

# "Suck It Up, Princess": Outreach and Diversity in FOSS Communities

Karen Rustad

13 May 2011

## Introduction

The development and proliferation of free and open source software (FOSS) is a well-loved example in the academic literature, used to explain the phenomenon of decentralized, gift-economy, post-industrial "barn-raising" and situate them as viable modes of production (Kollock 431-2). While FOSS has been well analyzed from technical and economic standpoints, there is surprisingly little sociological or anthropological work on the people who belong to the FOSS community and contribute to FOSS projects. Who it is that builds these crucial underpinnings of the information economy likely has significant implications for what kinds of software get built, who can use it, and any number of other downstream social and economic effects.

Massive volumes of ink and bits have been spilled documenting the demographics of open source software contributors and lamenting the hugely disproportionate lack of women and non-white people among them. Surveys of FOSS projects have found that women make up, at best, 5 percent of FOSS contributors; a more likely figure is around 2 percent (Ghosh). For many FOSS contributors, these figures are unfortunate but immutable constants; within the community there is skepticism that they can be improved much, if at all. Most FOSS communities see themselves as meritocracies; many participants honestly believe that all that matters with regard to whether you can join or gain status within the community is whether or not you produce smart ideas and good code. As the current chair of the Python Software Foundation, Jesse Noller, writes,

I truly, fundamentally believe that OSS is the perfect proof that *anyone* and *everyone* can contribute to something. Why? Code and projects are blind – or if they aren't, they should be. They don't care if you're a sentient ham sandwich, they don't care if you don't have the proper schooling, how old you are, the color of your skin, or your religious beliefs. They don't care where you are located or what you did before you contributed something. They just care about what you

have contributed – doc patch, code patch or feature. That thing you have submitted should be able to stand on its own merits, regardless of anything outside of that atomic change. (Noller)

Consistently with this perspective, most FOSS developers attribute the lack of diversity in FOSS to women and other groups simply not being interested in contributing (Krieger 16). This purist meritocratic outlook is noble in some respects, but it ignores or even masks the realities of community politics, nasty flamewars, bias incidents – all the messy human factors that even geeky, avowedly-rationalist FOSS developers are heir to and which have real effects on who chooses to work on FOSS and who does not.

Clearly part of FOSS' diversity problem is inherited from the larger technology community; only 20 to 30 percent of tech industry workers are female, and women's share of undergraduate computer science majors has dropped from 20 percent to as low as 12 percent (Krieger 4, Stross).<sup>1</sup> However, even that low proportion is almost an order of magnitude higher than FOSS's share of women. Additionally, FOSS also has disproportionately few South Asian contributors, despite the fact that this group is actually overrepresented in the tech industry as a whole (Laroia, "Diversity"). Clearly, therefore, there exist barriers to diversity in FOSS that are not simply the same issues plaguing CS departments and technology companies in general, but are unique to the FOSS community.

Particular sexist incidents within the FOSS community, such as FOSS conference presenters including pornographic images in their slide decks<sup>2</sup>, the head of the Free Software Foundation declaring that it was a sacred duty of community members to relieve "women who have never used EMACS" of their "EMACS virginity"<sup>3</sup>, and cases of nonconsensual groping and

1 A selection of theories of why there are fewer women/minorities in tech: less socioeconomically advantaged kids are less likely to have had a personal computer in the 1980s (Restructure!) and the average age of first having one's own computer is significantly higher for girls than for boys (Krieger); computer science department culture is less appealing to women (Margolis & Fisher, 2002); and repeated failure is part and parcel of coding and boys are better taught to handle failure than girls (Scientist Carrie).

2 [http://geekfeminism.wikia.com/wiki/CouchDB\\_talk](http://geekfeminism.wikia.com/wiki/CouchDB_talk)

3 [http://geekfeminism.wikia.com/wiki/EMACS\\_virgins\\_joke](http://geekfeminism.wikia.com/wiki/EMACS_virgins_joke)

sexual assault at FOSS conference after-parties<sup>4</sup>, are fairly well-documented (Aurora). Such incidents are often cited as the reason for why so few women are active in FOSS. However, people typically would only encounter such explicitly sexist attitudes and incidents once they were already involved in FOSS; such incidents certainly could turn off new female contributors and make them leave the community, but it seems unlikely that they would affect their rate of initial entrance into FOSS so dramatically as to be the only factor at work.

Less well explored is implicit sexism in how open source projects systemically attract and retain new contributors (or, more accurately, usually fail to). In this paper, I argue that unintentionally sex-biased mechanisms of "recruitment" in themselves may account for FOSS' demographic imbalances.<sup>5</sup> If so, this suggests that the recent wave of new outreach efforts to improve the experience of first-time contributors and provide clearer entry points into FOSS may have the side effect of remedying the community's diversity problem. I focus mainly on sex bias because that disparity has the greatest amount of relevant coverage and commentary within the FOSS community. However, the goal is to point out unacknowledged obstacles and filters in FOSS recruitment that would likely account for some amount of underrepresentation of racial, ethnic, linguistic, and other minority groups as well.

## Why do FOSS contributors get involved?

It may be a surprise that a majority – about 54 percent – of FOSS contributors benefit materially from their participation in the FOSS community (Ghosh). Eighteen percent of FOSS contributors are paid to administrate FOSS; almost 16 percent are paid to develop it. FOSS

---

<sup>4</sup> Recent account of one such assault: <http://blog.nerdchic.net/archives/418/>

<sup>5</sup> In this paper, I use the term "recruitment" loosely to describe the ways in which new contributors become involved in FOSS. However, it should be noted that most FOSS projects do not actively "recruit" in any traditional sense of the word (e.g. athletic recruitment). At best, projects seek to make it *less difficult* for outsiders to submit patches or other contributions; the "recruitment" interaction is almost always passive on the part of the project. This arguably may be a weakness of FOSS communities and contribute to FOSS projects' low overall survival rate. Notably, the new wave of FOSS outreach is significantly more proactive than previous efforts.

developers also benefit indirectly from their FOSS participation; almost eighteen percent of contributors report getting a job because of their FOSS experience.

