AN ABSTRACT OF THE THESIS OF

J. Victor Kuechler for the degree of Master of Arts in Interdisciplinary Studies in Computer Science, Speech Communication, and English presented on June 5, 2013.
Title: The Emergent Qualities of Diversity in Free and Open Source Software Communities: A Critical Review and Theoretical Discussion

Abstract approved:

_____

Carlos Jensen

Free and open source software (FOSS) projects primarily rely on the efforts of volunteer contributors from around the world. For this reason, recruiting and retaining contributor is vital to the sustainability and growth of FOSS projects. This notion became the jumping-off point for this three-part investigation into the cultural structure and social dynamics of the FOSS community. In Chapter 2, we analyzed mailing list discussions initiated by newcomers to a FOSS project and found that receiving timely and supportive responses was positively correlated to newcomers' future participation. In Chapter 3, we examined mailing list subscription logs, and found a disproportional attrition rate among women along every step of the FOSS joining process, further documentation of a well-known lack of gender diversity in FOSS. Finally, in Chapter 4, we examined the current demographic composition of the FOSS community, and the lack of diversity in a more general sense, as well as the mechanisms that perpetuate this situation. We present two theoretical

frameworks—group faultlines theory and critical systems thinking, that can help explain this current homogeneity, as well as guide future research.

The Emergent Qualities of Diversity in Free and Open Source Software Communities: A Critical Review and Theoretical Discussion

by
J. Victor Kuechler

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Arts in Interdisciplinary Studies

Presented June 5, 2013
Commencement June 2013

Master of Arts in Interdisciplinary Studies thesis of J. Victor Kuechler presented on June 5, 2013.

APPROVED:

_____

Major Professor, representing Computer Science

_____

Director of Interdisciplinary Studies Program

_____

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

_____

J. Victor Kuechler, Author

ACKNOWLEDGMENTS

The creation of this thesis and my ability to negotiate, process, and understand the many complex issues herein, have undoubtedly been shaped and "coerced" by a community and cohort of individuals. First and foremost, I would like to thank Dr. Carlos Jensen for hiring an English major in 2009 to write software documentation, introducing him to Free and Open Source Software, funding his graduate program, and strategically inspiring him to create this compendium of academic literature. I am indebted to Steve Kunert for his mentorship and friendship over the last seven years. I strongly believe that his guidance led me to pursue the technological side of writing. I am indebted to Dr. Gerald Voorhees for helping me understand the virtual community and its nuanced norms, behaviors, and values, as well as his readiness and willingness to participate in my committee, having just joined the university. I am also grateful to Dr. Paul Kopperman for acting as my GCR. Countless thanks are due to my fellow compatriots in the Human-Computer Interaction Group. Without your feedback, friendship, and support, I would not have made it this far. I specifically want to thank Jennifer Davidson for sharing her passion for FOSS and diversity with me, and I applaud her ability to handle my chronic loquaciousness. Although I can't name them all, I would also like to hypothetically high five Darren, Jose, Jordan, Carrie, Vignesh, Ben, and Soroush for being exceptional colleagues.

I finally would like to thank a special lady who decided I was worthy enough to be her husband. Thank you, Abby, for believing in me and supporting me throughout this graduate program. I do this for you, because "I can do all things through Him who strengthens me." –Philippians 4:13

CONTRIBUTION OF AUTHORS

Dr. Carlos Jensen is the primary author of the first manuscript (Chapter 2), while Scott King conducted the original data analysis with the help of the HCI group. I provided a significant amount of research and writing to the paper.

I am the primary author of the second manuscript (Chapter 3). Claire Gilbertson conducted the original data analysis as part of her Master's thesis. I built upon her work and published this conference paper with the help of Dr. Jensen. Although this published paper is based on data from Claire's thesis, it is a significant rewrite and formulation from her original work.

I am the primary author of the third manuscript (Chapter 4).

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

Chapter 1. Introduction to Body of Work

The manuscripts collected in this thesis are both a chronology of publication and an evolving pursuit of understanding participation in free and open source software (FOSS) communities. In Chapter 2, we explored the impact of community responses to first-time mailing list posters in FOSS. We found that positive feedback from the community is positively correlated with future participation in the community. We also discovered that an astonishingly low number of women post on FOSS mailing lists.

Since we know that FOSS is a male-dominated community, we next sought to understand if there were gender differences in mailing list participation. To answer this question, we analyzed mailing list subscription logs in seven FOSS projects hosted at OSU's Open Source Lab. We found that mailing list subscription logs of women were disproportionately lower than man, and we proposed that somewhere in the early joining process women are leaving the community. We know that gender has become a hot-button issue in FOSS over the last fifteen years due to a variety of politicized events and discussions within the community. While these events are surely impactful, we still don't have information on why women leave the community.

After scouring through literature on gender conflict in FOSS, we realized that the FOSS community has sporadically discussed conflicts related to diversity, but usually in response to a recent, salient diversity conflict. To our knowledge though, no one has presented a theoretical framework to formally address the roots of diversity conflict. We address this in Chapter 4, and conduct a theoretical discussion of the current demographic composition of the FOSS community, as well as present two theoretical approaches to understand

diversity conflict: group faultlines theory and critical systems thinking. These approaches are important because they provide us with two methods to further understand how the cultural structure of FOSS can explain the nature of FOSS conflict.

Chapter 2. Joining Free/Open Source Software Communities: An Analysis of Newbies' First Interactions on Project Mailing Lists

Carlos Jensen, Scott King, Victor Kuechler

Free/Open source software (FOSS) is an important part of the IT ecosystem. Due to the voluntary nature of participation, continual recruitment is key to the growth and sustainability of these communities. It is therefore important to understand how and why potential contributors fail in the process of transitioning from user to contributor. Most newcomers, or "newbies," have their first interaction with a community through a mailing list. To understand how this first contact influences future interactions, we studied eight mailing lists across four FOSS projects: MediaWiki, GIMP, PostgreSQL, and Subversion. We analyzed discussions initiated by newbies to determine the effect of gender, nationality, politeness, helpfulness and timeliness of response. We found that nearly 80% of newbie posts received replies, and that receiving timely responses, especially within 48 hours, was positively correlated with future participation. We also found that while the majority of interactions were positive, 1.5% of responses were rude or hostile.

Proceedings of the 44th Hawaii International Conference on System Sciences (HICSS), vol. 1, no. 10, pp. 4-7, Jan. 2011.

2.1 Introduction

People who join a Free/Open source software (FOSS) project often start by downloading the source code, documentation and following the community's discussion on mailing lists and posting questions and comments. Newcomers to an open source software project, often called "newbies," play an important role in the community; they make up a pool of potential future developers and contributors that are vital to the project's long-term survival and growth.

With the vast majority of contributors being volunteers with no financial incentive or contract, there is less preventing them from leaving the project than might be the case for those engaging in a commercial closed-source software project. Therefore, a continuous influx of people willing to contribute is essential to a FOSS project's sustainability. Since newbies are a potential development force in training, it is important to examine how well current tools and community structures help them meet their information needs.

A project's mailing list is one of the main avenues for people of all experience levels to communicate about virtually any aspect of a FOSS project. They simplify mass communication and discussion, since emails sent to a mailing list go out to everyone subscribed to it. These mailing lists are often archived as well, thereby preserving the history and culture of the community for newcomers to examine. These archives are an important source of information regarding past design and implementation decisions, as well as information on project leadership and its decision making process. Newcomers' first interactions with the community will often take place on such lists.

This study examines the first posts of newcomers in eight FOSS mailing lists: two from each of MediaWiki, GIMP, PostgreSQL and Subversion. These

projects were chosen because they are well-known FOSS projects with years of activity and available mailing list data. In addition, these projects span a broad set of software domains, represent variably sized developer and user communities, and target users ranging from non-technical writers and artists, to programmers and system administrators. As such, they are prototypical of many FOSS efforts. Each project had a user list and a separate developer list, both of which we analyzed for information pertaining to poster gender, nationality, politeness, helpfulness and timeliness of response. This study examined the following research questions:

RQ1: Do newbies generally receive timely replies to their first post?

RQ2: How does a newbie's gender or nationality affect the replies he/she receives?

RQ3: How does the treatment of newbies differ between user lists and developer lists?

RQ4: How often do newbie posts result in incidents of flaming?

Focusing on the quintessential first step in the FOSS project joining process and the first interactions newbies have within the community, this study can help to inform FOSS communities about how they are treating their newbies, and what effect this has on future participation. The structure of this paper is as follows: First we review related work done in this area. Next we describe our data-collection and processing. In section 2.4, we describe the results of this data collection, the amount of data, characteristics of the eight mailing lists, etc. In section 2.5, we discuss the research questions and present the data, and in section 6 we present our conclusions.

2.2 Related Work

Several studies have been done which document the importance of newcomers to FOSS projects. Krogh et al's study of Freenet [9] emphasizes the importance of newcomers, noting the high turnover rate among developers, and that recruiting is a concern among the developers. Ye and Kishida [22] assert that newcomers, even passive end-users, are vital to the success of FOSS projects. The departure of key contributors is a real threat in FOSS since there are rarely contracts or obligations to force work on the project. Even end-users are important because they are potentially a source for future contributors, and even when they do not contribute code, they *"play a role similar to that of the audience in a theatrical performance who offers values, recognition, and applause to the efforts of the actors"* [22].

Much has been written about the joining processes in commercial/closed source software (CSS) projects. Sim and Holt [18] studied four employees of a commercial software company joining a new project. They coined the term "software immigrants" to describe these people, as they were not traditional novices given their extensive computer knowledge, but were challenged with acquiring knowledge about a new project. The company assigned mentors, and over many hours the immigrants would learn naming conventions, how to set up the system, use the tools, and so forth. The mentors also played key roles in helping the newcomers integrate socially with the team, introducing them at lunch or during coffee breaks. After 2-3 weeks, the immigrants were receiving task assignments. They were simple at first, usually easy bug fixes and optimizations. However, the immigrants were able to do more complicated tasks after about four months and do them independent of their mentors.

