



‘Patches don’t have gender’: What is not open in open source software

new media & society

14(4) 669–683

© The Author(s) 2011

Reprints and permission:

sagepub.co.uk/journalsPermissions.nav

DOI: 10.1177/1461444811422887

nms.sagepub.com



Dawn Nafus

Intel Labs, USA

Abstract

While open source software development promises a fairer, more democratic model of software production often compared to a gift economy, it also is far more male dominated than other forms of software production. The specific ways F/LOSS instantiates notions of openness in everyday practice exacerbates the exclusion of women. ‘Openness’ is a complex construct that affects more than intellectual property arrangements. It weaves together ideas about authorship, agency, and the circumstances under which knowledge and code can and cannot be exchanged. While open source developers believe technology is orthogonal to the social, notions of openness tie the social to the technical by separating persons from one another and relieving them of obligations that might be created in the course of other forms of gift exchange. In doing so, men monopolize code authorship and simultaneously de-legitimize the kinds of social ties necessary to build mechanisms for women’s inclusion.

Keywords

open source, F/LOSS, gender, knowledge economy

Introduction: Research problem and methods

Free/libre/open source software (F/LOSS)¹ developers are a loosely knit group of programmers who forgo traditional intellectual property rights in favor of what they see as better collaboration, knowledge exchange, and ultimately improved software technology. Early commentary from both scholarly and media sources has focused on F/LOSS as a moral response to capitalist economies (Kelty, 2004: 498) and as a progressive transformation in orthodox systems of capitalist production (Kelty, 2004; Lessig, 2000; Weber, 2004). According to Weber (2004: vii), ‘by experimenting with fundamental notions of what

Corresponding author:

Dawn Nafus, Intel Labs, JF-60, 2111 N.E. 25th Avenue, Hillsboro, OR 97124, USA

Email: dawn.nafus@intel.com

constituted property, this community has reframed and recast some of the most basic problems of governance.' While some have critiqued F/LOSS as a form of unpaid labor dependent on capitalist structures (Terranova, 2000), it primarily has attracted comparisons with alternative models of economic organization, such as gift economies (Raymond, 1998; Zeitlyn, 2003) and barter economies (Ghosh, 1998). Many F/LOSS contributors use these comparisons to evoke the promise of alternative, fairer, and more democratic social organization.

However, a scant 1.5 percent of F/LOSS participants are women (Ghosh et al., 2002), while women constitute 28 percent of people in technical roles in firms which rely on closed intellectual property arrangements (Evans et al., 2007; Ghosh et al., 2002). There is no evidence that women's participation level has changed from Ghosh's study. What has changed is the growth in large commercial enterprises that contributed to F/LOSS as part of paid labor. The present study saw no indication that this changed women's exclusion. Despite more women working in commercial organizations as a whole, it appeared that within these firms it was the men who were the participants in F/LOSS.

This low female participation rate is particularly surprising for a community in which anybody can participate. Both employees of firms and volunteers contribute to F/LOSS, unlike proprietary software firms that rely on paid employees. Therefore there are no formal barriers in terms of educational attainment, where women are often subtly but powerfully discouraged (Margolis and Fischer, 2002; Turkle 1986; Turkle and Papert, 1990). Nor are there the same rigid time commitment requirements that affect women's advancement in formal employment (Becker, 1993). In this sense we should expect higher rates of female participation in F/LOSS, not lower. Further, in daily F/LOSS activities, developers are constantly reminded about how increasing the number and diversity of minds focused on a problem produces better software. F/LOSS relies on consensus-building in smaller projects and, in larger projects, relatively flat formal organizational structures and charismatic leadership to make technical decisions. Inclusion of any and all interested parties is a central, constitutive part of F/LOSS social organization. Yet, F/LOSS is an extreme example of a much wider trend. Women's participation in technology production as a whole is going down, not up, in most industrialized countries (Evans et al., 2007). Kotamraju (2004) documented the masculinization of web design during the dot-com bust. Informalization of new media labor is disadvantaging women by rendering them more dependent on the whims of men in an 'old boys network' for consulting work (Gill, 2002).

Taken together, these trends supply ample reason to look at issues of gender and technology anew. In 2005–2007 a team of three anthropologists examined why F/LOSS had such a poor record of women's inclusion. Other team members have focused on gendered notions of freedom and aesthetics (Leach et al., 2009). Here I will show that the ways in which F/LOSS instantiates notions of openness in everyday practice exacerbate the exclusion. 'Openness' is a complex construct that refers to more than just intellectual property arrangements. It weaves together ideas about authorship, agency, the circumstances under which knowledge and code can and cannot be exchanged, and the social ties that are legitimate and illegitimate to make. Although this study cannot explain women's declining participation in all technology professions, to the extent that F/LOSS constitutes a significant shift in how information technologies are produced, the underlying issues that generate this exclusion might indeed be part of a wider trajectory in how technologies are created.