Much academic literature and research in motivations and organizations is devoted to the distinction between intrinsic and extrinsic motivations. Specifically, intrinsic motivations apply to "activit[ies] undertaken for one's immediate need satisfaction. Intrinsic motivation 'is valued for its own sake and appears to be self sustained'" (Osterloh & Frey 539). Extrinsic motivation, on the other hand, applies to activities that "satisfy [participants'] needs indirectly, especially through monetary compensation" (Osterloh & Frey 539). These two forms of motivation do not typically mix evenly; experiments have found that introducing extrinsic motivators into a primarily-intrinsically motivated activity reduces participants' levels of intrinsic motivation. This effect, called the "crowding out" effect, is sometimes so severe that the addition of extrinsic motivators actually lowers participants' overall motivation levels (Osterloh & Frey 541-2). One might expect that the external rewards that many contributors now receive for working on FOSS software, then, would reduce the amount of intrinsic motivation within the community.

Nonetheless, FOSS contributors' motivations remain primarily intrinsic and appear to be "un-crowded-out." Ghosh et al's 2002 survey found that the number one reason developers get involved with FOSS was to learn and develop new skills, selected by almost 80 percent of respondents. About one-third of respondents join FOSS to "participate in a new form of contribution"; almost half participate to share their knowledge and skills; thirty percent are motivated by the ideological belief that software should not be a proprietary good (Ghosh). Extrinsic motivational factors are significantly less widely held; only 23 percent of respondents joined FOSS to improve their job opportunities, and only 4 percent join to make money (though this figure jumps to 12 percent as a reason to remain a contributor) (Ghosh). It

appears that FOSS contributors start out nearly universally as volunteers, then eventually either find paid FOSS-related work or notice benefits from their FOSS experience in their job search generally. Given the continued importance of intrinsic motivational factors among experienced FOSS developers, extrinsic rewards for FOSS work do not appear to crowd out their initial motivations. Research specifically into the motivations of contributors to the Apache Software Foundation's web server project, Jakarta project, and XML project (all free software) support this conclusion; being paid to work on Apache did not affect contributors' intrinsic motivations, yet did increase their level of contribution as well as their status motivations to contribute (Roberts).

### **How are FOSS contributors typically recruited?**

Most current FOSS contributors underwent a slow progression from "newbie" to recognized contributor with commit access to a given FOSS project. This traditional mode of recruitment is uncoordinated and almost entirely passive on the part of the project, yet appears regularly in retrospective accounts of how current developers got involved (Bolton). First, the individual is exposed to a given software project, prompted by their need for a given piece of software for school, work, or personal use. The individual may also be exposed to free software generally, perhaps through a work such as Richard Stallman's *Free Software, Free Society* or the GNU Manifesto. The individual uses the software and perhaps lurks on the software project's mailing list. After a period of time – often years – the individual finds a bug in the software or finds that they have a need that the software does not quite meet. Given sufficient motivation and technical savvy, the individual writes a patch of the software to "scratch their itch" and sends it to the project's maintainer. If the patch meets the project's standards and the project maintainer agrees that it is an improvement, he or she eventually

integrates the patch into the official version of the software. After a few repetitions of the itch-patch-submission cycle, the project maintainer is likely to decide that the individual is trustworthy and useful enough to merit commit access for the project. While technically speaking, the individual has been a "contributor" to FOSS ever since they wrote their first patch, typically it is only at this point that the individual is formally recognized as being part of the FOSS project's community.

While most current FOSS contributors got involved through this traditional decentralized, depersonalized "recruitment" process, FOSS projects are starting to develop and use more modern, interventionist methods of recruiting new contributors. For example, Ubuntu has organized special "hug days" (bug-triaging days) where they explicitly ask for help from potential contributors on their mailing list and make sure that their IRC channel is staffed with veteran developers to assist new people.<sup>6</sup> In spring 2011, FOSS outreach project OpenHatch partnered with desktop FOSS projects such as GIMP and Vidalia to create "Build It" days – special events where a given project invites new contributors to get involved and provided special advice and assistance in getting the software built on their computers (so that they would have the necessary development infrastructure set up and working so they could change the software and write patches later).<sup>7</sup> FOSS-related conferences typically offer "sprints" – a couple days of the conference, usually taking place after the official talks and sessions, for participating projects to spend coding and recruiting new members from the conference attendees.<sup>8</sup> Sometimes FOSS projects will organize in-person meetups and hackathons on their own as well. These in-person hacking sessions provide a clear, focused space for development and community-building.

---

6 <https://lists.ubuntu.com/archives/ubuntu-devel-announce/2009-August/000605.html>

7 <https://openhatch.org/blog/2011/april-build-it-week/>

8 Example: <http://us.pycon.org/2011/sprints/>

Some large FOSS projects also have explicit mentorship programs and support groups, intended to help get new contributors involved and make the community more welcoming. Some groups, such as Debian Women and Ubuntu Women, are specifically targeted at helping minority populations get a foothold in FOSS.<sup>9,10</sup> Others, such as Fedora Mentors, are broadly targeted to new contributors of all sorts.<sup>11</sup> Both sorts of mentorship programs have varying levels of success, depending on how many willing, energetic mentors they can attract and how effectively they make potential mentees aware of their existence.

