unique differences in etiquette and syntax in each protocol which designers must learn to effectively communicate [24].
- **Documentation and content sharing -** Documentation and sharing of process and design typically occurs in FLOSS-based wikis, websites, and content management systems, each of which has their own syntax and processes both inherent to the software itself and the best practices of the community adopting it that must be learned for effective usage. For example, the Fedora Project uses a FLOSS application called MediaWiki and the Fedora Community has defined a set of conventions for documents on the wiki that include naming, structure, and media upload guidelines.
- **Non-real-time communication -** Non-real-time communication in many FLOSS communities involves email-based FLOSS mailing lists and blogs, which each have a rich and nuanced culture and set of etiquette all their own [30].

The reliance on text-based communication in FLOSS communities eases the communication barriers between contributors from different cultures that speak different native languages, as non-real-time text is easier for non-native language speakers to pour over and translate. In addition, technology that provides rough translations of text exists and even real-time 'bots' exist that sit in real-time FLOSS communication channels to provide rough translations of discussions between languages [34]. Oral communication is more difficult to gain fluency in compared to text-based communication; this is evident from the differing experiences FLOSS contributors have interacting online vs. in-person at FLOSS conferences. Not only is oral communication a challenge between native and non-native English speakers, but it also is a challenge between non-native English speakers who speak different native languages. For example, there have been cases at FLOSS conferences where non-native speakers were requested to provide subtitles for the recordings of their presentations as non-native speakers of other languages had a difficult time interpreting the speech of the speaker.

Process

The FLOSS culture's commitment to freedom and transparency is not limited to the choice of tools used in design. The transparent and open processes by which FLOSS design should occur according to Trudelle [35] may also pose a cultural challenge to designers wishing to contribute.

When a professional usability practitioner conducts a usability study for a corporate client, typically the practitioner is expected to sift through the raw data, analyzing it and drawing conclusions which are relayed to the client in various manners including usability reports and highlight reel videos of usability tests. If raw data is shared with clients, it is typically not shared publicly but rather within the privacy of the organization's internal network.

FLOSS communities are more participatory than this. After drawing conclusions from raw usability data in the Fedora community, for example, I have been asked to present the raw data from which the conclusions were drawn in order to back my conclusions as well as allow for other community members to draw and share their own conclusions. Not only this, but it is expected that such data is provided in open formats (for example, Ogg Theora format for videos, Ogg Vorbis format for audio, Open Document Formats for spreadsheets and reports) in public forums (FLOSS project wikis, websites, and email-based mailing lists to name a few) – all under an open license that allows for the redistribution of the usability data and content produced.

Such an open and transparent process can be of great benefit to the quality of usability research output, as more eyes analyzing the usability videos and scrutinizing the conclusions drawn can call into question inaccuracies and other mistakes in the research that can result in its improvement (in the same manner that more eyes on the code of FLOSS can result in less bugs and more rapid improvement than less-sifted-through code [29].) However, this example also brings to the surface a number of challenges posed by the transparent and open culture of FLOSS to usability practitioners:

- Participants in usability tests meant to benefit FLOSS must sign release forms that allow for the public posting and open license of raw usability data. This is particularly tricky in usability tests that necessitate the involvement of youth, disabled persons, and other groups which may be particularly sensitive to the public distribution of personally-identifiable information such as photographic likeness and demographic details.

- Sharing video data online can be quite costly and consume considerable bandwidth – for potentially small payoff as it is not entirely likely FLOSS community members will have the patience or skill to sit through usability test videos and provide useful analysis.

- Most audiovisual equipment does not encode in formats unencumbered by patents (such as Ogg Theora and Ogg Vorbis) necessitating the transcoding of audiovisual content to these formats which is costly both in terms of time and processing power.

- Practitioners simply may not be accustomed to such open access and posting of content and may need to periodically check themselves in order to stay mindful of whether or not they are working in a silo-ed manner (via private email and internal networks, for example.) It takes time and experience to learn when in a process is it effective and necessary to post work publicly and in which forum posting it is most appropriate. When writing an email to share an idea with a developer, for example, is it appropriate to also carbon-copy the relevant project mailing list as well? Or is it more effective to keep the conversation private and deal with potential community backlash in case some important decision is made behind the 'closed doors' of private email?