Before I discuss the study results, I would like to explain our methods. Primarily we conducted participant observation, carried out intermittently across two years in the United Kingdom, France, and Germany across multiple software projects large and small. We spoke with men and women, leaders and contributors, volunteers and those who conduct F/LOSS work as part of paid employment. This entailed becoming involved in the daily affairs of software projects, participating in informal meet-ups and formal conferences, observing and participating in online communication, and learning to code. We used a combination of semi-structured and open-ended interviews alongside participant observation. We focused our interviews on perceptions of gender, social organization, forms of formal and informal power, ontological understandings of software code, and motivations for contributing code. In addition, we conducted an online survey of 1500 F/LOSS contributors.

While the results of the quantitative work corroborate results of the ethnographic work, here I focus on the ethnographic findings as they reveal the conceptual framework of 'openness' with which men exclude women. First, I discuss the harsh treatment women face in F/LOSS, incorporating the literature on gender and socio-technical construction and identity. Next, I discuss the influence of 'moral liberalism' (Coleman and Golub, 2008) that has shaped the ways that gender is interpreted as artificial, and therefore 'fair game' as a resource for harassing, insulting, and excluding. In the final sections I show how this 'artificiality' is used as a point of contrast to buttress the realness of otherwise rather precarious code, and to curtail alternative forms of agency to which F/LOSS communities are decidedly not open.

Women's experiences of gender in F/LOSS

There is an extensive gender and technology literature that provides convincing reasons why women are less likely to be found in technology production. For example, Lin (2006) discovered that in F/LOSS communities programming environments were designed assuming a history with obsolete technologies which new entrants to the field did not have. Women tended to be newer to programming, and thus were excluded through design choices. As in previous studies (Gill and Grint, 1995; Green and Adam, 2001; Lie, 2003; Sorensen, 2002; Turkle, 1986, Turkle and Papert, 1990; Wajcman, 2000) women's exclusion is part of the broader problem of socio-technical construction, where both the material aspects of computing and the social identities that people create for themselves through engaging with programming are cultures made by and for men. This literature shows how programming cultures sustain certain forms of masculinity which make women concerned about being 'unfeminine' in their connection to technology or 'too feminine' by attracting unwanted male attention. Such tensions often result in women's shaken confidence in abilities, as well as creative ways of establishing agency and control (Geneve et al., 2008).

In the course of fieldwork, we found that notions of gender did include all the problems of identity and socio-technical construction that the gender and technology literature has robustly explored. Documentation work and other tasks seen as less technical were more likely to be done by women. Indeed, part of the different female participation rates in proprietary software can be accounted for by the fact that F/LOSS often disposes with this work. F/LOSS developers are less concerned whether anyone actually uses the software – an indifference that will become quite salient in this account.

The gender-technology relation, however complex and significant in its own right, is complicated in F/LOSS by other circumstances particular to its organization. Corneliussen (2009) has identified the false impression of stability in the gender and technology literature: if women are 'still' not in ICTs in equal numbers, this is falsely taken as indicative of an unchanging set of issues. Margolis and Fisher (2002) found that institutional changes in how undergraduate education is delivered can significantly affect women's participation, engagement and retention, including addressing in pedagogy the assumptions about previous use of obsolete technologies – indeed, the very same that Lin identifies for F/LOSS. Yet although F/LOSS has made radical institutional change, indeed rejecting traditional institutional forms entirely, women's participation rates have gone in precisely the other direction. Indeed, F/LOSS developers have rejected the very mechanisms through which Margolis and Fisher's university made change.

With these findings in mind, we explored which aspects of women's experiences are particular to F/LOSS. F/LOSS is organized in a way that intensifies and harshens forms of sexism women experience elsewhere. We found that women were sexualized, hurtful and offensive talk was openly defended, and women were obliged to remind men not to stare and point at them. From a female participant:

One IRC channel I used frequently made jokes about rapes ... At one stage [one particular joker I objected to] ended up telling me, if I couldn't 'stand the heat I should get out of the kitchen' and then I said, 'What would you say ... if I had been raped and I took exception to be used as a subject of humor?' and he said: 'that's too bad, but you need to learn to live with it.'

While sexist behavior is not unique to F/LOSS, here it is as constant as it is extreme. Many women went as far as hiding their gender by using gender-neutral names in online forums. They told us they could not find suitable mentors because upon discovering their mentee's gender, male mentors treated the relationship as a dating opportunity. Some told us that men would treat them as if they were their mother, asking for advice about how to dress and behave and then refusing to enter into a technical dialogue thereafter.

The chilling effects of this treatment are not hard to see. One woman described an incident that happened just before delivering a keynote speech at a community event. She was reviewing her notes just before going onstage when a stranger turned to her and told her that her way of giving a talk was not the right one. Obviously gawking at her wedding ring, and asking awkward questions about her husband and children, he explained that giving a talk is just like telling bedtime stories to children, surely a mode of communication with which she would be more comfortable. Whether or not it was deliberate, the effect was a flustered presentation and a disinterest in contributing to that community again. Similarly, another exasperated female participant complained in an interview: 'They just don't realize that making comments like that is just really, really frustrating.'