Finally, likely the most formalized entry point into FOSS development to date is Google's Summer of Code (or GSoC) program. GSoC gives students stipends to spend the summer working on improvements for a particular FOSS project. Since 2005, GSoC has brought together over 4500 successful students and over 3000 mentors (Google). While GSoC has been highly successful in generating open source code and giving students real-world programming experience, its record on integrating new contributors into the FOSS community is more spotty. In 2006, at least, only 30 percent of GSoC participants continued to contribute to their mentoring organizations' projects after the end of the summer (Kerner). The proportion of continuing contributors varied widely from project to project, however, and there is some thought that practice and advice on how to mentor well might increase this figure; to that end, for the 2011 batch of GSoC participants FOSS veterans Donnie Berkholz, Lydia Pintscher, and Kevin Smith posted a list of dos and don'ts for GSoC mentors on Google's open source blog.<sup>12</sup>

The additional structure and formalism of GSoC appears to attract contributors who would otherwise have been too intimidated to get involved in FOSS. As current co-maintainer

---

9 <http://women.debian.org/home/>

10 <http://ubuntu-women.com/>

11 <http://fedoraproject.org/wiki/Mentors>

12 [http://google-opensource.blogspot.com/2011/04/dos-and-donts-of-google-summer-of-code\\_21.html](http://google-opensource.blogspot.com/2011/04/dos-and-donts-of-google-summer-of-code_21.html)

of FOSS project Drupal and 2005 GSoC alumna Angela Byron stated in an interview:

I held a...strong belief that everyone who actually worked on open source software was *really smart* and that I, being only a lowly community college student, couldn't possibly participate on that level. [...] Then...one of my instructors told me about Google Summer of Code (GSoC), a program which provides stipends to students to work on open source projects over the summer. This poked a *tiny* little hole in the "you must be THIS smart to contribute" wall I had built up in my head, because I figured, hey, they know I'm a student so they must know I don't know *everything* yet... So I figured I would apply and see how it went. (Druckman)

Even despite uneven success in helping students build enduring relationships to the FOSS projects they work on over the summer, by creating a clear, concrete entryway into FOSS development and making it clear that that entryway is not just for the best and brightest but for anyone seriously interested in helping write code, GSoC's structure has helped contributors join FOSS who would not have otherwise been motivated or successful under the community's tradition of benign neglect in recruitment. Through GSoC and other recent, more-proactive innovations in FOSS recruitment, many projects are discovering, in a nutshell, that "there are people waiting to be included, who are just, *not*, and if we simply ask them, they would join."

(Laroia, "Get")

## **What assumptions are embedded in current FOSS recruitment practices?**

Typical passive methods of FOSS recruitment filter the pool of potential contributors in three main ways, each with implications for why women or other minorities would be less likely to get involved.

### **1.) Homogeneity Through Friend Networks**

An individual's initial contact with FOSS often happens person-to-person, via a friend, sibling, or classmate.<sup>13</sup> Stereotypically, first FOSS contact looks like two teenagers hacking on a

---

<sup>13</sup> A number of stories of users' first encounters with Linux can be found at <https://bbs.archlinux.org/viewtopic.php?id=15691&p=1> and <http://www.hackertthreads.org/Topic-32787>.

PC in the basement, and one of them suggesting that they install Linux for fun. Given FOSS's existing gender and ethnic makeup, the teenager making the suggestion is almost certainly white, male, and geeky; because "birds of a feather flock together," and young people's friendship networks tend to be homophilous on gender and ethnic dimensions (among others), the second teenager probably shares these characteristics (McPherson, Smith-Lovin, & Cook). Similar effects occur later on as well, in contexts such as FOSS user groups or meetups: "One of the largest reasons our meetup groups are less than five percent female, in a technology world where 20 to 30 percent of the people are female,...is we invite people to the meetup group who we know and our friends are probably people like us" (Laroia, "Get"). Homophilous diffusion of FOSS naturally helps perpetuate its current demographic monoculture.

## **2.) Requisite Time, Confidence, and Previous Exposure**

Most FOSS projects have poorly-written, incomplete documentation. A significant number have virtually no documentation at all. Since the FOSS community is dominated by developers (with few managerial types, technical writers, designers, or other sorts of stakeholders that typically contribute to software within firms), projects both lack sufficient volunteers with the appropriate skill sets to write good documentation and tend to underprioritize documentation writing and other non-coding parts of their project. As Gina Trapani notes,

[I]nterface design [along with other non-code contributions] doesn't get nearly the respect and prioritization it should by typical OSS engineers. [...] The solution isn't simply getting more designers and UX experts involved in open source, it's something much more difficult: changing community values around those skills. Miliano writes:

For a designer to contribute to an open source project, there would have to be developers committed to implementing the work, to work on "polish" and "froofy things" instead of "real features" and other "important things." That's a hard nut to swallow, and what volunteer

project owner will ask all their volunteer contributors to, please, stop working on your pet projects within this codebase and let's actually cut features and work on UI and usability and design?

[...] The key here is to prioritize design and usability upfront, rather than accept a mess of software with plans to slap a pretty veneer on afterwards. (Trapani)

Recently there have been a few initiatives to make good, complete documentation an integral part of FOSS development, most notably the Read the Docs project.<sup>14</sup> However, the need for this project and the fact that it stands out prove the poorly-documented rule.

Poor documentation drastically narrows the pool of potential contributors to a given FOSS project in three ways. **First, it selects for highly motivated people with a lot of free time.** Figuring out how to download, satisfy dependencies for, modify, build, and push patches for a given piece of software is far more frustrating and takes far longer if the steps for how to do so are vague, erroneous, or incomplete. This disproportionately filters out potential female contributors because most women have significantly less leisure time than men. On average, male undergraduates have more than twice as much free time as their female counterparts (Heeter). This gap continues into adulthood; while seven out of ten parents believe that child care duties should be distributed equally between parents, moms report being the primary caregiver two-thirds of the time (Dunnewind). Women also "spend about three to seven times as many hours as men on cleaning and laundry tasks" (Dunnewind). With women disproportionately taking on the "second shift" of running dual-income households, they are less likely to have the time and energy to pursue hobbies such as FOSS development.