Outside of the specific example of usability testing, user experience design work and other contributions for FLOSS projects in general are expected to provide 'source.' In the same manner in which the sharing of source code is necessary for FLOSS developers to collaborate on the implementation of FLOSS software, it is important that designers submitting mockups, icons, and other design artifacts to a FLOSS project make available the source file that generated the design. While flat or bitmap format design artifacts that are not easily modified are still useful for communicating design to developers in a project, submitting source files for those designs is even more beneficial. If a designer ends up disappearing from a project due to time commitments elsewhere, for example, having the source from which their mockups was generated enables another designer to efficiently pick up where the original designer left off without having to redesign the basic framework of the mockups from scratch. Usability experts are better able to learn from previously-completed projects if the raw data and analysis was made publicly available.

One benefit of conducting design and usability work in such an open manner and providing the raw data publicly is that it makes it possible for FLOSS community members or even HCI students to observe and follow the process and potentially pick up some of the involved user experience skills. This may lead to an expanded pool from which FLOSS projects can draw user experience practitioners to help improve FLOSS. It may also present an opportunity for HCI instructors to provide more real-life examples of design practice than may be available elsewhere in their instruction. This follows the general principal of “welcome and allow different levels of participation” as a method of growing expertise, as describe by Wenger et. al. [36]

Lack of an established design community

Many FLOSS communities as well as the HCI community itself have identified the lack of an established design community within FLOSS. This is a difficult conundrum, indeed: to create an environment that encourages designer participation, it helps to have designers actively engaged in FLOSS in order to create that environment, but if you do not have enough designers engaged to create a shift in environment, how can you attract additional designers?

Green, Tollinger, Ratterman, Pyrzak, Eiser, Castro, and Vera posit that the developer-centric culture of FLOSS has resulted in more value placed on code in comparison to design [23].

The FLOSS community needs designers to teach proper user-centric development processes and help reform the currently-broken ones. The FLOSS community is a highly-technical and developer-centric community. Developers far outnumber designers and usability practitioners. This means the language and even the approach FLOSS projects take to solving problems tend to be focused on implementation and technology rather than starting with a real-life user problem to solve and determining appropriate implementation afterwards. This can result in FLOSS developers being unsure of how best to utilize a designer's time, which means their calls for help may not attract designers skilled in richer design practices. For example, it is far more common for a designer in FLOSS to be expected to apply surface-level enhancements such as icons and artwork to an application that is designed from a fundamentally-flawed base. FLOSS developers may be unfamiliar with design processes and may not know that design input is far more valuable at the beginning of a development process than as a 'coat of polish' on the end [3]. The self-assignment model of FLOSS development described by Crowston, Li, Wei, Eseryel, & Howison [8] further exacerbates the problem – the developers often cannot help designers define appropriate tasks and the designers are unable themselves to self-assign.

This situation requires that designers involved in FLOSS must be patient and willing to mentor developers in how the design process works so the developers understand how best to make use of the designers' time and how to approach and work with them.

That the FLOSS community is developer-centric also means that a great deal of focus has been put towards developing FLOSS tools for development. Since there have traditionally not been many designers in the community, there has been less focus and effort put forth to build and improve upon FLOSS tools necessary for design. This means the FLOSS tools available for design are not as rich and are not of the same level of quality as FLOSS development tools. Green, et. al. confirm this assessment [23].

The dominance of developers in the FLOSS community has resulted in methods of communicating and working that often assume a working technical knowledge of computer programming that in many cases exceeds a typical designer's background. For example, in many projects within the Fedora community, designers have been expected to check code out from a version control system**,** manipulate system-level configuration files, and manually build and run the code, all simply to be able to view and evaluate the interface of the software. This overly-technical environment can be overwhelming and discouraging to designers who try to establish authority and rapport with fellow community members. It can be struggle for a designer to do so when so much of his or her knowledge and skill is called into question on the basis of his or her perceived lack of technical prowess, which in non-FLOSS projects they might never be expected to possess.

The minority status of designers within the FLOSS community can at times result in designers (and other contributors, including translators and technical writers) to feel inferior and under-valued.