Gender-blind sexism

Both men and women do acknowledge there is a problem of some kind. In our survey, 75 percent of women and 65 percent of men believed that it is very true or mainly true that 'in online contexts women often get more attention as a woman rather than as a F/LOSS

participant.' Women's 'very true' responses were much higher than men's (32% compared to 16%). Yet few F/LOSS participants see themselves as sexist or hostile to women. There are near universal refrains within F/LOSS that gender ought to be irrelevant, even though there is an acknowledgement in survey responses that it is not irrelevant in practice. The acknowledgement of poor treatment of women in our survey does not begin to capture the rawness of the nerve that the issue touches for the community. For example, a sure way to begin a flame war – that is, a series of inflammatory messages posted on an internet discussion board – is to raise the issue of how to include more women. Word had spread quickly around message boards that our research was taking place, sparking irate tirades about how we were the sexist ones by even studying gender.

We were not the only ones accused of sexism. For many F/LOSS men, there was a notion that acknowledging gender at all was itself sexism. This discourse can be traced back to early texts read widely by the community. A F/LOSS founding father, Eric Raymond, asserted in a foundational text that 'after all, if one's imagination readily grants full human rights to future AI programs, robots, dolphins, and extraterrestrial aliens, mere color and gender can't seem very important any more' (Raymond, 2005). In practice, however, men and women approach the acknowledgement of gender in different ways. For many men, and some women, gender's 'unimportance' can be turned into a weapon. A male participant wrote in an online debate about a group that helps women contribute to one large software project:

I think the whole idea of 'Debian Women' is flawed. All it does is give / reiterate to people the idea that women are somehow different to men when it comes to computers and should be treated differently.

And from the same discussion ...

Instead of saying 'Linux geeks should be nicer to newbies,' [the article] says 'Guys should be nicer to women.' No ... To ask for different treatment for different genders is SEXIST, and when the stated goal is to minimize sexism, it becomes counter-productive.

In Raymond's rendition, gender is made to be 'mere' gender, and its lack of importance is taken to be a positive aspect of F/LOSS sociality. In our research, predominantly male developers interpreted the acknowledgement of gender as reiterating the artificial differences between the sexes. There are echoes here of a 'post-feminist' myth that equality has been won, and so further examination and questioning constitutes sexism.

Yet there is also a clear ontology that is specific to F/LOSS preoccupations. Gender's 'artificiality' constituted grounds for dismissal of its acknowledgement. Both men and women explained the lack of women in the field in terms of a 'misperception' that computers were for men, that there was nothing necessary and biological about the affinity between men and computers. In calling it a *misperception*, they seemed to suggest that computers already provided a kind of gender equality, and rejected the idea that a perception could be a social fact carrying ontological weight. The quote that I borrowed for the title of this paper is taken from an online F/LOSS discussion of whether to take gender into account in thinking about the future of the community. In it a male coder asserted that software patches (that is, lines of computer instructions that fix problems in previously

written software) did not have gender. The technological, in other words, is believed to be orthogonal to the social. In this discourse, the pressing need is not to consider such artificial constraints, but to make patches work, whatever their social origin. That is, patches not only do not have gender, they are also believed to be more real than gender.

Concerns are not, in fact, always about getting down to the business of software creation. Entire conferences are devoted to how to organize and maintain vibrant F/LOSS communities. This would not be possible if community members held a strict social/technical divide, as the patches comment suggests. It is primarily when it comes to gender that this 'pressing need' to get past the social suddenly emerges. For example, we held a lengthy discussion with a well-placed leader in one of the largest and most successful F/LOSS projects at a conference on building strong F/LOSS communities. He glowingly enthused about the international diversity of his members, and how diversity strengthened code quality. He observed that it was 'unfortunate' that only 2 percent of members in his organization were women. We raised a number of techniques that could be used to reach out to them, including building on the formal mentoring structures within the community that already helped newcomers, and recommendations from Margolis and Fischer (2002). His tone turned. He told us he did not *believe* in special help for women. He was genuinely concerned about the absence of women, and clearly valued diversity, but rejected any possible course of action. This was not because he believed our suggestions were likely to be ineffective; his obvious discomfort revealed that women's absence posed fewer problems than the method to change it.

Women in the community expressed similar sentiments. Some expressed embarrassment with being connected to inclusion efforts through their gender, such as support groups for women. Others took it as an occasion to assert that they were the only ones responsible for their own careers. In one firm that contributed to F/LOSS, it was women who conducted less prestigious debugging work, and men exclusively monopolized the more prestigious technical decision making. Yet all claimed no one had been discriminated against and that the women had chosen freely their line of work.

Successful inclusion strategies downplayed gender and made explicit that 'special help' was not desirable. Debian Women, a group of women and men gathered together to support and mentor programmers, made clear at every opportunity that their group was open to male participation. Although they did not treat the acknowledgement of gender as sexism, they struggled with how to acknowledge gender in an environment where others did. The gendered name was constructed as a 'mere' acknowledgement that women tended to be newcomers, or else 'happen' to not enjoy the aggressive tone of online conversation. They argued that men too did not appreciate the harsh tone and constant personal attacks. By downplaying who their target audience was, while acknowledging that it is women who are hit hardest, the group focused on helping newcomers, which most did not question as a legitimate activity. Nevertheless, many remained unconvinced of its legitimacy.