**Second, poor documentation selects for people who are highly confident in their technical skills, irrespective of their actual ability.** Getting involved in a FOSS project for the first time, especially one without good documentation, is an act of faith. Without help and

---

<sup>14</sup> <http://readthedocs.org/>

support from veterans of the project, there is no guarantee that a new contributor's flailing efforts to get their dev environment working will *ever* pay off in the form of improvements and patches. This is especially true if it is one's first time contributing to a project. Veteran developers may not be familiar with the specific technologies or packages used by a new project that they want to work with, but because they have gone through the build process successfully before, they are more likely to have faith that they will figure it out eventually. In the absence of effective documentation or mentorship, new developers must substitute experience with sufficient bravado and confidence in their own technical capabilities to believe that sooner or later, "when things go wrong, you can solve it" (Buckland).<sup>15</sup>

Men are more likely to have this critical level of "bravado" than women. Women hold themselves to a higher standard than men, particularly in science and technology matters. In a study of gender and self-assessment using a fictitious spatial reasoning skill, Correll found that "women and men held different standards for what constituted high ability... [W]omen believed they had to earn a score of at least 89 percent to be successful, but men felt that a minimum score of 79 percent was sufficient" (Hill 65). Women also rate their abilities lower than equivalently-talented men do. (Hill 61) As Jean Hsu writes,

When I walked by a departmental career fair, I paused to look at some of the companies I might want to apply for next semester, and [Professor Li] told me to sign up for some interviews for that day. I said I didn't feel prepared and wanted to wait a semester until I felt like I had more of a basic foundation. He turned to a professor next to him and said "Jean doesn't know how good she is." He probably would not remember that exchange, but for me his support was eye-opening. I realized that while I did decide to major in CS fairly late in the game, I really was good at it, and my harshest critic was really myself. (Hsu)

Having high standards for themselves may push women to succeed – indeed, by some measures women actually do better than men in required science and technology-related

---

<sup>15</sup> See especially the distinction made between "category A" people and "category C" people in the lecture, starting about fifteen minutes in: <http://www.youtube.com/watch?v=hE716Adoiiw>.

classes. (Hill 21) However, the accompanying erosion of women's confidence in their abilities also closes many doors. High-achieving women in academic and technical fields often also struggle with "impostor syndrome," denigrating or explaining away their achievements as luck or accident rather than daring to attribute them to their own abilities and hard work.

(Kanazawa) While both men and women can suffer from impostor syndrome, it appears to be more common among women. This pervasive feeling of being an "impostor," "fraud", or "wannabe" makes people

be less willing to put themselves forward, feeling that they are not qualified, by eg: not applying for jobs, promotions, and other employment opportunities; not submitting papers to conferences or journals; disclaiming or understating their experience/skill when speaking or writing; or nervousness about talking to others in their field, especially if those others are perceived as highly skilled/experienced" (Geek Feminism Wiki, "Impostor")

The confidence gap between men and women does not simply result in harmless humility – it affects women's willingness to take risks and the career paths that women eventually choose. As Correll found,

boys were more likely than their equally accomplished female peers to enroll in calculus [and and choose a college major in science, math, or engineering] not because boys were better at math but because they believed that they were better at math. [...] If girls do not believe they have the ability to become a scientist or engineer, they will choose to be something else. (Hill 61)

Thus, it is likely that the confidence gap between men and women, particularly in relation to self-assessed technical ability – could filter out many potential female contributors as they quickly become discouraged when first attempting to work on a FOSS project or, believing themselves not "l33t" ("elite") enough, never try to get involved in the first place. The reason for Angela Byron's delayed entry into FOSS contribution bears this hypothesis out:

I've been interested in free software ever since I first heard the term back in 1995, back when I completed my first successful Linux installation... I became a fierce advocate of open source alternatives among my family and friends, and I was totally "that person" in school... However, I held an equally strong belief that

everyone who actually worked on open source software was *really smart* and that I, being only a lowly community college student, couldn't possibly participate on that level. So I would silently cheer these people on, [...] wistfully hoping that one day, far into the future, maybe after I had 30 years of experience or something, I could someday join their ranks.

[...] Once I got to "this side" of the contributor wall I realized how silly the myths I had in my head about open source contributors was. While there are indeed very Einstein-level smarties out there contributing to open source, most of them are just people like me... (Druckman)

Women are also more likely than men to see their talents and intelligence as fixed, unmutable values. Linda Babcock and several colleagues developed a survey and scale to measure individuals' positions on a "turnip" to "oyster" spectrum – whether one's attitude better resembled "you can't get blood from a turnip" (a fixed, accepting mindset) or "the world is your oyster" (a flexible, opportunistic mindset) (Babcock & Laschever 19). They found that women were 45 percent more likely than men to score on the "turnip" side of the scale (Babcock & Laschever 20). This attitude differential likely stems from differences in how girls and boys are taught in school:

Girls, who develop self-control earlier and are better able to follow instructions, are often [...] told that we are "so smart," "so clever," or "such a good student." This kind of praise implies that traits like smartness, cleverness and goodness are qualities you either have or you don't.

[O]n the other hand, [...] just trying to get boys to sit still and pay attention is a real challenge for any parent or teacher. As a result, boys are given a lot more feedback that emphasizes effort (e.g., "If you would just pay attention you could learn this," "If you would just try a little harder you could get it right.") (Halvorson)

Thus, when women encounter an obstacle, they are more likely to see it as an inexorable sign of their own incompetence, while a man encountering similar obstacles is likely to be "trained - intensely trained - to fail and select retry, that is to say, to iterate" and to approach them "as interesting, eminently achievable, potentially fun exercises with a confidence borne of past successes" (Tycho, qtd in Scientist Carrie). Social and developmental psychologist Carol

Dweck's research into what she calls "fixed" versus "growth" mindsets comes to similar conclusions (Dweck & Leggett, 1988; Blackwell et al., 2007; Dweck, 2006, 2008; cited in Hill 47) . This difference in perspective drastically affects one's well-being and chances of success.

If you've internalized the message that not immediately mastering something means you aren't "good" and "smart" -- that tricky [puzzle] is just miserable and demoralizing. Even if you finally manage it, it feels like a hollow victory. Your repeated failures just prove how stupid you are and how terrible you are...  
(Scientist Carrie)

If a person with this attitude and low self-confidence in their technical abilities is unable to get their chosen FOSS project running on the first or second try, they are likely to conclude that they are simply not smart enough to participate and give up rather than (in order of usefulness) blaming the erroneous or incomplete project documentation, dispassionately experimenting to try and figure out the project build "puzzle" on their own, or asking for help.