Also studying a closed-source environment with mentors at HP Labs, Berlin [2] found that project newcomers' main strategy for familiarizing themselves with code was to copy and experiment with existing code. When they asked their mentors questions, the mentors would often provide them not only with the basic answer, but also vital information related to system design, design rationale and project history. Mentor sessions were very interactive and in-person. Though each newcomer was assigned a mentor, newcomers felt reluctant to ask questions when stuck. They reported feeling a need to figure things out on their own whenever possible. Berlin notes that as one becomes more familiar with a project and gains knowledge to offer other experts in return, a developer becomes less reluctant to seek their assistance.

According to Ye and Kishida [22], a fundamental difference between OSS and CSS is the "*role transformation of people involved in a project*." In FOSS, all users are potential developers, whereas in closed source, the developers and users are two clearly separated groups. Ye and Kishida identify eight roles for FOSS community member:

- Passive User: simply uses the software
- Reader: uses the software and reads source code to understand how it works
- Bug Reporter: finds and reports bugs, but does not fix them
- Bug Fixer: fixes bugs, needs to understand part of the source code
- Peripheral Developer: contributes features to the software, but sporadically
- Active Developer: contributes new features and bug-fixes regularly
- Core Member: guides and coordinates development

- Project Leader: has main vision, directs project (sometimes later replaced with core members)

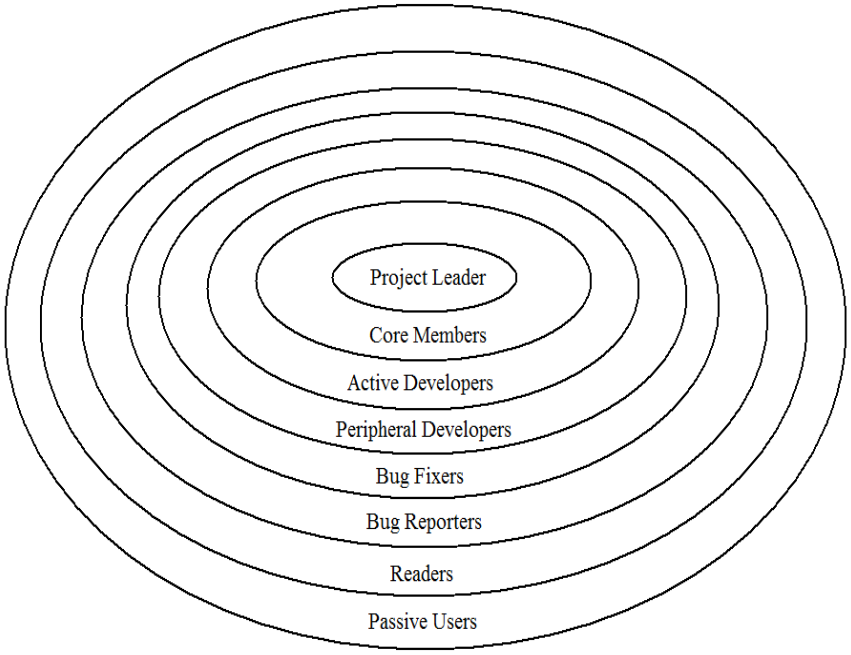

Figure 1. The Onion Model

These roles and people's ability to transition between them is commonly represented as the "onion model" [5, 7]. The basic theory being that while some newcomers might skip some of these steps, progress through these roles at different paces, or not aspire to become a project leader or even a developer, newbies generally start at the outer layer of this model and move inwards until they reach their own personal goals for participation. Jensen and Scacchi [7] studied role migration in the Mozilla, Apache, and NetBeans projects, documenting the paths people follow to join. This study showed that contributors join for a variety of different reasons, and that this pushes them to gravitate to different roles within projects.

The mailing list is a particularly useful tool for coordinating FOSS. A developer for Freenet asserted, "if you wanted to join an open-source project, the first thing you do is get on the mailing list" [9]. Gutwin et al's 2004 [6] study of NetBSD, Apache httpd, and Subversion, three widely used FOSS projects, found that mailing lists and text chat were the main source of group awareness (i.e. who is working on what parts of the project, what their plans are, etc.). They also found mailing lists to be useful for learning who the experts are on a topic. Since the messages are delivered to the whole group, the relevant people will make themselves known by joining the conversation. Text chat connects developers, like the mailing list, but with more off- topic social discussion and in real time.

Studying Freenet, an open source peer-to-peer file sharing network, Krogh et al. [9] discovered developers-to-be would often lurk (observe) on the project and its mailing list(s) before actively participating. This lurking period would last anywhere on the order of weeks to months and was needed to make them feel comfortable before contributing to a technical discussion. In 40% of the cases examined, the first email by a future developer was to an ongoing technical discussion, rather than being an initiation of their own discussion.

Bird et al. [3] analyzed the Apache web server, PostgreSQL, and Python communities. They posit that there is a tension between the time for acquisition of skill and reputation needed to become a developer and eventual loss of interest. The median time between mailing list participation and first patch submission varied greatly. For Apache and PostgreSQL the time was 2 months, but for Python it was half a year. Social status, measured by social networks, also correlated with the chance of attaining developer status.

Lampe and Johnston [12] showed that the feedback given to new members in an online community could have a strong effect on future participation. They studied Slashdot, a popular online tech community where members can leave comments and some members could rate those comments. An analysis of server logs revealed that 11,079 users joined between November 1 and December 6, 2004. 1,763 (16%) of these users posted 6,467 comments. Slightly more than 55% of new members who made comments only made one. The new members who did not receive any feedback on their first posted comment were less likely to comment again, regardless of whether the feedback was positive or negative. These results suggest that feedback of any type is important to encouraging the participation of and the integration of new members in an online community.

The 2003 Freenet study by Krogh et al. [9] followed the project's developer mailing list for a year. 11,210 messages over 1,714 threads were made by 356 people. The creation of new threads was common, with 78% of participants starting at least one. However, 10.5% of participants did not get a reply and did not appear on the mailing list again. Participation was very concentrated with 4 developers (1.1% of population) posting half of the messages. Code commits were also concentrated. About 8% of the participants had CVS commit access, but four out of the 30 made 53% of the commits. As for message content, non-developers would ask more general questions, such as how to get the Freenet software running, and those joining the project would report more bugs.

Crowston and Howison [5] looked at centralization in the bug tracker data for FOSS projects hosted by SourceForge. Centrality was essentially a measure of how much participants on the forums communicated with other people. One of

their main findings was that centralization of interactions on the forums varied greatly across projects, following a normal distribution.

A study of the Apache project forum by Lakhani and von Hippel [11] between 1996 and 1999 found that the vast majority of users were classifiable as purely information seekers, smaller groups as either purely information providers or as both providers and seekers. Sowe et al. [20] also found that a small portion of the information providers posted a very large portion of the answers to questions. Two percent of the providers posted 50% of the answers. Posts by the information seekers were much more distributed, with 24% of them asking 50% of the questions. The study also found that 39% of information seekers received no reply.

Finally, a growing body of work has emerged on the roles and numbers of "lurkers" in online communities. Lurkers in this case refer to members of the community, most often mailing lists, who do not themselves participate by posting. Depending on the size of community, lurking levels have been found to be between 50% [15, 19] and upwards of 90% [8, 13]. While little data has been presented on the lurking rates in FOSS mailing lists, there is little reason to suspect that these would deviate significantly from the above- mentioned.

Lurkers lurk for a variety of reasons. In FOSS communities, new users are actively encouraged to lurk until they learn the culture of the project. This is meant to not only ensure that they avoid the worst social faux pas, but also ensure that they are caught up with the project and familiar with some of the key members. Some lurkers never take the next step and participate in discussions, and some lurkers are newbies in waiting. The latter are likely to be influenced by

their perception of how their colleagues are being treated before deciding whether to take the next step in their journey to participation.

2.3 Methodology

We examined four FOSS projects: MediaWiki, GIMP, PostgreSQL and Subversion. PostgreSQL and Subversion are development and infrastructure projects, which target software developers and system administrators. MediaWiki and GIMP, on the other hand, are end-user oriented projects. This split focus was intentional; we wanted to cover a wide range of users in this study.

Other selection criteria were the availability of mailing list archives and project popularity. Each of these projects had a developer list and a separate user mailing list. The user lists are "mediawiki-l," "gimp- user," "pgsql-general," and "svn-users." They can be thought of as the general catch-all lists to communicate about non-developer aspects of the projects. The other mailing lists are aimed at developers and technical discussions.

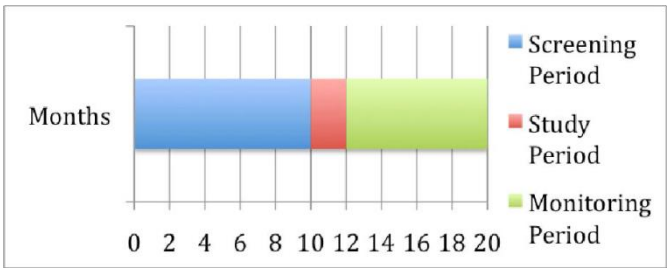

Figure 2. Overview of study sampling period

We obtained mbox files for each mailing list through their official websites or archives. We collected data from January 1, 2005 through August 31, 2007 and divided that data into intervals: People who did not post between January 1,

2005 and October 31, 2006 but posted between November 1, 2006 and December 31, 2006 were classified as newbies. The interval between a newbies first post and August 31 2007 was used to determine what percentage of newbies posted more than once, as well as the replies they may have received. This gave us an eight-month window to look for follow-up posts by newbies identified by the end of our study (December 31 2006).

The following are the message headers we extracted along with brief descriptions:

- "From": The identity of the sender
- "Subject": The message's topic/summary
- "In-reply-to": Identifies message chain, if any
- "References": IDs other messages referenced
- "Date": The date the message was sent
- "Message-ID": Unique message identifier

Unfortunately, some posts contained malformed headers. We excluded all messages with invalid dates, "from" fields and "message IDs" from the study.