What liberalism does for F/LOSS

Constructing gender as artificial serves as a contrast to the 'realness' of technology like software patches, but it also draws on additional discourses outside of technology. It echoes longer-standing ideas about liberal feminism, premised on the idea that equality would be

had once formal equality of opportunity was achieved and choices were widened. Like F/LOSS developers, this tradition treats equality in terms of minimizing gender differences such that gender might no longer matter, and approaches technology as a neutral tool which men monopolize (Schiebinger, 2001). While this discourse has proven controversial within feminist scholarship, in F/LOSS it is taken as given that liberal feminism is what equality means. This is part of a wider discursive turn after the civil rights movement and second wave feminism that made it socially unacceptable to use physical characteristics as a basis for creating inequality. As both Nakamura (2009) and Bonilla-Silva (2006) show, substituting culture for 'biological' race exacerbates inequality. In Nakamura's work, computer gamers who exclude, harass, and insult Chinese game players believe that 'the problem ... isn't that they are Chinese, it is that they "*act Chinese*"' (Nakamura, 2009: 139). Gamers rendered Chinese-ness into an individual choice and artificiality where Others are to blame for their own marginality. Similarly, Bonilla-Silva's study of 'color-blind racism' shows how long-standing liberal notions of meritocracy are used in white discourse to exculpate whites from systematic inequality while carefully avoiding reference to physical characteristics. These liberal discourses make it an apparent mystery why African-Americans are disproportionately poor. F/LOSS men and some women similarly used their professed blindness to gender to conduct gender-based exclusion while not thinking of themselves as sexist because they carefully avoided reference to unavoidable physical characteristics.

Gill (2002) asks why a discourse which makes gender more visible is not deployed by a majority of women in new media production. In F/LOSS, making gender visible is difficult. Advocacy groups for women's inclusion, such as Debian Women and Linux Chix, claimed that there is a very real culture hostile to women, and it is not their 'mis-perception.' It is precisely here where the difficulty starts. To have to argue that gender is both something utterly artificial and meaningless and yet very real put these advocates in an interpretive bind regarding what is and is not real. This is a central, vexing bind critical to F/LOSS production.

While F/LOSS draws on elements from Western liberal traditions to conduct gender-based exclusion, it does so in very particular ways that not only exacerbate the exclusion of women, but simultaneously inform practices of code sharing and notions of authorship, and indeed call into question the widespread comparison with gift economies. Coleman and Golub (2008) have described hacker morality in general, and F/LOSS specifically, as based around a moral liberalism. They argue that much of the F/LOSS cultural ethos relies on early notions of liberty formulated by John Stuart Mill (2008/1859) and other classical Western liberal thinkers who posit that freedom lies in individuality being as unfettered by others as possible. They show how talk about F/LOSS, either in terms of community making, intellectual property, or ethics are grounded in a liberal intellectual tradition – drawing on notions of free speech, meritocracy, privacy, the power of the individual, self-cultivation of knowledge, etc.

In practice, this means female achievement was unthreatening as long as it could be construed as a matter of individual choice. Few were prepared to question whether it was possible for an individual woman, through heroic efforts, to become successful in F/LOSS. In the previously mentioned firm, the work space was literally divided down the middle along gender lines. The more technical developers, all male, sat on one side of the room and the staff who did the documentation and testing work, sat on the other. Yet all fifteen

employees insisted this was mere coincidence, and that no one had been ‘discriminated against’ (their words). The women insisted that it was their individual choice to work in support rather than development, and that the office geography therefore constituted an accident of multiple individual decisions. To them, a power-free invisible hand had made this possible.

In this firm, workers are neither ignorant nor foolish; rather, their beliefs enabled them to retain the possibility that they *could* have done the developers’ job if they chose to do so. Their individuated notion of agency, centered on ideas about choice, left them unwilling to acknowledge the materially obvious ways in which their participation had been socially shaped. F/LOSS participants instead kept intact the possibility that through sheer individual zeal, nearly all paths are open. The notion that there is something systemic about the exclusion of women troubles the powerful sense of agency these women experience through beliefs about choice.

While choice is constitutive of agency for many women, for a few the skillful deployment of liberal ideals allows them to exclude competitors who might have different notions of agency. For example, one leader publicly called herself an ‘open source diva.’ The label did two important things. It rendered gender an eccentricity, in the sense that ‘diva’ – a term loaded with gender connotations – was a rare acknowledgement that she even had a gender at all. Yet it was immediately made absurd by the unseriousness of the job title. It also individuated her: try being another diva with one already in circulation! This strategy works because it made it difficult for her audience to see how her performance of gender might have come from somewhere other than her personal inventiveness. In turn, she attacked those who acknowledge gender as a problem for being ‘whiners’ and not ‘doers.’ By echoing the pressing need to go back to doing technical work (i.e. be ‘doers’), she again diminishes the realness of gender, suggesting it is only talk rather than real action. By making ‘whiners’ of those who exercise agency by seeking solidarity with others and building organizational resources, rather than rendering themselves quirky individuals, she makes any alternative form of social organization impossible while positioning herself as a sufficiently individuated productive person.