At least within the Fedora Project, however, a high value is placed on professional design work, and consultation with the Fedora Design Team is constantly sought. The demands of the community on the relatively small number of designers present within it is enough to be powerfully overwhelming, providing yet another source of discouragement and stress on the designers themselves. Overloaded with design tasks, the few designers present within the community are challenged when trying to make time to help on-board new designers approaching the community to help build capacity, as well as to make time for outreach programs to other non-FLOSS designers and HCI students.

As a suggestion for further study, examining closely those FLOSS projects that do have active designer contributions, assessing the health of each, and attempting to qualitatively determine the factors that may be at work may help identify ways to cultivate a healthy design community within in FLOSS projects.

CREATING A PLACE FOR DESIGNERS IN FLOSS

What follows are recommendations on how to address the challenges we've just reviewed for designers interested in FLOSS. These recommendations are based on experience in the Fedora Community for creating a place for design within a FLOSS community. Fedora's design community, as other FLOSS design communities, is still nascent but has shown growth over the past year, likely due in part to some of the following efforts:

Be visible in the community

It can be overwhelming for a designer new to FLOSS to join a FLOSS community – as mentioned earlier, developers may not know the best tasks towards which to direct a designer's attention, and the minority status of designers in the community and sheer amount of work needed can be insurmountably overwhelming. Making designers more visible within FLOSS communities, as recommended by Schwartz and Gunn [32], can help alleviate these issues. Additionally, Bird, Gourley, Devanbu, Swaminathan, and Tsu found support for the notion that social reputation impacts the chances a given developer will be accepted in a FLOSS project [4].

Create a place for design

Creating a sub-community, such as the Fedora Design Team within the Fedora Project, is a good way to provide a 'safe haven' for designers within the project to find each other, share their experiences, and mentor each other in FLOSS culture and tools.

Such a team can help build a culture of design within a FLOSS community simply by growing and interacting with other community members. Visibility is important because it will make community members consider design more often and more deeply.

Furthermore, having a central place other community contributors can go to in order to request help relieves the pressure individual designers may feel when queried for help directly. Tasks directed towards the team rather than particular individuals can be assigned amongst members of the team on the basis of available team-wide capacity, helping to relieve individual designers from an unreasonable burden.

Having a central place where design requests are sent and tracked is also of benefit to new designers coming on board. On the Fedora Design Team, we use a FLOSS application called 'Trac' to manage design requests in the form of 'tickets.' Each ticket is triaged, or analyzed in terms of priority and domain, by design team members and labeled appropriately: icon art requests are labeled 'icon art,' usability testing requests are labeled 'usability,' so on and so forth. Tickets are claimed by individual designers, and tickets that designers are not actively working on are marked as open. This results in a neat set of categorized tasks that new designers can peruse in order to find a project in need of their help, rather than requiring those new designers to ask around and dig out what tasks need to be done in a more time-consuming manner.

Hold real-time design 'hackfests'

Hackfest are a common type of event in FLOSS communities. While much FLOSS development occurs on non-real-time communication methods such as mailing lists, blogs, and version control commit messages, contributors regularly check in with each other in real-time in order to discuss and make higher-level decisions. Many times, this takes the form of in-person or online 'hackfests,' which are simply a gathering of FLOSS contributors at a pre-determined time with a pre-determined set of tasks or problems to discuss and solve.

Fedora Design Team's recent 'hackfest' (an interaction-design hackfest focused on Fedora stakeholder interviews for the website redesign on 24 Nov 2009 [9]) culminated in some very positive results – the team gained two new contributors and resulted in the completion of 3 full stakeholder interviews and the assignment of 3 additional ones to volunteers. Furthermore, it increased visibility of the design team and its process across the project: the event was announced on a project-wide blog, its results were summarized on discussed on that same project-wide blog [10], and curious Fedora contributors from other teams within the project attended the event in order to observe and learn more about design.

Schwartz and Gunn [32] report success in involving designers in FLOSS projects using similar events called 'Usability Sprints.'