A concern when analyzing mailing lists is that of people posting from multiple email addresses. This raises an issue when trying to determine how many posts a user made. There are ways of matching posters to email addresses in a reasonably accurate fashion. Since the "from" header in emails can contain both the sender's address and his or her name, Sowe et al. [20] matched addresses to people by attributing addresses with the same name to the same person. Bird et al. [3] took into consideration small variations in the way names are presented. For instance, "Steven Jobs" might also identify himself as "Steve Jobs." To resolve this identification problem we processed all (email address,

name) pairs for each project's mailing lists to group them into clusters, with each cluster representing a user. The program's main steps are as follows:

1. Normalize names by removing leading and trailing whitespace, periods, and titles (Jr., Dr., Sr., etc.)
2. Break names up into First, Middle, and Last.
3. Merge clusters if they meet any of these criteria:
    a. The email addresses are identical.
    b. The first and last names are identical, or in reverse order.
    c. The first and last names are contained in another address.
    d. The full names are similar.[1]
    e. The email addresses before the "@" symbol are equal and at least six characters long.
    f. The email addresses before the "@" symbol are equal but less than six characters long, and the similarity of the full names is at least 0.75.
4. Generic full names like "webmaster", "admin", and "developer", as well as common names like "Sam" were excluded from clustering unless accompanied by a last name.

Once clustered, we manually inspected clusters to break up clear mismatches.

---

[1] We defined the similarity S of two names, x and y, as:
$S_{x,y} = 1.0 - (D_{x,y} / \max(\text{length}_x, \text{length}_y))$, where $D_{x,y}$ is the Levenshtein distance between x and y [8]. Since $D_{x,y}$ can range from zero (x and y are exactly the same) to the maximum length of x or y (meaning x and y share no characters), we divide by the maximum length to obtain a normalized value over [0,1]. If this is at least 0.85, we consider the names similar. This rule groups names with high similarity, such as "Steve Harding" with "Steven Harding" (S = 0.93).

In order to examine the effects of replies, we needed to group messages into threads. “In-Reply-To” and “References” headers have been found to be unreliable for this purpose [23]. Reference headers can be truncated due to size limitations. Users can initiate new threads while using the “Reply” button to save effort. Inversely, someone might “reply” to a message without using the reply button, causing the message to be classified as a new thread. Grouping messages by subject headers is also problematic. Email clients are inconsistent in the use of “Re:” tags and people sometimes edit them when replying.

We used Kuchling’s [10] implementation of Zawinski’s [23] algorithm with minor modifications. Zawinski’s Algorithm considers “In-Reply-To”, “References” and “Subject” headers together to make reasonable judgments about threading.

For the purposes of this study, we defined a “newbie” as someone who posted a message to a mailing list in the November 1, 2006 to December 31, 2006 interval, and who had not posted a message in the preceding 22 months. While it is conceivable that a project member could take such a long hiatus from a long-running project, we felt reasonably confident that such cases would be exceedingly rare, and that project turnover could in any case have made the person a relative unknown after such a long period of time.

After identifying the newbies, we examined their first post to the list, as well as the first three responses, to their message. For each post, participants were categorized according to nationality and gender as far as possible. We categorized nationality broadly, either as U.S./unknown or non-U.S. poster. True poster nationality is difficult to discern from a mailing list post alone due to the ubiquitous nature of the .com suffix, and the proliferation of free email services

like gmail.com, but there are several clues. The most obvious is the email domain name suffix. People with addresses whose suffix corresponded to a non-U.S. entity according to the Internet Assigned Numbers Authority (IANA) were categorized as non-U.S. participants. While this is of course not necessarily true, what we cared about was outward perception, whether someone replying to a newbie post might assume that this person was a non-U.S. person. Another clue was the use of non-English email software or tags, as evidenced by reply lines such as "am 04.11.2006 um 10:46 schrieb Julien Pons:" and the message signatures, which sometimes included the posters' locations and/or telephone numbers. Again, while none of these are definitive proof that the person is non-U.S.-based, a casual reader might make such an assumption. If there was no basis for saying otherwise, a poster was categorized as a person of "US/unknown" nationality.

A participant's gender was classified as male, female or unknown. Like nationality, we had no accurate ways of determining gender from messages, but we examined names to make reasonable guesses. Each first name was compared against a list of the 1,000 most common male and 1,000 most common female names according to the U.S. Census Bureau, as well as a shorter list of top male and female German names [1, 21]. These lists were chosen for their breadth, as well as their availability. Since the U.S. is ethnically diverse, the top names include many non- English names. Previous studies have shown that the bulk of FOSS contributors hail from North America and Europe [17], amply represented on our lists.

If a poster's name was found exclusively in the male name list, we categorized the poster as male. If her name was found exclusively in the female list, she was

categorized as female. Posters whose names were found in both lists (i.e. unisex names like "Jamie" and "Jessie") or not found in either list (i.e. uncommon names and internet nicknames) were classified as unknown. Exceptions were made for names that were much more common for one gender than the other, such as James which was over 300 times as common for males than females [21]. These results were treated as suggestions, and could be overwritten during manual review given strong evidence in the message body.

Messages were grouped into one of the following:

- Valid and on-topic message.
- Not English/Unintelligible: The reviewer was unable to make sense of the message.
- Subscription message: A failed subscription/ unsubscription to the list
- Spam
- Off topic (provocative), aka "Baiting"
- Off topic (other)

Reviewers also tagged messages that were overtly friendly/polite or aggressive/rude/profane. Finally, each message was rated for its helpfulness:

- Helpful: Provides useful/specific information
- Not helpful: Fails to provide useful/specific information for the topic of discussion
- Not sure
- Not applicable: Message is first in thread, or helpfulness cannot be determined

Because these criteria are subjective, three people rated each message. Threads containing posts that received completely different ratings on general message type or tone were discussed by the reviewers to determine a final rating. When two out of the three raters agreed on a classification, the majority ruled.

2.4 Results

Table 1. Overview of mailing lists and projects studied

Mailing List[2]	Posts	Unique Posters	% by Core Members
mediawiki-l	8,422	806	66%
wikitech-l	7,094	399	71%
gimp-user	2,431	397	56%
gimp-developer	2,267	224	64%
pgsql-general	19,606	1,827	70%
pgsql-hackers	16,360	568	83%
svn-users	13,721	2,331	63%
svn-dev	10,586	559	83%

[2] Statistics calculated for Nov. 1, 2006 to Dec. 1, 2007. "Core member" defined as being in the top 10% most active posters for the list.

Table 2. Newbies posting to lists between November 1, 2006 and December 31, 2006.

Mailing List	Newbies Initiating	Newbies Replying	Total
mediawiki-l	83 (95.4%)	4 (4.6%)	**87**
wikitech-l	19 (61.3%)	12 (38.7%)	**31**
gimp-user	39 (75.0%)	13 (25.0%)	**52**
gimp-developer	10 (76.9%)	3 (23.1%)	**13**
pgsql-general	148 (86.5%)	23 (13.5%)	**171**
pgsql-hackers	36 (76.6%)	11 (23.4%)	**47**
svn-users	271 (90.9%)	27 (9.1%)	**298**
svn-dev	37 (82.2%)	8 (17.8%)	**45**
TOTAL	**643 (86.4%)**	**101 (13.6%)**	**744**

In the two months analyzed, 643 newbies started threads by posting valid messages on one of the eight mailing lists. The results in this study are based on the first four replies in the threads started by those newbies. 101 additional newbies were excluded from out study because they replied to someone else's thread in their first post rather than start their own thread. This was a necessary measure because, in addition to being a small minority of newbies (accounting for only 13.6% of all newbies) it is difficult to classify which subsequent messages are addressed to the newbie vs. the original poster. Replies on threads that a newbie initiated are less ambiguous, at least if one limits the analysis to examining the first replies, as is the case here. This is important because we want to examine how replies to newbies affect their future posting.

To determine how many newbies persisted in their attempts to join the community we examined the postings made during the 3-month period from March 1, 2007 through May 31, 2007 (about 3 months out from our initial observations) for new posts from our newbies. Few (13.5%) posted again in that time. Looking further into the future (from June 1, 2007 and August 31, 2007), this group was again halved (6.4% of the original group). This indicates a significant drop off of newbie participation over time ($\chi 2 = 18.359$, p- value < 0.001). The data clearly shows that most newbie posters do not contribute to the long-term discussion. It should also be noted that a small group (3% of the original newbies) did not post during the March-May interval, but reappeared during the June- August interval. It is possible that these newbies were still lurking in the community after their first postings.

There are many potential reasons for these low numbers. The majority of newbies posting on mailing lists are seeking help to a specific technical problem, and not out of a conscious long-term plan to join the community. Therefore, once these newbies resolve their problem, with or without the help of community, a large majority of them are likely to disappear until some new issue emerges.

Comparing the portions of newbies who continued to post on the mailing lists during the first interval (3-6 months) and the second interval (6-9 months), no distinction could be made between those in user lists and those in developer lists ($\chi 2 = .048$, $p = 0.827$) or between those in developer projects versus non-developer projects ($\chi 2 = 2.771$, $p = 0.096$).

Across all lists, 250 posters (26.5%) were identified as non-U.S. persons. 695 (73.5%) were identified as of U.S. or indeterminate nationality. This contrast with the 2003 FLOSS Survey in which 23.49% of participants were from the U.S.

and the country with the most respondents was Germany at 25.17% [17]. Given the lack of precision in our method for identifying non-U.S. persons, this discrepancy is not surprising.

Only 16 females were identified out of 945 total posters participating on the newbie-initiated threads. Of these 16, only one was not a newbie. 582 males were identified, leaving 347 posters of indeterminate gender. Disregarding people of unknown gender (36.7% of all posters), 2.68% of posters were female. Taking into consideration only the developer lists, 2 out of 176 posters (1.14%) were female. This is in the ball-park of what was found in David et al.'s 2003 survey of OSS developers (1.61% female) [17].