While Coleman and Golub show the shifting ways hackers deploy liberal ideas, the construct of gender as an artificial veneer somehow separable from individuals, and therefore fair game for discriminatory practices, suggests that some aspects of liberal ideas are not terribly multivalent. Opening intellectual property, maintaining a strong contributor base, debating how code is agreed to be incorporated into the larger project: all these activities can be negotiated as a normal part of F/LOSS activity to the extent that they support the goal of making ‘better’ software. But they cannot threaten the real business of supposedly meritocratic technology production. Raising questions about exclusion and power, and entertaining solutions deployed by other technical organizations, does so because they suggest that merit may be more than a matter of individual worth.

From meritocracy to pushyocracy

F/LOSS members understand technology as having its own moral imperative. Above all, code must evolve and progress. To participants, this does not require any further justification. Particularly valued, evolved, pieces of code are called ‘Good Things.’ A Good Thing is written with capitals and pronounced with corresponding moral emphasis. Good Things

are believed to be as true as they are beautiful, again reflecting a reliance on long-standing liberal intellectual heritage.

The uncontested way to expand knowledge boundaries within F/LOSS is by working openly and sharing the code. Participants believe that code, like a Popperian falsifiable hypothesis, ought to demonstrate its worth only through the scrutiny of others. Linus Torvalds, the leader of the community of developers that built the Linux operating system, is often quoted as saying that with enough eyeballs, all bugs are shallow (called 'Linus' Law'). Other researchers have likened software production to craft production² (Coleman, 2001) or pleasurable play (Klief and Faulkner, 2002). In our study this is indeed part of coders' imagination. Members often describe their work as 'scratching an itch' by producing something tangible and craft-like. Scratching an itch is a common reason why people become involved with communities and why they stay. Yet tradition and repetition, key elements of other forms of craft production, have no place other than as building blocks upon which to take one's own work further. Re-doing work similar to that of other coders does not scratch the itch satisfactorily, whereas it generally does among craftspeople. In this way, the craft system looks suspiciously like a system of science. It is not considered interesting to just make a media player or word processor, but only *new kinds* of media players or word processors that exemplify some transformation in knowledge. Through the emphasis on subjecting new code to public scrutiny, F/LOSS programmers put themselves on a technological, generative, and creative frontier: technology's edge (Helmreich, 1998).

Like a system of science, novel code circulates more readily in F/LOSS than that considered repetitive or traditional. New programming languages similarly proliferate at a rate that confounds everyone involved. No one can claim absolute expertise (Born, 1996; Downey, 1998; Ullman, 1997). We experienced exactly this problem in learning to participate in the F/LOSS community in Paris. What at first was a trade-off between learning the native language of the field and getting on with the ethnography subsequently turned out to be a never-ending spiral of new technical forms of which community members were themselves challenged to keep on top.

With unknowns all around, and an ethos concerned with making better code, it becomes necessary to establish which code pushes the boundaries and which is mere replication. Establishing code as something that pushes the edge becomes a source of lasting legitimacy and reputation. To achieve this kind of legitimacy, participants say the code should be able to speak for itself. That is, the technology itself ought to be the authority, just as the 'patches don't have gender' claim refers authority back to the software. The liberal notion of gender as irrelevant works nicely in a context where technical skill ought to be evident in the work itself. One contributor reported an instance where a developer was not able to convince his colleagues about the worth of his code due to his limited English language skills, but believed success was possible because the code could speak for itself. Code can speak for itself to the extent that it 'works;' that is, it does have some discernable material effect independent of the coder pressing buttons, and presumably independently of that coder's gender. That effect is in the form of machine operations that bring something to life on the screen, or other people's screens. With code that significantly pushes boundaries, however, it is less clear that code can speak for itself through the mere fact that it functions successfully. Such 'edgy' code is meant to be unstable, unfunctioning even. The new knowledge embedded in code is believed to be more valuable than the software's usefulness to an end user.

These notions of code oblige other coders to go digging through lines of text within the source code to identify how it works, and thus uncover knowledge and techniques that might be brought to other coding problems. In principle, how the code works is right there, embodied in the lines of code which have been made open to other programmers. It is incumbent upon the reader of lines of code to spot it, to look for 'how it works' in the code and not in the gendered relations that organized who was in a position to make technical choices. The knowledge is contained within code as a subtext. It has no direct effects in the sense of making a machine act without the code's creator, but it nevertheless is key to make new possibilities come forth. Code is simultaneously the science experiment and the account of it submitted for peer review. Successful code is supposed to operate without interventions of its makers. In doing so it ought to reveal something new, laying possibilities for technology's edge to shift. In this way, developers grant code its own agency.

The fact that human interpretation made lines of code discernable in practice was not something F/LOSS participants cared to spend a good deal of time considering. Participants granted the technology itself the ability to push the boundaries, not people reading the code. Yet if technology is meant to speak for itself on one hand, and yet be so advanced as to be unable to do so in practice, questions of who or what pushes technology forward become problematic, divisive and contested. This logic does not leave much room for what Turkle and Papert call epistemological pluralism (1990), that is, an acknowledgement that there are multiple ways of solving problems. They describe a situation where among school-children, boys tend to develop styles of programming that rely on master plans, black boxing the details and treating them as autonomous building blocks. Girls, both school-children and college-age computer science majors, wanted to know what was in the insides of these black boxes. They wanted to figure out how all the parts worked – in a sense getting close to the machine in the way that F/LOSS coders otherwise do value. As a result, they innovated in unexpected ways by building on pieces that go unseen by male students; yet, within education systems, they are told that their approaches are wrong. In F/LOSS there is a great deal of consternation about what is 'good' and 'bad' code, and less of a sense that there could be equally valid methods. Women's advocates found this problematic even if they were careful not to attribute differences in coding style to gender.