Attend in-person conferences

Many FLOSS projects have regular in-person conferences for contributors to meet, discuss, and work on FLOSS in-person: Fedora has the bi-annual Fedora Users' and Developers' Conference (FUDcon), the GNOME project has both the annual GNOME Users And Developers Everywhere Conference (GUADEC), and the annual GNOME Boston Summit, there are also more general FLOSS community conferences such as the annual Python Conference (PyCon), and the annual Ottawa Linux Symposium (OLS), just to name a small few.

Designers should consider attending and speaking at such conferences. Not only do these events provide the rare opportunity for rich interaction with other contributors in solving design problems, but they also presents an opportunity to:

- Increase awareness of user experience design as a needed discipline with FLOSS.
- Increase awareness of the problems discussed in this paper than the FLOSS community must help solve in order to attract and retain more designers.
- Reach many contributors at once in demonstrating best practices for software design and interacting effectively with designers.

Attending a FLOSS conference and relating to fellow FLOSS contributors on a human level could also help dispel feelings of intimidation and inferiority on the part of FLOSS designers – and it can help build solid rapport and respect between designers and developers which is invaluable during design discussions in times of conflict.

Blog and post frequently within community forums

A great way to build visibility, appreciation, and greater understanding of design practices within a FLOSS community is to post regularly and openly about design within that community.

Many FLOSS communities, including the Fedora Project, have what are referred to as 'blog planets,' which are simply an aggregation of all of the blogs written by community members. Many community members read these planets on a regular if not daily basis, and they serve as one of the primary cross-project communication mechanisms.

By following and posting frequently about design to these blog planets as well as to project mailing lists, a designer can make a name for themselves within the community, building trust and rapport with other contributors, as well as bring greater awareness of FLOSS design to the project – the importance of socializing in such venues in FLOSS development is noted by Scacchi [31].

Designers on 'Planet Fedora' blog about how the design process works itself as well as present design work in-progress for feedback from the community. Presenting design work in particular is a good way to show the value of design – fore example, before and after screenshots could attract a great deal of positive commentary from blog readers and help build support for attracting more designers to a project.

Adapt to the culture of sharing

For successful participation in a FLOSS project, designers should consider ways in which they may be willing to be more open in terms of the licensing under which their design work is offered to the project as well as the transparency in which their design work is created and shared.

License Considerations

The Fedora Project has addressed the issue of contribution licensing in particular with a document called the 'Contributor License Agreement' (CLA) [14]. Any contributor joining the Fedora Project must sign and agree to the CLA in order to be granted an account and submit work to the project. The CLA grants the Fedora Project and its primary corporate sponsor Red Hat, Inc. a royalty-free license to contributions submitted to the project. Some FLOSS projects have similar agreements, some do not. In the case a designer decides to work for a FLOSS project that does not clearly define the license under which contributions must be submitted, such as the Fedora CLA, the designer should consider clearly defining this with the project up-front to ensure both their design work is consumable by the project it is meant to help as well as protect the designer's rights. At that point, the Creative Commons Attribution and/or Share Alike licenses may be useful to convey the necessary rights [7].

Process

In terms of opening the design process, openness to a new way of working and sheer discipline on the part of the designer is recommended:

- Always make sure that source is available for every design artifact provided, preferably source that is able to be opened and modified in FLOSS (note that some proprietary formats such as Adobe's PSD format are within some limits able to be manipulated in FLOSS programs such as GIMP.)
- When a key decision about a design in made in a closed forum, take care to summarize the problem, discussion, and conclusion/decision in a public forum as soon as possible following the discussion.
- Whenever data is gathered, work to ensure it can be licensed under as open a license as possible so that the community can reap the full benefits of it. When a restricted license is necessary, openly sharing the reasons why upfront can help avoid community backlash.

The Fedora Project blog planet frequently has examples of design discussion summaries. In addition, the Fedora wiki Design page has a repository of design artifacts and source in a format that is quickly becoming the standard for such design projects.