Because we were performing statistics on such small sub-samples, we used Fisher's exact test. Fisher's exact test is more accurate for small samples than the chi-square test [14]. We did not find female participation more or less common on user lists versus developer lists according to ($p = 0.748$).

Turning the attention to our four research questions:

RQ1: Do newbies generally receive timely replies to their first post?

We found that our FOSS communities were generally responsive to newbies. The majority of newbies received a reply within 24 hours. Across all lists, only 23.3% of newbies did not receive any replies. A particularly interesting finding is that newbies posting on user lists were significantly less likely to receive a reply than those posting on developer lists ($\chi 2 = 3.958$, $p = 0.047$). 14.3% received their first reply between one day and one week after their post, and 1.4% received the first reply after one week. This does not include any private replies newbies may have received directly via email.

None of the newbies who failed to receive a reply within 24 hours of posting their first question were still posting to the mailing lists beyond the study period. No significant relation was found between being a newbie and a non-newbie when it came to receiving a response from a core member ($\chi 2 = 0.615$, $p = 0.433$). Considering only threads started by newbies, no significant relation was found between posting on a user list versus a developer list regarding receiving a core member reply ($\chi 2 = 2.538$, $p = 0.111$).

RQ2: How does a newbie's gender or nationality affect the replies he/she receives?

No statistical distinction could be made between the helpfulness of replies to non-U.S. newbies and the helpfulness of replies to U.S. and indeterminate nationality newbies ($\chi 2 = .728$, $p = 0.393$). Newbies who were perceived to be non-U.S. nationals received a lower percentage of polite replies than other newbies in 5 out of 8 lists. Statistically, however, no distinction could be made between the US/unknown newbies and non-U.S. newbies in this respect ($\chi 2 = .041$, $p = 0.839$). A greater percentage of non-U.S. newbies were never replied to, and the ones who did receive a reply received a larger portion of later replies than their US/unknown counterparts.

Gender is an interesting factor given the large ratio of males to females. Would males see FOSS as their domain and display sexist attitudes toward the female newbies? Or would they make a special effort to treat females better because they are so rare in FOSS? Something to keep in mind in investigating this issue is that the sample of females was very small. Therefore, instead of the standard chi-square test for determining independence of factors, we again used

Fisher's exact test [14]. Examining the helpfulness of replies to male versus female newbies, no distinction could be made (p-value = 0.765).

Furthermore, examining the tone of replies to male versus female newbies, no distinction could be made either (p = 1). Therefore, this study found no benefit or drawback to appearing to be a female newbie on a FOSS list from a courtesy standpoint. While the raw data shows a smaller percentage of females receiving replies than males, the difference was not statistically significant, possibly due to the small female sample size (Fisher's exact test, p = 0.330).

RQ3: How does the treatment of newbies differ between user lists and developer lists?

The data indicates that replies to newbies' posts were generally helpful (69.3%). At least half of replies were helpful on every list, except for on the gimp-developer list. There were no statistically significant differences in the helpfulness of replies between the user lists and developer lists of the projects we examined ($\chi 2 = 1.482$, $p = 0.223$).

It is interesting to note that the developer-oriented projects, PostgreSQL and Subversion, had a statistically-significantly larger proportion of helpful replies than the other lists ($\chi 2 = 15.595$, $p < 0.001$). While we did not examine the reasons in detail, one contributing factor may be a potential difference in the technical sophistication of the two newbie populations. The greater traffic of the developer-oriented project lists could also presumably mean that there are more people available to answer the newbies' posts, increasing the likelihood someone knowledgeable could respond.

Out of the 827 replies analyzed, only 6.8% were polite. The vast majority, 91.8%, were rated as neutral in tone. This reflects the often business-like and to-the- point nature of most interactions. An interesting finding is that newbies on developer lists received more polite replies than newbies on user lists across all projects ($\chi 2 = 3.902$, $p = 0.048$). Furthermore, newbies on lists for non-developer projects received more polite replies on average than those on developer projects ($\chi 2 = 6.025$, $p = 0.014$).

RQ4: How often do newbie posts result in incidents of flaming?

1.5% of newbie replies were rated as rude/profane. While this may not seem like a large number, it is important to keep in mind two things: These were only the public replies, our data does not include private replies or flaming between established members, which potentially is much more common. Second, because these replies are public, they have a chilling effect that goes beyond the individual being flamed. As Krogh et al showed [9], newbies often lurk for weeks or months before posting their first message. During this period they study the community and its culture. Given such a long potential lurker period, chances are that they will see at least one newbie getting flamed before they post their first question.

More interestingly, core members or project leadership were responsible for most of the examples we found of flaming of newbies. This can, of course, be explained by the fact that these are the people most likely to have addressed "annoying" questions in the past, and that they are the people with the greatest demands on their time. However, they are also the people who have the greatest

capacity for damaging potential contributors' self-esteem and promote a negative project culture through their example.

2.5 Discussion

Many of the results from our study warrant further study and discussion. We were relatively unsuccessful at identifying non-U.S. contributors, relying on email suffixes and "tells" in the message formatting. This led us to assume the position that someone was a U.S. person unless proven otherwise. The ubiquitous nature of the .com domain suffix and large free email providers meant that the vast majority of our subjects were classified as U.S. persons. While it is entirely possible that U.S. participants in fact dominate the four projects we chose to study, our numbers are so different than what David et al. [17] found that we doubt this. We feel that this is a potentially important issue to examine, though a more robust technique for determining nationality must be used.

There is a significant literature on the effects different cultures have on teamwork [16], and the cost of overcoming the resulting communication problems in terms of lost productivity. Given that many FOSS projects do indeed involve contributors from a wide range of countries and cultures, it would be interesting to see how these projects cope with these barriers, especially given that low-bandwidth communication channels such as email and IRC have been found to be most problematic for resolving such issues [16].

The small number of female participants on these mailing lists was an interesting finding in and of itself. The fact that these numbers closely match the numbers reported for female participation in FOSS seem to indicate that it is not the technical aspects of FOSS projects that have led to such low female

participation rates (after all, the majority of mailing list participants have yet to really delve into the technical aspects of the project), but rather that the culture, reward structures and the social aspects of FOSS themselves may be barriers.

More work needs to be done to determine why such a small population of women participate in FOSS mailing lists. One possible source of data is to examine subscription logs to determine whether women explore FOSS discussions and are repulsed, or whether they avoid these communities altogether.

Another key issue in this paper was the prevalence of the flaming of newbies. While 1.5% may not seem like a very high number, there are several additional factors that have to be taken into consideration. As already mentioned, these are only the public flames, and does not include any private flames sent directly to the user rather than the mailing list. While we have no numbers on how frequent such posts are, it seems safe to assume that they would be more common than public flames. In any case, the 1.5% figure should be seen as a lower-bound. Secondly, we only considered flames sent to newbies. Flaming also likely occurs between established members. To a lurking newbie, determining who is an established member and who is a newbie may be impossible. Thirdly, given the volume of posts on some of these lists, a flaming rate of 1.5% of newbie posts is equal to one flame per week on average, which means that this sort of behavior will be persistently visible.

Determining the extent of the chilling effect that flaming has on FOSS communities would require a more controlled study, which would include tracking and surveying lurkers and newbies. While time- consuming and difficult, we believe that such an effort would be worthwhile.

We were interested to find that there were relatively few differences in the cultures of user and developer oriented projects, at least as they pertain to the treatment of newbies. Some of this may be confounded by the fact that we do not have data on the similarity of the two newbie populations, and especially their respective literacy with FOSS projects and mailing lists. It could be the case that technical communities are more demanding than user-oriented communities, but that the developer-oriented communities encounter more experienced and careful newbies.

One question raised during the review of this paper was whether lurking behaviors are the same on user lists as they are on developer lists. In truth, much is still left to study with regards to lurking in FOSS projects. In fact, most studies of lurkers have been in non-technical communities. It is entirely possible that newbies to user lists are less likely to lurk, and more likely to post their question, thus being less affected by flaming behavior. All that we can say is that in respect to the behaviors we were able to observe, there were few differences between the two. Gaining access to subscription logs would help settle the matter, as one could calculate the average delta between subscription to the list and the posting of the first message for these two communities.

Finally, the most important question, the question driving this study, whether the treatment newbies receive on FOSS mailing lists impacts their transition into more advanced roles in the project, still needs further study. We were surprised to find such low return rates among posters. While the vast majority of newbies post to mailing lists because they have a problem that they need solving, the hope is that this interaction will lead them to deeper involvement. Assuming that this is the long-term stickiness rate of projects, this

is a staggering attrition rate of only 1 in 16.67 posters sticking around. Thus, in order to replace the population of gimp-developer (the smallest mailing list in our study), you would need to recruit a staggering 3,733 newbies, which at its current newbie poster rate would take more than 62 years!

Obviously projects do go through cycles with an ebb and flow of growth and decline, but we also have many examples of FOSS projects that have remained stable and necessary for prolonged periods of time and therefore all is not doom and gloom in the FOSS world. Further study of FOSS development and sustainability is highly recommended.

2.6 Conclusions

We were able to show that the majority of newbies, on the whole, receive prompt replies to their first posts. We found in this study, as previous studies have shown, that prompt feedback is essential to continued participation. Furthermore, the appearance of indifference to newbie questions can have a chilling effect on lurkers, who may decide to give up without even asking their own questions.

We did not find any statistical effects of gender or nationality. It is entirely possible that such effects to occur, however, our statistical power to examine gender effects was limited due to small sample sizes. Our ability to determine national origin was limited as well. It is entirely possible that mailing list members were able to pick up on linguistic clues to determine national origin, which we did not do, and thereby alter their response to these users. It is also possible that large numbers of misclassified users polluted our statistics.

We found a mixed bag with regards to the differences between technical and user-oriented lists. While the user-oriented lists were friendlier, newbies on technical lists were more likely to receive helpful replies. This may in part be because the newbies to the technical projects may have been more skilled at asking the right questions, and therefore received more helpful replies.