**Finally, poor documentation selects for people who have contributed to other FOSS projects and are already familiar with the various trappings thereof.** If you ask people what knowledge is necessary to work on a given open source software project, they usually focus on what languages and libraries the project's code is written in or with. In truth, familiarity with a project's programming language is only a tiny sliver of the knowledge that is necessary in order to contribute code, and much of that knowledge is tacit – "acquired and stored within individuals," uncaptured in writing and thus nontransferable (Osterloh & Frey 539). Common hidden prerequisites include comfort using the command line, knowledge of bash commands, ability to effectively use common version control systems, familiarity with bug trackers, and so forth. Additional tacit knowledge covers how, exactly, these tools are expected to be used in a given project (e.g. separating out different changes into separate branches and patches, marking bugs with the appropriate level of severity and tagging them with the right people, code styling conventions, etc.). Veterans of a given project are likely to

underestimate the difficulty of getting up to speed with the array of tools their project uses because they are so used to using them – they may have even helped choose what infrastructure to use for the project in the first place.

To be fair, it would be a lot of work for each project to write project-specific tutorials and style guides to cover all this background knowledge. But many projects do not effectively acknowledge all their hidden prerequisites or link to others' tutorials and resources for each of them; much of their project's tacit knowledge could be made more explicit, but simply has not been. Thus, the most likely new contributors to a given FOSS project are likely to have already contributed to a different FOSS project with similar infrastructure and prerequisites, shrinking FOSS's potential recruitment pool and reinforcing FOSS's existing demographic profile.

### **3.) Assertiveness, Persistence, and a Thick Skin**

Without formalized recruitment and mentorship procedures or a pre-existing relationship to a current contributor, new FOSS participants must be willing from the very beginning to "put themselves out there" – initiate their own participation on faith that their contribution is wanted and needed, ask "stupid" questions of unknown veteran developers, nag project leaders to review their patches, defend the merits of their approach to a problem on mailing lists, and so forth. All of these activities may be challenging for shy people in the best of circumstances; for a new contributor with no trusted contacts within the project and less than perfect confidence in one's own technical savvy, such alienating, intimidating environs are likely to lead to them running away or never trying to contribute in the first place. As one contributor to Debian, a FOSS project notorious for being especially prickly, recounted,

In 2004 or so, I saw Debian as the cool kids' club, that awesome project that I wished I could be a part of. By 2006, I managed to get over myself, read the New Maintainer's Guide, and find a way to get involved. As of mid 2009, I am a full-blown Debian Developer. I have real ultimate power. But I sometimes do still feel hesitation akin to "Imposter Syndrome". [...]

In the past year of being a Developer, one thing I've seen is that other contributors ask me privately for help. Rather than blast the public lists like debian-mentors [a list intended to help new contributors], they email or IRC private-message me, or SMS me, or find me at a Linux Users Group event. I'm lucky to know these people, and they're lucky to have me as a safe person to ask questions of. [...] [At the "Debian for Shy People" session,] I asked, "How many of you have someone you can ask embarrassing questions of?" Of the forty people crammed into [the room], *two people* raised their hands. One person put it plainly, "I don't know anyone else who does Debian." (Laroia, "Debian")

Since becoming a Debian Developer, Laroia has gone on to be extensively involved in improving FOSS outreach. Without others in the community to safely ask questions of, and a developer environment that is indifferent or even hostile to newcomers, it is easy for potential contributors' shyness to prevent them from getting involved for years, or ever.

Women are especially likely to be limited by this sort of anxiety. Linda Babcock's research found that women score significantly higher than men on a scale of "negotiation apprehension", with two and a half as many women as men feeling "a great deal of apprehension" about negotiating (Babcock & Laschever 114). Her surveys found that "[w]omen felt particularly uneasy about scenarios involving work or activities in which they felt less expert than men (such as getting their car fixed). [...] Men associated words such as *exciting* and *fun* with negotiation far more than women, who were more likely...to choose words such as *scary*" (Babcock & Laschever 114). In describing their experience of negotiation, "women were more likely to pick metaphors such as *going to the dentist*" (Babcock & Laschever 114). Research has found that this increased anxiety makes it less likely that a woman will go through with a negotiation interaction (Babcock & Laschever 115).

Most of the apparent unfriendliness of FOSS projects is unintentional. However, to some degree it appears that these characteristics have become a point of pride for FOSS developers

and users.<sup>16</sup> The obscurity and difficulty associated with using or contributing to their software is taken as an indicator of high intelligence and hacking skills; in the worst cases, this unfriendliness becomes a form of hazing, where you prove that you belong in the FOSS community by demonstrating that you know how to deal with its most alienating elements.<sup>17</sup> After Vitorio Miliano wrote that closed-source/non-free software has more female contributors than FOSS because it has a friendlier culture, he received this reply: "So [women] have to be spoonfed? What makes them so goddamn important if they haven't proven they're willing to contribute?" As Miliano notes, this attitude is precisely the problem:

What you see as spoon-feeding is what normal men and women [...] see as proper, instructive socialization. "Hi, welcome to the community, here's a housewarming present, here's how things work, we could really use some help over here and I'd love to show you how to work on it[...]" When you volunteer your mortgage payments and your children's educational future and your commuting time to move into a neighborhood, the neighborhood welcomes you and shows you the ropes because now you're all in it together. With open source, you're also volunteering your time and knowledge and effort, but there's no welcoming committee, no backyard barbecues, just a bunch of unappreciative keeping-up-with-the-Jones', where if you don't have the internal motivation to contribute, you won't last. (Miliano)

Normal humans like to be welcomed. Recruitment processes for any number of other communities and collaborative efforts – university students, churchgoers, Habitat for Humanity volunteers – typically include elements to orient new members and help them become useful. Yet, for various reasons, such efforts are drastically undervalued and neglected by most FOSS projects. Feeling like an outsider – being the only woman, non-coder, or other minority member in the room, on the email list, or in the IRC channel – only exacerbates

---

16 The title of this paper came from a comment made by a longtime FOSS developer at PyCon 2010 to a woman who was listing factors that, to date, had made her hesitant to get involved in FOSS, as mentioned in this blog post: <http://jackdied.blogspot.com/2011/03/pycon-detritus.html>. While in this case the jibe appears to have had a positive, proactive effect (it got under the woman's skin, with the end result that she now contributes to FOSS) the attitude behind such comments is telling.