Upstream your work

As previously discussed, it can be difficult to discern which FLOSS community to get involved with given a particular design problem to solve. When in doubt, start with the most 'upstream' project involved in the domain. This ensures that any design work you offer to the FLOSS community as a whole benefits that community as widely as possible. For example, if you wish to work on improving the design of a font selection dialog on FLOSS desktops, if you focus your work on Fedora only, you limit the impact your design can have. If you work on your design one step upstream from Fedora, in the GNOME community, the KDE community will not be able to benefit as easily from your design input and again your design reach is limited. If you try to work one more step upstream, under the Freedesktop.org project, your design will have the opportunity to impact both the GNOME and KDE desktop projects. Working upstream increase the reach of your design and its benefits to FLOSS in general.

It can be difficult to determine what the appropriate upstream project is for any given design issue. If you communicate with fellow community members that you would like to work as far upstream as possible, they are likely to be able to help you identify the most appropriate place to jump in.

Adapt to the communication methods

Many communication methods and habits are ingrained in the FLOSS community from well over a decade of precedent. While some change may be possible through patience and time, adapting to the etiquette and syntax of current modes of communication will help designers embed themselves within the community.

Finding a mentor within the community who has the time and experience to explain the communication methods and standards is the best way for a designer to address this challenge. Many FLOSS projects have mentorship and other on-boarding programs for new contributors that serve as a good way to find such a mentor. The Fedora Project, for example, has an official mentors program [15] as well as well-documented communication guidelines and etiquette [12].

Get user feedback early and often

As in any design process, obtaining user feedback early and often is important for a FLOSS design process. One difference between FLOSS and non-FLOSS user feedback, however, is that the very process of seeking FLOSS user feedback can help build support for design projects. The FLOSS community in general is more receptive to work that is produced in the open from the start rather than the 'cathedral' model of private creation and refining with a later release out in the open upon completion [29]. By presenting in-progress design works in calls for user feedback, designers provide early access to design work in the community making it possible for community members to weigh in on design decisions before the final initial release of the design project.

Provide mentorship

As the design capacity for many FLOSS projects is woefully low, it is important for those designers who are involved to reserve time for mentorship – both in mentoring fellow designers in becoming effective FLOSS contributors, and mentoring current FLOSS contributors in design practice. As Benson suggests, teaching developers user-centric techniques such as paper prototyping could help them develop greater understanding of design in order to improve their applications [2]. Capacity-building through mentorship is critical for design to have a bigger positive impact on FLOSS.

To this end, the Fedora Design team identified specific design projects - such as the website for the Fedora Games Spin, a special edition of Fedora targeted for gamers – as mentorship opportunities and structured the projects in ways that made it easier for Fedora contributors to work in a design capacity and learn design skills along the way:

- **Break the design project into small, manageable chunks and explicitly assign them:** Over 100 games are part of the Fedora Games Spin. The website design needed to provide a catalog of these games. In order to enable open participation, the list of games was divided into much smaller, manageable chunks and potential contributors within the community were asked to claim a chunk of games to work on a design for in the game catalog. 7 Fedora contributors stepped forward, were explicitly assigned tasks, and the team was able to pull together 127 game catalog entries in the final weeks leading up to the Fedora 12 Games Spin release. Explicitly assigning tasks to design contributors may be a way to enable designers to jump the intimidating model of self-assignment in FLOSS noted in the literature [8].
- **Facilitate discussion and indicate clear deliverables:** Following the initial Fedora Games Spin website release, the Fedora Design Team identified a design problem – how can we make it easier to browse the 127 games in the catalog? A discussion was started on the team list, various solutions discussed, and a clear deliverable defined – a mockup of a catalog browsing interface. Contributors shared different mockups of what the interface could look like. The discussion continues today. Making sure a mockup of the interface as a deliverable was clearly defined to contributors and coaching them in its creation and providing gentle critique helped in continuing the design process.
- **Give credit where credit is due:** After a design contributor, especially for new design contributors, contributed to the Games Spin website project, the Fedora Design Team lead ensured that a timely thank you identifying the contributor and their specific contributions and achievements was shared in the Fedora Community, on the Fedora Planet shared blog. As a result of this positive feedback, contributors involved in the project continue to stay engaged in the design team. Eric S. Raymond notes a similar tactic aids Linus Torvalds in the development of Linux, “Linus was keeping his hacker/users constantly stimulated and rewarded—stimulated by the prospect of having an ego-satisfying piece of the action” [2].