Epistemological intolerance is not simple closed-mindedness; it is necessary to creating certainty in a highly unstable, uncertain environment where it is unclear whether one's activities are 'real' or not. F/LOSS communities have two key ways of mitigating against uncertainty, enabling people to make claims that they are indeed on technology's edge: emphasizing the truth status of code and externalizing code. Both of these are called into question in discussions about gender. To take the first, truth claims are central in debates about the worth of software. Truth statements ('if 'x' is true, then y') are central in programming languages. The software functions, and thus can act upon the world, if the truth statements are true. Making software 'work' is a zero-sum situation; it either runs or fails to run, and members describe with exhilaration moments of finally making something work. Judgments are made about 'good code' and 'bad code' as if all good code works and all bad code does not. Yet knowledge-pushing code rarely works as a stable, usable system. Nevertheless, the binary truthfulness of working code sets the tone of online discussion. A person is not likely to find a way to the edge if the code he or she writes never works. Conversely, contributors can point to their own working code as the only true method, de-emphasizing the multiple ways that someone could get something to work.

There is a telling but necessary slippage between the technology and its creator. Although it is considered ideal that good code would speak for itself, in reality authors must vociferously defend their proposed code in order to demonstrate knowledge and establish what good coding is: that is, they must create the truthfulness of code, and thus technology's edge, through highly masculinized, aggressive online talking. This is especially necessary the more edgy the code is, as there is always more uncertainty there. In this context, the practice of re-using code, and thus the reason for it to be openly readable in the first place, is not necessarily about developing shared understanding of underlying principles behind them as claimed in Popperian notion of openness that underwrites Linus' Law. We might expect that a Popperian model of open knowledge production would value the documentation of code to facilitate shared understandings, and push the knowledge-making community along further as a whole, but here documentation work is not highly valued. What is more important is the way that code cultivates individual agency for its authors and others which, in the end, only legitimately can be expressed through yet more code.

The itch that developers said they scratch is precisely the moment knowledge comes to be recognized as self-generated, and in personal narratives coders constructed these moments as profoundly transformational – people talked about being able to say 'I did that' as the moment they become a coder. Many of our research participants began reading computer manuals and books from a very young age. Yet they recognized this only as an independent autogeneration of knowledge between person and machine, not a communication between author and reader. In this way, code not only crystallized achievement, and conferred technical leadership, but produced a transformative knowledge of self.

Despite the belief that code reveals knowledge, participation in discussion can be even more important than actual lines of code. We found cases where people who do contribute a great deal of code, but do not get involved in flame wars about the worth of code, are perceived as less knowledgeable. If code embodies knowledge as a kind of subtext, flaming asserts that subtext in no uncertain terms. While online communication is known to lend itself to flaming (Winter and Huff, 1996), it appears that in F/LOSS it is particularly rewarded. For example, one of our participants, known for frequent flaming, explained his behavior as a way to elicit objective truth. He considered that if someone is wrong, not only should they should be told so, but the point should be insisted upon until they accept it, in much the same way that code can be made to work with enough debugging. The important thing was for truth to be established, regardless of how disruptive or anti-social it may be. Indeed, flames were carried out until one person backed down.

Acrimonious talk was a prime reason women, as well as some men, said they leave F/LOSS. In their leaving, they often cited a central community value: that communities insufficiently get down to the real business of technical work, and instead rely on flaming to make decisions about which piece of code is worthy of becoming part of the larger software project. Critics call it a 'pushyocracy' as opposed to a meritocracy. This accusation is undermining to the notion that better code is being produced through open scrutiny. It suggests that merits are not being found in a purely external plane revealed by whether or not the code works.

While the (constructed) truth status of code secured a person's place at the edge, so did the system of code circulation. Similar to the way truth statements operate, the code either got incorporated into the community's software, or it did not. Incorporation revealed whether leadership has occurred. The thing-like transaction, as opposed to a discursive

co-creation of knowledge, supported the view that authors of code are right or wrong. This model of exchange individuated both producer and recipient. In the gift economies to which F/LOSS is often compared, gifts are given to create ties and obligations among people. But in F/LOSS, code was just as readily given away as a way of cutting ties. Openness was treated as a way of saying 'it's yours now, you do what you like.' Our research participants did not necessarily know, or care, who the receiver of their code was. Whether the code or the knowledge embedded in it gets taken up by subsequent authors is anyone's guess. Any other communication with users of the code, such as manuals that enable others to read and interpret it, did not really count as a part of making code free. This is work that women often do.