Finally, flaming was more common than we expected, and the potential negative effects of such behavior could be significant, especially on women and minorities who may be more anxious about standing out or their ability to fit into the FOSS culture.

2.7 Acknowledgments

We'd like to thank the HCI group at the school of EECS at Oregon State University – Eric Betts, Jill Cao, Eunyoung Chung, Heather Lonsdale, Yunrim Park and Jordan Strawn – for their help in scoring and reviewing mailing list messages, as well as their advice and feedback on this paper. We also thank the contributors and administrators of the Mediawiki, Gimp, PostgreSQL and Subversion projects for sharing their data with us. We'd also like to thank Dr. Tom Rothamel for his advice on Python scripting.

2.8 References

1. About.com. (2008). Top German Names for Boys and Girls. Retrieved April 25, 2008 from http://german.about.com/library/blname_topDE.htm

2. Berlin, L. (1992). Beyond Program Understanding: A Look at Programming Expertise in Industry. Retrieved April 2008 from http://www.hpl.hp.com/techreports/92/HPL-92- 142.pdf

3. Bird, et al. Mining email social networks. Proceedings of the 2006 International Workshop on Mining Software Repositories, 28th International Conference on Software Engineering, Shanghai, China, 2006, 137-143.

4. Black, Paul E. (Ed.), Algorithms and Theory of Computation Handbook, CRC Press LLC, 1999, "Levenshtein distance", in Dictionary of Algorithms and Data Structures [online], U.S. National Institute of Standards and Technology. 14 August 2008. Retrieved April 9, 2008 from http://www.nist.gov/dads/HTML/Levenshtein.html

5. Crowston, K. and Howison, J. The social structure of Free and Open Source software development. First Monday 10, 2 (2005), http://firstmonday.org/issues/issue10_2/crowston/index.html

6. Gutwin, C., Penner, R, Schneider, K. Group Awareness in Distributed Software Development. Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work, Chicago, Illinois, 2004, 72-81.

7. Jensen, C. and Scacchi, W. Role Migration and Advancement Processes in OSSD Projects: A Comparative Case Study. Proceedings of the 29th International Conference on Software Engineering, Minneapolis, Minnesota, 2007, 364-374.

8. Katz, Jon. Luring the Lurkers. Dec. 29, 1998. Accessed Aug. 9, 2010. http://slashdot.org/features/98/12/28/1745454.shtml

9. Krogh, G., Spaeth, S., and Lakhani, K. Community, joining, and specialization in open source software innovation: a case study. Research Policy 32, 7 (2003), 1217-1241.

10. Kuchling, A. M. (2006). JWZ's Threading Algorithm. Retrieved February 28, 2008 from http://www.amk.ca/python/code/jwz

11. Lakhani, K. and von Hippel, E. How open source software works: "free" user-to-user assistance. Research Policy 32, 6 (2003), 923-943.

12. Lampe, C. and Johnston, E. Follow the (Slash) dot: Effects of Feedback on New Members in an Online Community. Proceedings of the 2005 International

ACM SIGGROUP Conference on Supporting Group Work, Sanibel Island, Florida, 2005, 11-20.

13. Mason, B. Issues in virtual ethnography. Proceedings of Ethnographic Studies in Real and Virtual Environments: Inhabited Information Spaces and Connected Communities. Edinburgh, 1999, pp. 61-69.

14. McDonald, J. H. (2008). Fisher's exact test of independence. Retrieved March 3, 2009 from http://udel.edu/~mcdonald/statfishers.html

15. Nonnecke, B., and J. Preece. Lurker Demographics: Counting the Silent. CHI 2000, The Hague, 2000, ACM Press. pp. 73-80.

16. Olson, G. M., & Olson, J. S. (2000). Distance matters.
Human-Computer Interaction, 15, 139 –178.

17. Paul A. David, Andrew H. Waterman and Seema Arora. First Full Findings from FLOSS-US: The Free/Libre and Open Source Developer Survey for 2003.

18. Sim, S. and Holt, R. The Ramp-Up Problem in Software Projects: A Case Study of How Software Immigrants Naturalize. Proceedings of the 20th International Conference on Software Engineering, Kyoto, Japan, 1998, 361-370.

19. Soroka, V., Jacovi, M., Ur, S. We can see you: a study of the community's invisible people through ReachOut. In Huysman, M., Wenger, E., and V. Wulf (eds.), Proceedings of International Conference on Communities and Technologies (C&T2003), Kluwer Academic Publishers,
Amsterdam, 2003, pp. 65-79

20. Sowe, S., Stamelos, I., and Angelis, L. Identifying knowledge brokers that yield software engineering knowledge in OSS projects. Information and Software Technology 48, 11 (2006), 1025-1033.

21. U.S. Census Bureau. (2008). Genealogy. Retrieved April 25, 2008 from http://www.census.gov/genealogy/names/names_files.html

22. Ye, Y. and Kishida, K. Toward an understanding of the motivation of open source software developers. Proceedings of the 25th International Conference on Software Engineering, Portland, Oregon, 2003, 419-429.

23. Zawinski, J. (2002). Message Threading. Retrieved February 28, 2008 from http://www.jwz.org/doc/threading.html

Chapter 3. Gender Differences in Early Free and Open Source Software Joining Process

Victor Kuechler, Claire Gilbertson, Carlos Jensen

With the growth of free and open source software (FOSS) and the adoption of FOSS solutions in business and everyday life, it is important that projects serve their growingly diverse user base. The sustainability of FOSS projects relies on a constant influx of new contributors. Several large demographic surveys found that FOSS communities are very homogenous, dominated by young men, similar to the bias existing in the rest of the IT workforce. Building on previous research, we examine mailing list subscriptions and posting statistics of female FOSS participants. New participants often experience their first interaction on a FOSS project's mailing list. We explored six FOSS projects – Buildroot, Busybox, Jaws, Parrot, uClibc, and Yum. We found a declining rate of female participation from the 8.27% of subscribers, to 6.63% of posters, and finally the often reported code contributor rate of 1.5%. We found a disproportionate attrition rate among women along every step of the FOSS joining process.

Open Source Systems: Long-Term Sustainability, vol. 378, I. Hammouda, B. Lundell, T. Mikkonen, and W. Scacchi, Eds. Springer Berlin Heidelberg, 2012, pp. 78–93.

3.1 Introduction

Although a similar percentage of men and women receive bachelor's degrees today [19], there is a significant difference in the percentage for computer science and engineering. Only 25% of IT workers are women [20], and women earn around 18% of IT-related bachelor's degrees [28]. A smaller percentage of women actively participate in FOSS, less than 2% [7, 9, 14, 18, 24]. Why do so few women participate in FOSS, and what can attribute to these differences?

FOSS projects need to attract and maintain active users. The volunteer nature of FOSS and the general lack of financial incentives to participate lead to high turnover, and the need for a continuous influx of developers [29]. The transition from user to contributor begins on a mailing list. Mailing lists are at the heart of all communication and discussion in FOSS projects, and therefore at the heart of all FOSS projects. Its archival nature also preserves past, present and future design and implementation decisions, as well as the project's evolving culture.

Joining a FOSS project often begins with lurking or silently observing the community by subscribing to its mailing list. Since the majority of communication occurs on mailing lists, we need to understand how this first step affects newcomers' motivations and future behavior.

A previous study found that almost 80% of newbies received a positive reply to their first post, and those who received a timely response were more likely to continue participating [11]. This study also found that messages from men and women were treated similarly in terms of tone, helpfulness, and likelihood of replies, yet significantly fewer women posted (2.68%). The study

did not address the time users lurked on mailing lists before posting, which may be an influential factor.

Building on previous research, we examined subscriber logs and data for six FOSS projects: Buildroot, Busybox, Jaws, Parrot, uClibc, and Yum. We examined the differences between posters and non-posters to determine the attrition rate of women at different stages of the joining process. More specifically we sought to examine the following research questions:

RQ1: Once subscribed to a FOSS mailing list, are women as likely to participate (post) as men?

RQ2: Do women participate (post) with the same frequency as males?

RQ3: Do women lurk longer than men before posting?

RQ4: Do men and women participate (subscribe) for equal amounts of time?

The paper is structured as follows: First we review work related to FOSS communities and project joining. Next, we describe our methodology for collecting and analyzing data. In section 4, we describe our results. In section 5, we discuss our research questions. We finish by reviewing our data and presenting our conclusion.

3.2 Related Work

The influence of FOSS has grown over the last decades and shows that FOSS software can be more reliable and perform better than proprietary software [27]. FOSS encompasses a great variety in projects, from the highly technical Linux Kernel, supporting operating systems like Ubuntu, Fedora and Debian, to

end-user applications like Android, Wikipedia, and business solutions like Open Office and the GIMP.

FOSS is a volunteer-driven development paradigm that brings together developers and contributors from around the world. Only 30% of developers are paid [7, 14] and what motivates contributors can be both intrinsic and extrinsic [5, 14]; a majority of FOSS developer surveys find that contributors are motivated by the opportunity to improve their programming skills [7, 14, 24]. Working on FOSS allows users of any age, education, or experience level to gain valuable skills. One study shows that a significant number of people "wished to improve software products for other developers," [9] and another finds that 77% of respondents thought giving back to the community is very important [14]. A passionate user may also start their own project because he or she has an unaddressed interest or need that could be met by a FOSS solution [22]. FOSS projects are also expanding to address humanitarian needs, which attract a different kind of developer [1].

Surveys of FOSS developer demographics, although outdated, show that FOSS communities are 98% male [7, 9, 14, 18, 24] with an average age of 27 [7, 9, 14, 24]. On average, FOSS developers are highly educated with 30% having at least a bachelor's degree and 10.6% a graduate degree. Most contributions are from Western Europe and the United States. Many FOSS communities have taken steps to address inequalities by starting different programs aimed at recruiting and retaining underrepresented groups. While some projects acknowledge the importance of different types of diversity, most focus on recruiting and retaining female contributors.