Sim and Holt similarly recommend mentorship as a pattern for success in ramping up developer “software immigrants” in software projects, although they mention it can be inefficient [33].

The opportunity to serve as a mentor, as time-consuming as it may prove, can prove richly rewarding. For example, I presented a session on 'Designing UI Mockups in Inkscape' (a popular FLOSS design program) at the FUDcon Toronto Conference on 5 December 2009. Only hours after attending my talk, Adam Miller, a developer in the Fedora Community member, participated in a hackfest focused on improving a software application by creating mockups using the design skills he had just learned [27].

CONCLUSION

This paper was written with the hope of providing insight about working within the FLOSS community to HCI researchers and practitioners unfamiliar to FLOSS, as well as serving as a springboard for further discussion on how the FLOSS and HCI communities can mutually benefit each other. The open manner in which FLOSS is created provides a potential real-life example of the software development process in action in which HCI instructors and students can participate to further HCI theory and practice. The FLOSS community is sorely in need of input from HCI practitioners in order improve the usability, accessibility, and utility of FLOSS for greater adoption worldwide.

REFERENCES

1. Bach, P. M., DeLine, R., and Carroll, J. M. 2009. Designers wanted: participation and the user experience in open source software development. In *Proceedings of the 27th international Conference on Human Factors in Computing Systems* (Boston, MA, USA, April 04 - 09, 2009). CHI '09. ACM, New York, NY, 985-994. DOI= http://doi.acm.org/10.1145/1518701.1518852

2. Benson C. 2004. Meeting the challenge of open source usability. Interfaces. Autumn 2004. 9-12. http://www.bcs-hci.org.uk/files/interfaces/interfaces60.pdf

3. Benson, C., Muller-Prove, M., and Mzourek, J. 2004. Professional usability in open source projects: GNOME, OpenOffice.org, NetBeans. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems* (Vienna, Austria, April 24 - 29, 2004). CHI '04. ACM, New York, NY, 1083-1084. DOI= http://doi.acm.org/10.1145/985921.985991

4. Bird, C., Gourley, A., Devanbu, P., Swaminathan, A., and Hsu, G. 2007. Open Borders? Immigration in Open Source Projects. In *Proceedings of the Fourth international Workshop on Mining Software Repositories* (May 20 - 26, 2007). International Conference on Software Engineering. IEEE Computer Society, Washington, DC, 6. DOI= http://dx.doi.org/10.1109/MSR.2007.23

5. Brafman, O. and Beckstrom, R. 2008 The Starfish and the Spider. Portfolio Trade.

6. Callaway, T. Fedora Project Licensing, https://fedoraproject.org/wiki/Licensing (2009).

7. Creative Commons, http://creativecommons.org (2009).

8. Crowston, K., Li, Q., Wei, K., Eseryel, U. Y., and Howison, J. 2007. Self-organization of teams for free/libre open source software development. Inf. Softw. Technol. 49, 6 (Jun. 2007), 564-575. DOI= http://dx.doi.org/10.1016/j.infsof.2007.02.004

9. Duffy, M. “Want to learn design skills? Want to help Fedora? Fedora Interaction Design Hackfest, Tuesday 24 Nov", http://mairin.wordpress.com/2009/11/20/want-to-learn-design-skills-want-to-help-fedora-fedora-interaction-design-hackfest-tuesday-24-nov/ (2009).

10. Duffy, M. “Fedora Interaction Design Hackfest Summary", http://mairin.wordpress.com/2009/11/24/fedora-interaction-design-hackfest-summary/ (2009).

11. Fedora Project Ambassador Team Membership Map, https://fedoraproject.org/membership-map/ambassadors.html (2009)

12. The Fedora Project: Communicating and Getting Help, http://fedoraproject.org/wiki/Communicate (2009).

13. The Fedora Project Frequently-Asked Questions, http://fedoraproject.org/wiki/FAQ (2009).

14. The Fedora Project Individual Contributor License Agreement, http://fedoraproject.org/wiki/Legal/Licenses/CLA (2009).