17 At PyCon 2011, I was asked half-jokingly, "Why can't these people [women, etc.] just read the docs and figure things out on their own? That's how I got involved!" Whether this mentality is simply reflective of a lack of understanding of how other people learn and participate or symptomatic of a sort of hazing ritual, this attitude seems fairly widespread.

unwelcoming or hostile infrastructure.

It is important to note that none of these assumptions or "filters" affects women or a particular ethnic group exclusively. There are plenty of people, men and women, who would otherwise be fantastic FOSS contributors who do not do so because they are busy, unsure of their technical prowess, shy, and/or discouraged by a project's seemingly-hostile atmosphere. It is also important to make clear that despite the dismal statistics, the FOSS community is not devoid of female contributors and project leaders; these "filters" are not absolute in their effects. My point is merely to show that women (and presumably other minority groups) tend to be disproportionately affected by these flaws in FOSS recruitment. On the flip side, this also means that women and minorities represent the greatest growth potential for outreach in the FOSS community; if FOSS were to go from 2 percent women to 50 percent women (or even just 20 percent women), that change would not merely make FOSS's demographic calculations more equitable but would also signal huge community growth as well, with thousands of new contributors coming "off the sidelines" (Laroia, "Get").

## **Conclusion**

Ultimately, most FOSS projects are not good at attracting new contributors of any sort. Sandeep Krishnamurthy's survey of active FOSS projects found that the median number of developers for a given project was just 4; the mode was one solitary developer (Krishnamurthy). With such a small contributor base it does not take much for a FOSS project to fail. Open source software veteran Karl Fogel estimates that as many as 95 percent of FOSS projects die on the vine (Fogel 12). Indeed, FOSS community members use the rather-morbid "bus factor" – how many key developers would have to get hit by a bus to make the project shut down – as a metric for the sustainability of a project; unfortunately, many projects, even

ones with an extensive user base, have bus factors of 1 or 2.

Recruitment of and outreach to new contributors are key to a FOSS project's long-term survival. However, at present the myriad obstacles to initial FOSS involvement mean that nearly everyone who does manage to join the community comes from the extreme, rarefied end of the bell curve in several dimensions – not just gender or ethnicity, but also things like personality and skill set.<sup>18</sup> Each incident or obstacle may on its own not be particularly troubling, but cumulatively they result in community monoculture and stagnation. "As the psychologist Virginia Valian writes in her book *Why So Slow: The Advancement of Women*: 'It is unfair to neglect even minor instances of group-based bias, because they add up to major inequalities'" (Babcock & Laschever 8). Without conscious effort to recognize and fix these molehills in FOSS recruitment and culture, they multiply to become mountains.

Fortunately, not all FOSS communities are unaware of these barriers and the "terror and frustration and vanishing" they cause (Laroia, "Get"). Dreamwidth, a fork of the code that runs blogging service LiveJournal, is notable as one of very few FOSS projects whose contributors are majority-female, with a substantial proportion of gay, trans, and genderqueer contributors as well. This is not an accident; as a project founded by women, Dreamwidth proactively prioritized inclusion and diversity from the very beginning. Its diversity statement, beginning simply with "We welcome you," is extraordinary by any measure and unique among FOSS projects.<sup>19</sup> Dreamwidth's "Getting Started" page for new developers states "If you're totally new to programming and/or contributing to Open Source projects, don't worry! We'll walk you through the process" – a friendly "Don't Panic"-esque notice that encourages even less-

---

<sup>18</sup> For instance, some commentators both in and outside the FOSS community suggest that participation in FOSS may select for people on the autistic spectrum, with various associated personality traits. However, to date there has been no medical or psychological work to confirm or refute such a link. See discussion in Krieger et al, p. 14-15.

<sup>19</sup> <http://www.dreamwidth.org/legal/diversity>

experienced coders to get involved.<sup>20</sup>

Another example of an unusually friendly FOSS project is ThinkUp, a social media-based brainstorming application headed by Gina Trapani and Anil Dash. In an interview, Dash explained why diversity was such a priority with ThinkUp:

It's really important to me that we have this open source project that's had remarkable early success and our lead coder is a woman, and she's out, and we have prominent developers in our community who are gay, who are women, who basically don't look like every other open source app out there. Essentially, that inclusiveness isn't just the goal of the crowdsourcing, but of the technology development itself -- because we all build our cultural assumptions into the tools we make. [Diversity] was a prerequisite. Absolutely essential. [...] It was more about what I saw as the failings of the tech industry where people would excitedly say, "You can get people to contribute to an open source app just by giving them t-shirts!" And the choice of t-shirts are men's medium or large. (Scola)

While diversity on gender, sexual orientation, and other typical socioeconomic dimensions was certainly a priority, the ThinkUp founders have explicitly stated that they see diversity in a more expansive way: "We encourage contributions not only by women, but open source newbies and non-usual suspects of all stripes: designers, user experience experts, writers, students, and enthusiastic users" (Trapani). ThinkUp does not have the prominent diversity statement that Dreamwidth does; its front page declares that the project "Welcom[es] All Developers!", but doesn't explicitly talk about sex, gender, race, etc.<sup>21</sup> However, it succeeds in recognizing that "it's the small things that count" in creating a welcoming environment (Stewart). For instance, the top of ThinkUp's code repository and developer website states, "**Welcome! You are a programmer!** The ThinkUp project eagerly welcomes new contributors from all communities, even if you don't think of yourself as a programmer. (Yet!)".<sup>22</sup> This statement immediately defuses potential contributors' self-doubt, uncertainty about their help

---

20 [http://wiki.dwscoalition.org/notes/Dev\\_Getting\\_Started](http://wiki.dwscoalition.org/notes/Dev_Getting_Started)

21 <http://thinkupapp.com/>

22 <https://github.com/ginatrapani/ThinkUp/wiki/Developer-Guide>

being wanted, and imposter syndrome symptoms, encouraging them to continue reading and get involved.