There are several text-based forms of communication used in FOSS projects to maintain project awareness. Internet Relay Chat (IRC) is used for real-time communication. Asynchronous communication in the form of blogs, wikis, forums, and bug-tracking systems archive project discussions, documentation and project news. These channels tend to augment mailing lists, the primary medium used for interaction in FOSS communities [10]. This tool allows everyone to participate asynchronously, keep up-to-date with new developments, bounce ideas back and forth, and encourage discussions about the project. Gutwin et al. not that "there is a strong culture of 'making it public' [in FOSS] where developers are willing to answer questions, discuss their plans, report on their actions, and argue design details, all on the mailing list" [10].

Several studies focus on mailing list activity and its influence on future participation. Lampe and Johnston examine the Slashdot community over a one-month period and found that more than 55% of newcomers made only one comment and those that received no feedback were less likely to continue to comment [15]. Krogh et al. studied Freenet's developer list for one year and find that of successful joiners, over three-quarters started a new thread, and that the 10% of participants who never received replies dropped out [13].

Jensen et al. focus on new users instead of developers [11]. They observe the first posts made by new users and how existing members greeted them. After examining the newbie's tone, nationality, and gender they find that newbies received equally prompt replies, but flaming or aggressive replies were not uncommon. They also argue that because this behavior is public it could have far-reaching effects; "Thus, while OSS participants were generally polite to newbies, it is possible that newbie expectations and perceptions of politeness

could be colored by how the regulars engage with each other" [11]. In other words, lurkers may be pushed away based solely on observed negative behavior on mailing lists.

Mailing lists keep an exact record of public discussions. "Mailing lists allow people to find out who the experts are, simply by initiating a discussion: because the messages go to the entire group, the 'right people' will identify themselves by joining the conversation." [10]. This makes mailing lists a great source of information, and subscription logs let us see not only who uses the list, but also who is lurking to test the waters.

FOSS communities are hierarchical in nature and the Onion Model in Figure 3.1, developed by Ye and Kishida [29], shows a very simplistic hierarchy of roles in FOSS communities. Newcomers often begin at the outer layers as passive users who may have subscribed to a mailing list or IRC channel. It is difficult to understand lurker behavior, in part due to the difficulty determining who is a lurker.

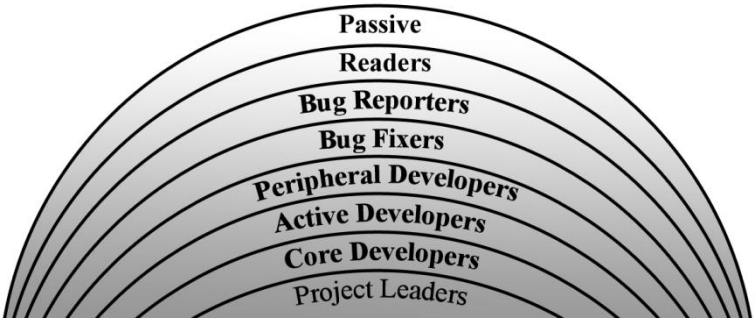

Figure 3. Onion Model of FLOSS joining process

Nonnecke and Preece examine lurking as a transitional phase needed in order for users to feel comfortable contributing to a technical discussion. They find that the transition could last anywhere from weeks to months, but they did

not examine the effects of gender on this process [21]. Other studies, mostly of discussion forums and mailing lists, show lurkers make up from 50% [21, 25] to 90% [12, 16] of an online community. Lurking is recommended to newcomers in FOSS communities as a method to learn the current state of the project, who to talk to, what channels of communication to use, and project culture.

3.3 Methodology

3.3.1 Data Collection

The Oregon State University Open Source Lab (OSL) hosts over 150 projects and "distribute software to millions of users globally" [2]. The OSL acted as agents in gaining access to mailing list data and user subscription logs, with projects' consent. We selected active projects with many users and high traffic. We also asked numerous other communities (not hosted at the OSL) for access to their data, with mixed results; some communities were willing to share their data but did neither use mailing lists nor keep subscription logs. Some were wary about sharing user's information, seeing potential privacy issues. Without a complete set of message files and corresponding logs, we could not use these projects for comparison.

The data we collected was all from mature, highly technical projects. These data sets spanned at least 500 days and have between 73 and 944 subscribers per mailing list.

3.3.1.1 Project Descriptions

We used data from six FOSS projects: Buildroot, Busybox, Jaws, uClibc, Yum and Parrot. From these we selected eleven mailing lists: Buildroot, Busybox, Jaws, Jaws-announces, Jaws-bugs, Jaws-commits, Jaws-developers, Jaws-general,

uClibc, Yum, Yum-devel, and Parrot. Table 1 shows the time coverage of each mailing list.

Buildroot is a technical project that helps users install Linux on embedded systems [6]. Buildroot has one mailing list and a very active community that communicates about commits, questions, bug reports, and patches.

Busybox is a highly technical project that merges numerous UNIX utilities commonly found in GNU fileutils, shellutils and others [2], specifically for embedded systems [3]. This project has one mailing list, busybox, which is the main source of communication and the suggested medium for communicating with the community.

Jaws is a technical framework and content management system (CMS) that encourages users to develop their own modules. We included this project because it was smaller than others and appeared to cater to a broader user base (anyone with web authoring skills). This project contained multiple mailing lists, which we combined to more effectively compare with the others. We analyzed at Jaws, Jaws-announces, Jaws-bugs, Jaws-commits, Jaws-developers, and Jaws-general.

Parrot is designed to compile byte code for dynamic languages. Parrot's website directs users to the parrot-dev mailing list for development and discussion. Other documentation sends new users to parrot-users, which is practically unused [26].

uClibc or the "microcontroller C library" [4] is a smaller alternative to the GNU C Library, and almost all applications supported by glibc are compatible. uClibc has two mailing lists–one for discussion and development (uClibc) and another for source commits (uClibc-cvs), which was dedicated to different files

for bug patches and other code changes. We chose to examine the list for discussion and development since this is where new users are most likely to interact with the community.

Yum or "yellowdog updater modified" is a package management system that provided tools to automate software installation, upgrading, configuring and uninstallation [30]. Yum works with RPM-based Linux distributions. This project had four mailing lists: rpm-metadata, yum, yum-commits, and yum-devel. We chose to examine yum-devel and yum since these mailing lists were more active and included a variety of users, including newcomers. The commits list mostly included code modifications and updates with few questions or other communication. The rpm-metadata list was not available at the time.

Table 3. Time period of each mailing list subscription log per project

Projects		# of subs	Start	End	Days
Buildroot		944	11/20/08	10/12/10	691
Busybox		695	11/20/08	05/18/10	544
	Announces		11/12/07	11/02/10	1085
	Bugs		11/02/07	08/30/10	1063
Jaws	**Developers**	73	11/12/07	10/07/10	1059
	General		11/12/07	11/02/07	1086
	Commits		11/12/07	11/03/10	1087
Parrot		698	07/30/08	05/16/10	989
uClibc		428	12/04/08	05/18/10	529
Yum		360	09/26/08	05/13/10	594
Yum-devel		112	09/26/08	05/18/10	599
All lists (mean)		3310	N/A		~600

3.3.2 Data Parsing

Using documentation from QMAIL [17], we created a Java program to parse the MBOX files and subscription logs to extract the following data (when available):

- Email address
- First name
- Last name
- Subscribe date (if any)
- Unsubscribe date (if any)
- Time on mailing list (Un-subscribe – Subscribe date)
- Number of posts
- Gender
- Date of first post
- Time spent on the list before first post (First post – subscribe date)
- Last Post
- Frequency of posts [(Last -First Post) / (number of Posts)]
- List (used when combining data from multiple lists)

The program iterated over users in the subscription logs for each MBOX file and counted the number of posts made by that person. When the users signed up for the list they could choose to add a first and/or last name in addition to the required email address. We used this information to determine the gender of the subscriber. When available we parsed this information from the logs. When this was not available, we attempted to extract a name from the email address. We used pattern matching to find possible names using the following patterns:

First.last@...
First_last@...
First-last@...

If these schemes did not offer a match, we added the entire username portion of the email to the first name field.

We recorded users' mailing list subscribe and unsubscribe dates; some users subscribed and unsubscribed multiple times. In these cases, we treated the first subscription as the join date and the last un-subscription as the unsubscribe date. Some users did not have a subscription or unsubscribe date in the logs. Since we only had data for approximately two years for each list, some long-time subscribers did not join or leave during that time period. For these we assigned a join and leave date based on their earliest and latest activity. For this study, we focused on newcomer behavior and lurking, so these experienced subscribers were less of an interest. We calculated the total hours spent on the list for each subscriber.

For each user, we counted how many posts, if any, he or she contributed. We did not thread the posts or group them in any way; each post whether a reply or a new topic was counted. For each poster, we recorded the date of their first and last post. From these data points, we determined the amount of time they lurked before posting and their posting frequency.

Using data from the U.S. Census, we matched names to lists of the most common female and male names. We identified 666 users using this process. Some common names are used for both women and men, for instance Alex, Robin, or Morgan. In these cases, we looked at the frequency of use for each gender for each name. If there was a disproportionate use in one gender, we assigned all users with that name to that gender. For example, Alex is ranked as the 63rd most common name for males in the United States, and 990th for

females. Therefore, all Alex's were assumed to be male. In cases where the rankings were close, we put the user in an "unknown" category.

Next, we manually filtered obvious "non-names" such as thepirate@yahoo.com and identified possible names that did not make it on the list of common U.S. names, or names that did not follow the aforementioned patterns. These names were then shown to other researchers and international students via a web application. These "reviewers" could assign a gender, mark the email address as "not a name," label it as an "unknown" for ambiguous names, or skip the name. We asked reviewers to only assign a sex where they were 100% certain. In the end, we identified 1594 users as either male or female, and were left with 975 unidentified users. Grouping unknown and "not a name" together, 41.66% of subscribers were unidentifiable. While this is unfortunate, we believe this represents a good effort and the users identified were a significant and representative sample of the overall community.