15. The Fedora Project Mentors Program, https://fedoraproject.org/wiki/Mentors (2009).

16. Fedora Project Overview, https://fedoraproject.org/wiki/Overview (2009).

17. Fedora Project Statistics, https://fedoraproject.org/wiki/Statistics (2009).

18. Fedora-FR – the French-Speaking Fedora Community, http://www.fedora-fr.org/ (2009).

19. Freedesktop.org, http://freedesktop.org/ (2009).

20. Frishberg, N., Dirks, A. M., Benson, C., Nickell, S., and Smith, S. 2002. Getting to know you: open source development meets usability. In *CHI '02 Extended Abstracts on Human Factors in Computing Systems* (Minneapolis, Minnesota, USA, April 20 - 25, 2002). CHI '02. ACM, New York, NY, 932-933. DOI= http://doi.acm.org/10.1145/506443.506666

21. Ghosh, R. A. Understanding Free Software Developers: Findings from the FLOSS study. Perspectives on Free and Open Source Software. Ed. J. Feller, B. Fitzgerald, S. A. Hissam and K.R. Lakhani MIT Press, Cambridge, 2005.

22. GNOME: The free software desktop project,

http://gnome.org/ (2009).

23. Green, C., Tollinger, I., Ratterman, C., Pyrzak, G., Eiser, A., Castro, L., and Vera, A. 2009. Leveraging open-source software in the design and development process. In *Proceedings of the 27th international Conference Extended Abstracts on Human Factors in Computing Systems* (Boston, MA, USA, April 04 - 09, 2009). CHI EA '09. ACM, New York, NY, 3061-3074. DOI= http://doi.acm.org/10.1145/1520340.1520433

24. Hentschel, Elke. Communication on IRC. Linguistik online 1/98. http://www.linguistik-online.de/irc.htm

25. The K Desktop Environment, http://kde.org/ (2009).

26. Karels, M. J. 2003. Commercializing Open Source Software. *Queue* 1, 5 (Jul. 2003), 46-55. DOI= http://doi.acm.org/10.1145/945074.945125

27. Miller, A. FUDCon - Day one : The awesomeness that is, http://pseudogen.blogspot.com/2009/12/fudcon-day-one-awesomeness-that-is.html (2009).

28. Nichols, D.M. and Twidale, A.B. The Usability of Open Source Software. *First Monday*, 8, 1 2003. http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/1494/1409

29. Raymond, E. S. 2001 *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly & Associates, Inc.

30. Rigby, P. C. and Hassan, A. E. 2007. What Can OSS Mailing Lists Tell Us? A Preliminary Psychometric Text Analysis of the Apache Developer Mailing List. In *Proceedings of the Fourth international Workshop on Mining Software Repositories* (May 20 - 26, 2007). International Conference on Software Engineering. IEEE Computer Society, Washington, DC, 23. DOI= http://dx.doi.org/10.1109/MSR.2007.35

31. Scacchi, W. Free/Open Source Software Development: Recent Research Results and Methods Advances in Computers. Ed. M. Zelkowitz Academic Press, San Diego, 2008. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.80.9111&rep=rep1&type=pdf

32. Schwartz, D. and Gunn, A. 2009. Integrating user experience into free/libre open source software: CHI 2009 special interest group. In *Proceedings of the 27th international Conference Extended Abstracts on Human Factors in Computing Systems* (Boston, MA, USA, April 04 - 09, 2009). CHI EA '09. ACM, New York, NY, 2739-2742. DOI= http://doi.acm.org/10.1145/1520340.1520395

33. S. Sim and R. Holt. The Ramp-up Problem in software projects: A case study of how software immigrants naturalize. 20th Int. Conference on Software Engineering, pages 361–370, 1998. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.93.3416&rep=rep1&type=pdf

34. Transbot Project Page, https://fedorahosted.org/lingobot/ (2009).

35. Trudelle, P. “Shall We Dance? Ten Lessons Learned from Netscape's Flirtation with Open Source UI Development.” In CHI 2002 Getting to Know You: Open Source Meets Usability Workshop. 2002. http://www.iol.ie/~calum/chi2002/peter_trudelle.txt

36. Wenger, E., McDermott, R., Snyder, W.M. 2002) *Cultivating Communities of Practice.* Harvard Business Press.

37. X.org Foundation, http://x.org/ (2009).