Defining a FOSS project's internal culture around friendly inclusion from the beginning is one way to avoid these problems; however, that does not give us much of a strategy for how to improve long-standing FOSS communities. Fortunately, a new generation of FOSS outreach programming has recently developed in two major FOSS communities and already such programs appear to be making huge differences in terms of both diversity and community growth generally. In 2009, two women in the San Francisco-area Ruby meetup group, unhappy with how few women attended the meetup, put on a free one-day workshop to teach "women and their friends" the basics of programming in Ruby on Rails, a popular language and platform for writing web applications (Mei).<sup>23</sup> That single workshop soon became a wildly-popular series of workshops, collectively called Railsbridge, hosted for free in the offices of San Francisco-based Web 2.0 magnates like Twitter and Pivotal Labs.<sup>24</sup> Consistently all sixty or so seats in a given workshop fill up within 24 hours of the event announcement. It turned out that there were hundreds of women, many already working in the tech industry in nontechnical positions, who had a thirst for learning to program but had not found such a friendly, welcoming environment in which to try. Many of these women continued their study of Ruby in book club-like meetings with other attendees as well as at hackathons hosted by the SF Ruby meetup group; a number have gone on to teach or assist with future Railsbridge workshops. The workshops and their aftermath did not just recruit new female members into

---

23 The "women and their friends" stipulation was meant to make it clear that the workshop was female-friendly without being exclusive, as well as to aid in recruitment: if you were a woman, you could attend automatically; if you were a man, you could attend, but only if you found a woman in your life who was interested in attending and would then bring you as her guest. This gave interested men an incentive to find a female friend or colleague who might be interested in learning to program and encourage them to attend the workshop. Roughly speaking, about a quarter to a third of Railsbridge attendees have been male.

24 More information on the workshops and related activism can be found at <http://workshops.railsbridge.org/>.

SF Ruby directly:

When the Ruby people brought in more women to the integrated community of their meetup group, they noticed that other women who didn't attend their workshops would start attending the meetup group. And what they were doing was undoing the self-reinforcing demographic problems. By having new people who represented a different demographic group, they would bring in their friends! (Laroia, "Get")

As a result, within a year the SF Ruby meetup group went from 2 percent female to 18 percent female (Mei). Other Ruby enthusiast groups are now attempting to clone the Railsbridge curriculum and run their own workshops across the country.

Railsbridge's efforts have also inspired enthusiasts of another open source programming language, Python, to do similar outreach. In February 2011, three members of the Boston Python meetup group, Asheesh Laroia, Jessica McKellar, and Ned Batchelder, organized a one-day workshop for women and their friends following a similar curriculum as Railsbridge, only teaching a Python-based set of technologies. The workshop was well attended and apparently successful (Laroia, "Get"). After Laroia spoke at PyCon (the main Python language conference) later that year about the workshop and FOSS outreach generally, attendees from several other Python user groups – in Minneapolis, the Bay Area, and Montreal, Canada – were inspired to run similar events, which have come to be known as PyStar workshops, in the following months.<sup>25</sup> Given how new PyStar is, it is not yet clear if it will have the same community-building, diversity-increasing effects in the Python community as Railsbridge had in the Ruby community, but so far the signs are promising.<sup>26</sup>

Vitorio Miliano believes that the problems with dominant modes of FOSS recruitment are inseparable from the problem of sexism in FOSS:

---

<sup>25</sup> More information on the workshops can be found at <http://pystar.org/>.

<sup>26</sup> Women's attendance of the Boston Python meetup jumped from about 2 percent (1 out of 50) in January, before the Boston Python workshop, to about 16 percent (11 out of 55) in May. <http://www.asheesh.org/note/community/boston-python-meetup.html>

I believe the problems with open source not being able to handle non-programmers in their projects is the same problem as the rampant sexism: open source culture is not feminist. Feminism is fundamentally about equality for *everyone*, not just women, and designers of any gender are just as alienated as women programmers, because it's not an equally welcoming environment. There's no perceived value in open source for mentoring, facilitation, disciplining of unruly users, training of newcomers or non-technical users, etc., which are needed to support both designers of any gender and women in any role. (Miliano)

Fortunately, the contrapositive of Miliano's argument appears to also be true: making FOSS more accessible and friendly in general appears to have the side effect of also making it more diverse. As FOSS-centered conference talks on outreach (such as Asheesh Laroia's PyCon lecture) and success stories from other projects' recruitment initiatives circulate, hopefully more FOSS projects will be turned on to the importance of being welcoming, concrete, and proactive in their handling of new contributors, "shy" or otherwise (Laroia, "Get"). Many members of the FOSS community have a certain allergy to the word "feminism" and are not especially motivated to address diversity issues for their own sake. As these new outreach ideas and practices spread, though, the incentive structure may be in place for FOSS projects to "accidentally" diversify their demographic makeup as they grow without having to grapple with that "f-word".

## Works Cited

Aurora, Valerie. "The Dark Side of Open Source Conferences." 1 Dec 2010.

<http://lwn.net/Articles/417952/> Accessed 13 Apr 2011.

Babcock, Linda and Sara Laschever, *Women Don't Ask: Negotiation and the Gender Divide*.

Princeton University Press. 2003.

Bolton, Nick. "How did you get involved with your open source community?" Stack Overflow. 5

Apr 2009. <http://stackoverflow.com/questions/718107/how-did-you-get-involved-with-your-open-source-community> Accessed 11 May 2011.

Buckland, Richard. "Lecture 1: Higher Computing." Lecture, University of New South Wales. 16

Mar 2008. <http://www.youtube.com/watch?v=hE7l6Adoiw>

Druckman, Katherine. "Angela Byron on Drupal 7." Linux Journal. 24 Feb 2011.

<http://www.linuxjournal.com/content/interview-angie-byron-drupal>

Dunnewind, Stephanie. "How can moms and dads achieve equality at home?" The Seattle

Times. 8 May 2004.

[http://seattletimes.nwsourc.com/html/living/2001922984\\_dadhel08.html](http://seattletimes.nwsourc.com/html/living/2001922984_dadhel08.html)

Fogel, Karl. *Producing Open Source Software*. O'Reilly Media. 2005.