Within this dataset were many extreme values; many users contributed little, and a few users contributed a lot. To normalize the data we arranged the users in each mailing list by the number of posts. If we found a jump of more than an order of magnitude between a user and the next highest contributor, we set this as a cutoff point and excluded the user from our set. We did this to prevent a handful of very frequent posters skewing our statistics. Table 2 shows the number of users excluded from each mailing list.

In addition to treating each list separately, we combined all in order to compare data across FOSS mailing lists. As some of the projects are commonly used jointly (Busybox, Buildroot, and uClibc) and projects have multiple lists (e.g.,

Yum and Yum-devel) it is possible that users were counted multiple times in these comparisons.

Table 4. Number of outliers excluded from each data set

Project	Buildroot	Busybox	Jaws	Parrot	uClibc	Yum	Yum-devel	All Lists
Male	0	0	2	0	0	0	0	2
Female	1	0	0	1	1	0	0	3
Unknown	0	0	2	0	0	4	4	10
Total	1	0	4	1	1	4	4	15

3.4 Results

This section explores our finding and relates them to our research questions. First, we begin by looking at the gender of the subscribers. Secondly, we examine the time subscribers spend lurking before their first post. Next, we present the posting frequency by gender of subscribers and lastly, inspect the amount of time users subscribe to the mailing lists.

3.4.1 R1: Gender of Subscribers and Posters

Given that women participate at a disproportionately low rate, even by IT standards, can we determine how early in the FOSS joining process these differences emerge? We know from the work of Jensen et al. that by the time a user posts, only 3% of posters are women [11].

In order to answer this question, we counted the number of women and men who subscribed to each mailing list; we found 1769 men and 162 women. 91.73% of all subscribers were male and 8.27% were female. This was more than a 50% decrease when compared to the 20% rate of women in IT, but still much higher than the population of women who contributed code to FOSS projects. Table 3 shows a breakdown of our findings.

Table 5. Number of men, women, and unknown subscribed to each mailing list

Project	Buildroot	Busybox	Jaws	Parrot	uClibc	Yum	Yum-devel
Male	556	423	48	289	218	177	58
Female	52	29	3	27	30	17	4
Unknown	336	243	22	382	180	166	50
Total	944	695	73	698	428	360	112

Are women as likely to participate (post) as men once subscribed? Table 4 contains an overview of the data. The percentage of female posters ranged from 0 to 10.58% of the total number of subscribers, with an average of 6.63%. This was a statistically significant decrease from the expected value of 8.37% of subscribers ($\chi 2=5.30$, $p= 0.0213$). 110, or 67.90% of women never post after joining a mailing list. In comparison, 1065, or 59.30 % of men never posted after they joined a mailing list.

Table 6. Number of men, women, and unknown posters to each mailing list

Project	Buildroot	Busybox	Jaws	Parrot	uClibc	Yum	Yum-devel	All Lists
Male	254	208	9	58	93	80	29	731
Female	21	8	0	4	11	6	2	52
Unknown	157	115	1	47	83	73	23	499
Total	432	331	10	109	187	159	54	1282

3.4.2 Posting Frequency

Do women post as frequently as men on these lists? In order to determine the number of hours between posts, we examined the time between a user's first and last post and divided this number by the number of posts for that particular user, see Equation 1. We looked at 563 users who posted at least twice. Table 5 shows the average posting frequency for each list, and for the combined data set.

Table 7. Average posting frequency of men, women, and unknown per mailing list

Lists	Male (hours/post)	Female (hours/post)	Unknown (hours/post	Δ M-F	Average (hours/post)
Buildroot	289.96	304.29	320.73	-14.32	301.03
Busybox	326.31	246.67	321.74	79.64	322.76
Jaws	168.67	N/A	N/A	N/A	168.67
Parrot	525.33	527.50	490.61	-2.17	512.47
uClibc	393.24	409.63	825.59	-16.39	565.80
Yum	399.26	115.20	749.36	284.06	541.60
Yum-devel	202.10	133.00	722.53	69.10	419.83
All combined	341.70	306.90	495.17	34.81	395.91

$$\frac{1}{n}\left[\sum_{user=1}^{user=n}\frac{(Last\ Post_{user} - First\ Post_{user})hours}{Number\ of\ Posts_{user}}\right]$$

Figure 4. Equation used to calculate the average number of hours between posts

The majority of our lists (excluding Jaws and Yum) showed that men and women posted equally as often. Statistically we did not find any significant difference between men and women here. We broke separated this set of data into more categories by looking at users who posted at least once, more than once, more than twice, etc.

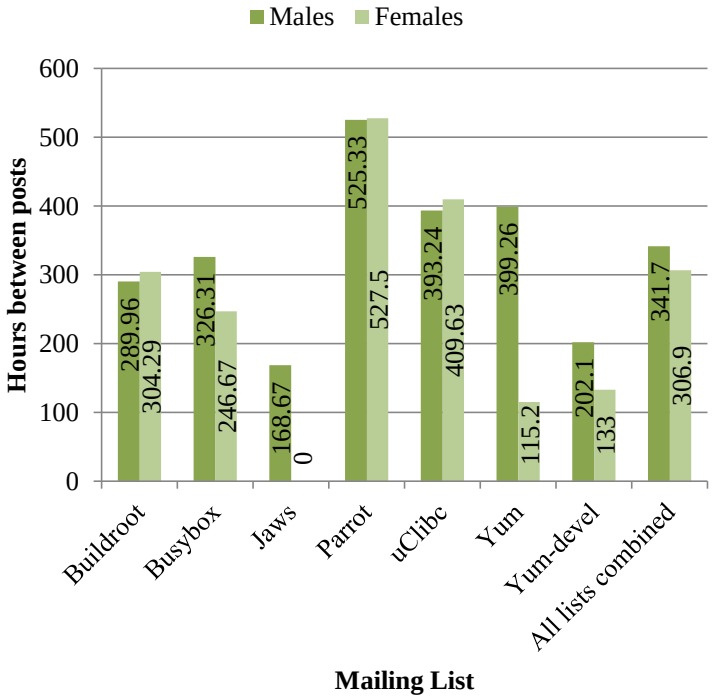

Figure 5. Posting frequency by hours between posts

We also observed that as we look at the "stickiness" of the community, which is measured by continued participation on the mailing list, the participation of women decreased. Women make up about 6% of posters who submitted between 1 and 3 posts, about 4% of those who submitted between 4 and 8 times, about 2% of those who submitted 9 or 10 times, and 1% submitted more than 10 times. Figure 6 shows a graphical representation of the same data. It is interesting to see a near linear trend in the data. Statistical tests show that the proportion of women posters decreased over time ($\chi 2 = 30.346$, $p = 0.0107$). This shows that over time women posted less often, which demonstrates that somewhere along the line women are discouraged from continued participation.

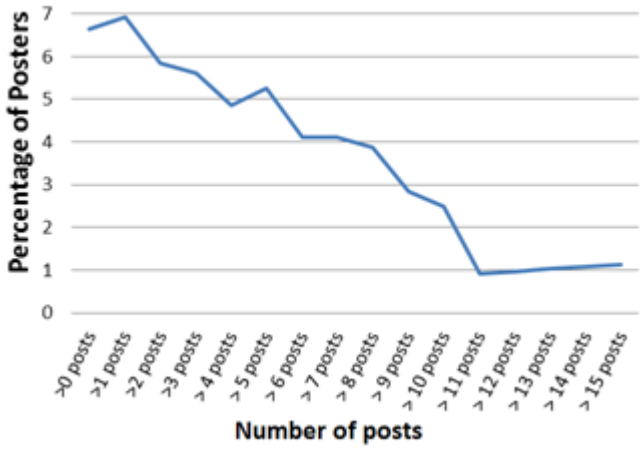

Figure 6. Posting frequency by hours between posts

3.4.3 Lurking Habits

The time spent observing a group before contributing is a formative experience that can encourage or discourage users. In most cases, women lurk less than men (note, no female posters were found in the Jaws data set and so it was excluded from the study). The Parrot mailing list turned out to be an extreme outlier; women lurked 2,235.00 hours, which was more than 50% longer than men who lurked 1,406.47 hours. The data from this Parrot (for both men and women) was an extreme outlier, with averages more than 2 standard deviations from that of the other lists. Parrot was therefore excluded from the analysis. A t-test analysis of the two populations (excluding Jaws and Parrot) showed no statistically significant differences in the mean lurking times (p-value=1.72).

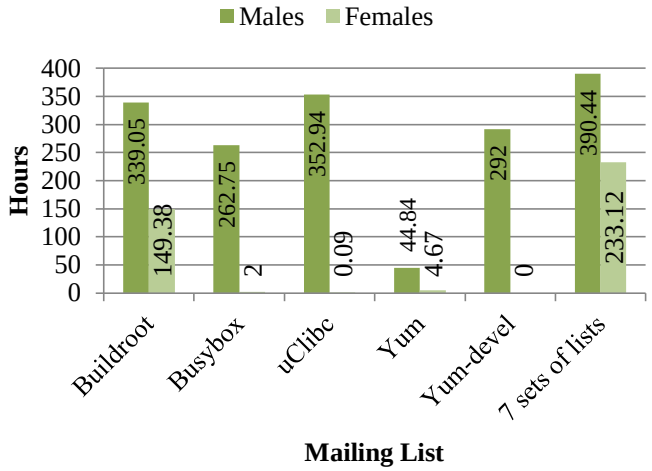

Figure 7. Posting frequency by hours between posts

3.4.4 Subscription Length

Do men and women participate (stay subscribed) for similar lengths of time? Interestingly, women and men subscribed for a similar length of time (except in Yum-devel and Parrot), and there were no statistically significant differences.