When conversation turned sour, the lack of connection between giver and receiver was wielded as a rhetorical weapon. A common way to construct oneself as more knowledgeable than others was to tell people who are asking questions about how some aspect of the code works to 'RTFM' (read the f***ing manual). Few felt obliged to write one, however. That is, there were no assumed obligations or ties upon which question-askers might draw if authority only came from the seeming merits of this or that code, embedded in the code itself which the receiver already had. Community members are decidedly *not* free to build ties that might oblige others to explain themselves, which is exactly what women's groups do, and exactly why they are considered problematic. If someone does not like being in one software project, the accepted course of action is to simply start another project elsewhere, not create an obligation for that community to include you. This particular form of exchange means that others can push the technology along further only *as* individually willful agents who have taken it upon themselves to 'read the f***ing manual.' Freed of interpersonal stakes, code authors do not have to be certain of what happens once the code is transferred: the satisfaction is in the fact that it works, and that it was 'really' launched out into the ether. Whether someone builds on it, or whether something like the promised innovation actually happens, is only a diffuse, future concern. Gender 'blindness' is a blindness that is more than a techie aversion to the social. It is also a liberal denial of ties that can create obligations in others.

Conclusion: Openness as RTFM

While Western liberal notions of freedom and individual self-sufficiency are resources for F/LOSS members to think about themselves and their actions, it is the gendered practice of 'free' exchange that gives the liberal principles their practical effect, and makes them far less multivalent than in Coleman and Golub's account. Here only one kind of knowledge is acceptable: externalized and truthful knowledge. That good code, in practice, has to be asserted and circulated by extra-technological means troubles this epistemological monoculture. The act of flaming can be acknowledged, indeed glorified, but is misrecognized as merely helping along technology's own progress in a way that is seemingly orthogonal to all things social. Addressing the exclusion of women is troubling because it upsets this necessary misrecognition. Women's flight from F/LOSS cannot matter because if it did, participants would have to confront their creations as made by social force rather than revealed ontological fact.

The cost of this acknowledgement would have to be paid in three ways: in people's reputations, in notions of the self and its relation to technology, and in notions of freedom. To take the first, being explicit about the work of interpretation in F/LOSS means that people risk losing a key way of establishing reputation: referrals to a technology-given truth. If reputations are to be preserved as having technical integrity, technical progress cannot be just an 'artificial' outcome of verbal prowess. A person cannot verbally beat the other into submission if an externally given truth does not give them the right (and rightness). The temporary deferral of agency to the technology itself is broken if women are to be deliberately included.

Inclusion similarly troubles ideas about the self and agency. Recall that communities value the notion that anyone can start tinkering with bug fixes. If gender is more than just arbitrary artificiality to be overcome through individual volition, it suggests there are in fact attachments and ways of doing things that were not necessarily chosen, as in the firm with the gender-divided room. Perhaps if gender were considered real and not an 'artificial' difference between men and women, maybe people would not, after all, just RTFM on their own, but establish the truth among themselves. Acknowledging this, as women's groups obliquely do, reveals the precariousness of the figure-it-out-for-yourself story that matters significantly to both men and women. Would the 'itch' to create be so compelling without the do-it-yourself story?

Finally, social ties evoke a closed system from which developers feel they have broken free through the very way technology circulates. 'Open' is believed to mean that technology circulates on its own accord, set free to have its own impacts on the world. In order to open the doors to 'anyone,' the possibility that there are social loops creating knowledge and passing it along – the very mechanism that both excludes women and could serve to include them – threatens the basis on which it is possible to claim that the door is open. That 'free' technology is not in fact freed of social pattern implies that such social unfetteredness did not in fact take place. Men and some women protect this very specific form of openness, even if it means closing the doors to those who do not fit, and even if it means establishing an epistemological monoculture, because this specific form of openness makes F/LOSS innovative and distinct from other forms of software production.

In F/LOSS, openness relies on a steadfastly closed epistemological frame that not only constitutes technology as apart from persons, but shapes this separability in such a way that code is more than just outside the realm of the social: it is downright *freed* from it. The social here is not exactly orthogonal to the technical as F/LOSS imagination has it; rather, social forms shape how ties are *severed*, as well as how they are built, between people. Not needing to know with whom code is being exchanged, or having a stake in their concerns, is as central to F/LOSS as open scrutiny to improve code quality. What this study shows is that claims of openness must be interrogated not just for the kinds of collaborations and new combinations they enable, but also in terms of what they proscribe.

Acknowledgments

The author would like to thank James Leach, Bernhard Kreiger, Lisa Nakamura, Dan Jaffee, Hannah Wallach and anonymous reviewers for their contributions. Bill Graves offered particularly generous careful reading and insightful comments that made this work possible. This work was supported by the European Commission FP6 (FP6-IST- 507524).

Notes

1. Open source is a more common term in the press and in commercial circles, but 'free' or 'libre' are also important terms for community members, hence I use the inclusive form, F/LOSS.
2. Here F/LOSSers are drawing on the masculinity of the eighteenth century blacksmith or wood-worker, an image that evokes a self-sufficient individualist without posing a threat to male dominance.