Geek Feminism Wiki. "Impostor Syndrome."

[http://geekfeminism.wikia.com/wiki/Impostor\\_syndrome](http://geekfeminism.wikia.com/wiki/Impostor_syndrome) Accessed 4 May 2011.

Geek Feminism Wiki. "Incidents." <http://geekfeminism.wikia.com/wiki/Category:Incidents>

Accessed 16 Mar 2011.

Google. "Google Summer of Code." 2011. <http://code.google.com/soc/>

Ghosh, Rishab Aiyer et al. "Free/Libre and Open Source Software: Survey and Study."

International Institute of Infonomics. 2002.

<http://www.flossproject.org/report/index.htm>.

Halvorson, Heidi Grant. "The Trouble With Bright Girls." 1 Mar 2011.

[http://www.huffingtonpost.com/heidi-grant-halvorson-phd/girls-confidence\\_b\\_828418.html](http://www.huffingtonpost.com/heidi-grant-halvorson-phd/girls-confidence_b_828418.html)

Heeter, Carrie. "Girls game less because they have less free time, study." 24 Jul 2009.

<http://www.physorg.com/news167660240.html>

Hill, Catherine et al. "Why So Few? Women in Science, Technology, Engineering, and Mathematics." AAUW. 2002.

Hsu, Jean. "My Experiences as a Female Software Engineer." 17 Jan 2011.

<http://www.jeanhsu.com/2011/01/17/my-experiences-as-a-female-software-engineer/>

Kanazawa, Satoshi. "The Imposter Syndrome." Psychology Today. 26 July 2009.

<http://www.psychologytoday.com/blog/the-scientific-fundamentalist/200907/the-imposter-syndrome>

Kerner, Sean Michael. "Was Google's Summer of Code a Boon or Bust?" InternetNews.com. 22 May 2006. <http://www.internetnews.com/dev-news/article.php/3607711>

Kollock, Peter. "The Economies of Online Cooperation: Gifts and Public Goods in Cyberspace." Chapter 9 in *Communities in Cyberspace*, Kollock and Smith, eds. 1999.

Krieger, Bernhard, James Leach, and Dawn Nafus. "Free / Libre / Open Source Policy Support (FLOSSPOLs): Integrated Report of Findings (Gender Strand) ." Cambridge: University of Cambridge. 2006. <http://www.flosspols.org/>

Krishnamurthy, Sandeep. "Cave or Community? An Empirical Examination of 100 Mature Open Source Projects." First Monday. 3 Oct 2005.

<http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/1477/1392>

Laroia, Asheesh. "'Debian for Shy People': What's next." 10 Aug 2010.

<http://www.asheesh.org/note/debian/post-shy.html>

Laroia, Asheesh. "Diversity in Free Software: South Asians as an example." 18 Dec 2009.

<http://www.asheesh.org/note/debian/indians.html>

Laroia, Asheesh. "Get new contributors (and diversity) through outreach." Lecture, PyCon

2011. 11 Mar 2011. <http://pycon.blip.tv/file/4880711/>

Margolis, J., & Fisher, A. *Unlocking the clubhouse: Women in computing*. Cambridge:

Massachusetts Institute of Technology. 2002.

McPherson, Miller, Lynn Smith-Lovin, and James M. Cook. "Birds of a Feather: Homophily in

Social Networks." *Annu. Rev. Sociol.*, Vol. 27, p. 415-44. 2001.

Mei, Sarah. "Moving the Needle: How SF Ruby Got to 18%." Lecture, SCALE 8x. 19 Feb 2010.

<http://www.sarahmei.com/blog/2010/02/20/scale-8x-slides-posted/>

Miliano, Vittorio. "Designers and women in open source." 17 Mar 2011. [http://vi.to/designers-](http://vi.to/designers-and-women-in-open-source.html)

[and-women-in-open-source.html](http://vi.to/designers-and-women-in-open-source.html)

Noller, Jesse. "On Contribution." 5 May 2011. [http://jessenoller.com/2011/05/05/on-](http://jessenoller.com/2011/05/05/on-contribution/)

[contribution/](http://jessenoller.com/2011/05/05/on-contribution/)

Osterloh, Margit and Bruno S. Frey. "Motivation, Knowledge Transfer, and Organizational

Forms." *Organization Science*, Vol. 11, No. 5, p. 538-550. Sep - Oct 2000.

Restructure! "If you were hacking since age 8, it means you were privileged." 26 Jul 2010.

[http://restructure.wordpress.com/2010/07/26/if-you-were-hacking-since-age-8-it-](http://restructure.wordpress.com/2010/07/26/if-you-were-hacking-since-age-8-it-means-you-were-privileged/)

[means-you-were-privileged/](http://restructure.wordpress.com/2010/07/26/if-you-were-hacking-since-age-8-it-means-you-were-privileged/)

Roberts, Jeffrey A., Il-Horn Hann, and Sandra A. Slaughter. "Understanding the motivations,

participation, and performance of open source software developers: A longitudinal

study of the Apache projects." *Management Science*. 2006.

Scientist Carrie. "Why don't girls play video games? or, a fail blog." 26 Apr 2011.

<http://scientistcarrie.blogspot.com/2011/04/why-dont-girls-play-video-games-or->

[fail.html](#)

Scola, Nancy. "An Interview with Anil Dash, Director of Expert Labs." techPresident. 2 Apr 2010. <http://techpresident.com/blog-entry/interview-anil-dash-director-expert-labs>

Stewart, Rebecca. "It's the Small Things That Count." 13 May 2011.  
<http://blog.theleadingzero.com/?p=235>

Stross, Randall. "What Has Driven Women Out of Computer Science?" The New York Times. 15 Nov 2008. [http://www.nytimes.com/2008/11/16/business/16digi.html?\\_r=1](http://www.nytimes.com/2008/11/16/business/16digi.html?_r=1)

Trapani, Gina. "Designers, Women, and Hostility in Open Source." 23 Mar 2011.  
<http://smarterware.org/7550/designers-women-and-hostility-in-open-source>