References

- Becker G (1993) *A Treatise on the Family*. Cambridge, MA: Harvard University Press.
- Bonilla-Silva E (2006) *Racism without Racists: Color-blind Racism and the Persistence of Racial Inequality in the United States*. New York: Rowman & Littlefield.
- Born G (1996) Immateriality and sociality: The dynamics of intellectual property in a computer software research culture. *Social Anthropology* 4(2): 101–116.
- Coleman G (2001) High-tech guilds in the era of global capital. *Anthropology of Work Review* 22: 28–32.
- Coleman G and Golub A (2008) Hacker practice: Moral genres and the cultural articulation of liberalism. *Anthropological Theory* 8: 255.
- Corneliussen H (2009) Disrupting the impression of stability in the gender-technology relation. Fifth European Symposium on Gender & ICT. Bremen, Germany, March.
- Downey GL (1998) *The Machine in Me: An Anthropologist Sits among Computer Engineers*. London: Routledge.
- Evans C, Glover J, Guerrier Y and Wilson C (2007) *Implementing Diversity Policies: Guiding Principles*. London: The Royal Academy of Engineering.
- Geneve A, Nelson K and Christie R (2008) Passion, women and the games industry: Influences on women's participation in the Australian digital content industry. Paper presented at Women in Games, Warwick, UK, 10–12th September 2008.
- Ghosh R (1998) Cooking pot markets: An economic model for the trade in free goods and services on the internet. *First Monday* 3(3). Available at: <http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/580> (accessed 27 September 2011).
- Ghosh R, Glott A, Krieger B, and Robles B (2002) *Free/Libre and Open Source Software: Survey and Study. Part IV: Survey of Developers*. Maastricht: International Institute of Infonomics /Merit.
- Gill R (2002) Cool, creative and egalitarian? Exploring gender in project-based new media work in Europe. *Information, Communication, and Society* 5(1): 70–89.
- Gill R and Grint K (1995) *The Gender–Technology Relation: Contemporary Theory and Research*. London: Taylor & Francis.
- Green E and Adam A (2001) *Virtual Gender: Technology, Consumption and Identity*. London: Routledge.
- Helmreich S (1998) *Silicon Second Nature*. Berkeley: University of California Press.
- Kelly C (2004) Culture's open sources: Software, copyright, and cultural critique. *Anthropological Quarterly* 77(3): 499–506.
- Klief T and Faulkner W (2002) *Boys and their Toys: Men's Pleasures in Technology*. Available at: www.rcss.ed.ac.uk/sigis/public/backgrounddocs/BOYS_AND_THEIR_TOYS-ZIF8.rtf (consulted June 2005).
- Kotamraju N (2004) Art versus code: The dendered evolution of web design skills. In Howard P and Jones S (eds) *Society Online*. London: Sage.
- Leach J, with Nafus D and Krieger B (2009) Freedom imagined: Morality and aesthetics in open source software design. *Ethnos* 74(1): 51–77.
- Lessig L (2000) *Code and Other Laws of Cyberspace*. New York: Basic Books.

- Lie M (2003) *He, She and IT Revisited: New Perspectives on Gender in the Information Society*. Oslo: Gyldendal Akademisk.
- Lin Yuwei (2006) A techno-feminist view on open source software development. In Trauth E (ed.) *Encyclopedia of Gender and Information Technology*. Hershey, PA: Idea Group Publishing.
- Margolis J and Fisher A (2002) *Unlocking the Clubhouse: Women in Computing*. Cambridge, MA: MIT Press.
- Mill JS (2008/1859) *On Liberty*. Forgotten Books. www.forgottenbooks.org.
- Nakamura L (2009) Don't hate the player, hate the game: The racialization of labor in World of Warcraft. *Critical Studies of Communication* 26(2): 128–144.
- Raymond E (1998) Homesteading on the Noosphere. *First Monday* 3(10). Available at: http://www.firstmonday.org/issues/issue3_10/raymond/index.html (accessed 27 September 2011).
- Raymond E (2005) *A Portrait of J. Random Hacker*. Available at: <http://www.retrologic.com/jargon/appendixb.html> (consulted June 2005).
- Schiebinger L (2001) *Has Feminism Changed Science?* Cambridge, MA: Harvard University Press.
- Sorensen K (2002) *Love, Duty and the S-curve: An Overview of Some Current Literature on Gender and ICT*. Available at: www.rcss.ed.ac.uk/sigis/public/documents/SIGIS_D02_Part1.pdf (consulted June 2005).
- Terranova T (2000) Free labor: Producing culture for the digital economy. *Social Text* 63(2): 33–58.
- Turkle S (1986) Computational reticence: Why women fear the intimate machine. In Kramarae C (ed.) *Technology and Women's Voices*. New York: Pergamon Press.
- Turkle S and Papert S (1990) Epistemological pluralism: Styles and voices within the computer culture. *Signs* 16:128–157.
- Ullman E (1997) *Close to the Machine: Technophilia and its Discontents*. San Francisco, CA: City Lights Books.
- Wajcman J (2000) Reflections on gender and technology studies: In what state is the Art? *Social Studies of Science* 30(3): 447–464.
- Weber S (2004) *The Success of Open Source*. Cambridge, MA: Harvard University Press.
- Winter D and Huff C (1996) Adapting the internet: Comments from a women-only electronic forum. *The American Sociologist* 27(1): 30–54.
- Zeitlyn D (2003) Gift economies in the development of Open Source software: Anthropological reflections. *Research Policy* 32(7): 1287–1291.

Dawn Nafus is an anthropologist at Intel where she conducts research on technology use in order to shape new products and strategies.