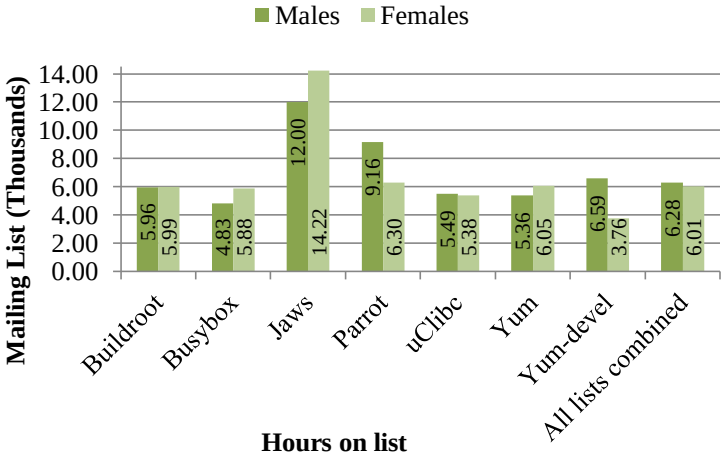

Figure 8. Subscription length by hours on list

3.5 Discussion

Jensen et al. [11] found that about 3% of posters in their sample were women. However, they focused on a variety of attributes, and gender was not the primary focus. As a consequence, their data had a much larger number of unknown participants, which may have skewed their data. We spent more time manually identifying users as male, female or unknown. We believe our results are in line with other studies that found between 1.5% and 2% of code-contributors are female [7, 9, 14, 18, 24]. We know that the FOSS joining process is complex and the commitment time needed to move into a developer role often excludes newbies with family or social commitments, something that may disproportionately affect women.

Building upon the findings in Jensen et al. [11], we found that 8.39% of the FOSS mailing list subscribers were women. This was significantly lower than the 20% of women in IT. In fact, we discovered an attrition rate throughout the joining and lurking process. Figure 6 shows the decreasing participation of women in FOSS communities.

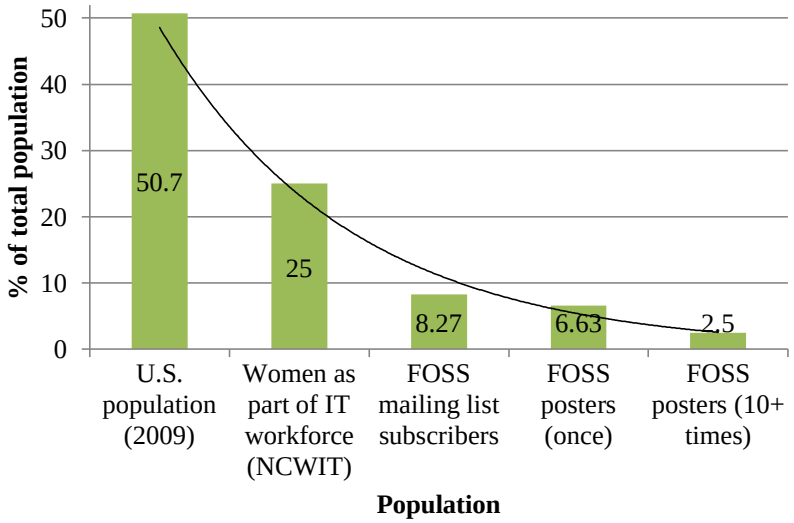

Figure 9. Women in IT and FOSS

Where and why do gender differences emerge? Joining a mailing list is the first step and location in the FOSS joining process that we can collect data. There are however, other non-documented steps: exploring a project's website, scanning documentation, downloading source code, chatting on IRC, or exploring forums, messages boards or wikis. This leaves a variety of interactions prior to, or in parallel with, the mailing lists that may also influence women to turn away from FOSS projects. Of those who subscribe to a mailing list, 67.90% of women and 59.30% of men never post, and in our sample only 6.63% of posters were identifiable as women. We found that the number of women sequentially decreases to just over 1% as the number of posts grow to 10, which is in the range of what has been found by other studies.

Over the last ten years, there has been a push to increase the amount of contributor diversity in FOSS. FOSS projects like Dreamwidth have also appeared and managed to attract a 75% female contributor base [23]. However,

from our data we cannot see that these efforts are having a strong global effect. More data is needed to determine this however, since the last major FOSS developer survey was performed in 2006 [18].

On average, men posted every 341.70 hours and women 306.90 hours. This shows us that although there is a high attrition rate among women, they are not being excluded from the FOSS conversation. That is not to say there are not many situations in FOSS where women have felt uncomfortable, excluded or specifically targeted in a demeaning way [8]. However, if all women were being forced away, we would expect women to post less frequently than males. What is interesting is that along every step of the joining process, we lose a disproportionate number of women. However, given the small sample of women and our choice of projects, it is difficult to draw statistical conclusions about the causes.

The all subscription logs varied in duration. In particular, Jaws and Parrot covered about 1,000 days, and we found that, on average, users from these two lists subscribed longer than users on other mailing lists, which covered less than 700 days. On nearly all of the lists, women subscribed for slightly less time than males, however we did not find any statistically significant correlation between gender and duration of subscription.

After examining posting statistics, we found that the only statistically significant differences between men and women were the average number of posts and the number of women who kept posting declined more sharply than men. We did not examine the type of messages posted, and it is possible that many users were not interested in joining the project, but rather asked one-time questions. It is unclear why women would be more likely to fall into this

category than men. Adding a message-type category to this line of investigation would be beneficial. In addition, the projects we chose are highly technical and therefore comparing them with less technical projects may yield other results.

What drives women away from FOSS in disproportionate numbers? After disproving a number of hypotheses, we are left with two likely factors: women are driven by different sets of motivations and cost-benefit tradeoffs than men or, the social dynamics in projects are more unappealing or hostile to women. As documented by Jensen et al. [11] this may not be blatant or intentional, but the kind of public flaming and aggression documented could be enough to distort participation among a small minority, such as women who may already be hesitant about how they will be received. This does not apply to all women and all projects, since there are many examples of individual success.

Most importantly, what our study shows is that this problem is likely not technical, because most women drop out of FOSS early in their project membership. Efforts to address diversity in FOSS should therefore focus on the first social experiences through programs such as mentorships and making sure novices find the help and support they need.

In the future it would be interesting to evaluate consumer or corporate-oriented projects hosted from a variety of locations. This study used data from very technical projects, which were all hosted in the same location. Also these projects had a similar number of users contributing, and incorporating smaller and larger projects may yield different and possible more representative results. A system to categorize posts from newbies might add insight about a user's intention to join a FOSS community or otherwise.

3.6 Limitations

This study used data from very technical projects of similar size, all hosted in the same location (osuosl.org). Our results may have been different if we had sampled less technical, or smaller or larger projects. Lastly, naming conventions, and the subjective nature of matching name and gender could have introduced errors. We did, however, ask reviewers to only assign a gender if they were 100% certain. Studies have shown that women are more likely to try to obscure their gender online, so our analysis may have been skewed.

3.7 Conclusions

Understanding the reasons behind the gender struggle in FOSS will lead to policies and strategies that encourage greater diversity in FOSS communities. As more companies adopt FOSS software, and FOSS projects diversify to serve a broader population, supporting community infrastructure will be a vital in addressing the issues related to the lack of women in FOSS.

We studied eleven mailing lists and corresponding subscription logs from six FOSS projects, with a combined 3,310 users, of which 1,769 were males, 162 females and 1,379 unknown. We found 8.39% of subscribers were women, which is less than half of the expected 20% population of women in IT. Only 6.63% of posters were women, and the only significant difference we found, in terms of behaviors of men and women, was the average number of posts. Another important finding was the proportion of women who made frequent posts. The percentage of women who posted at least 10 times decreased to about 1%.

On average, males lurked slightly longer (390.44 hours) than females (233.12 hours) before posting to the mailing list for the first time. Also, males

subscribed about 270 hours longer than women (less than 5% difference). However, we did not find any statistically significant values in these averages.

Acknowledgements

Thanks are due to the HCI group at Oregon State University, OSU's Open Source Lab, and the project leaders of Buildroot, Busybox, Jaws, Parrot, uClibc, and Yum, for their willingness to participate in this study.

References

1. About Us. The Humanitarian FOSS Project, http://www.hfoss.org/index.php/about-us
2. About OSUOSL. OSUOSL, http://osuosl.org/about-osuosl
3. Andersen, E.: BusyBox, http://busybox.net
4. Andersen, E.: UClibc, http://uclibc.org
5. Bitzer, J., Schrettl, W., Schroder, P.: Intrinsic Motivation in Open Source Software Development. Journal of Comparative Economics 35(1), 160–169 (2007)
6. Buildroot, http://buildroot.uclibc.org/
7. David, P., Waterman, A., Arora, S.: The Free/Libre/Open Source Software Survey for 2003. Stanford University (2003), http://www.stanford.edu/group/floss-us/
8. Fisher, A., Margolis, J.: Unlocking the Clubhouse: the Carnegie Mellon Experience. In: Proc. SIGCSE Bulletin, vol. 34, pp. 79–83 (2002)
9. Ghosh, R.A., Glott, R., Krieger, B., Robles, G.: Free/Libre and Open Source Software: Survey and Study, Part 4: Survey of Developers (June 2002), www.flossproject.org/report/

10. Gutwin, C., Penner, R., Schneider, K.: Group Awareness in Distributed Software Development. In: Proc. CSCS 2004, pp. 72–88. ACM Press (2004)
11. Jensen, C., King, S., Kuechler, V.: Joining Free/Open Source Software Communities: An Analysis of Newbies' First Interactions on Project Mailing Lists. In: Proc. of HICSS 2011, pp. 1–10 (2011)
12. Katz, J.: Luring the Lurkers (1998), http://news.slashdot.org/story/98/12/28/1745252/Luring-the-Lurkers
13. Krogh, G., Spaeth, S., Lakhani, K.: Community, joining, and specialization in open source software innovation: a case study. Research Policy 32(7), 1217–1241 (2003)
14. Lakhani, K.R., Wolf, R.G.: The Boston Consulting Group Hacker Survey (2002), ftp3.au.freebsd.org/pub/linux.conf.au/2003/papers/Hemos/Hemos.pdf
15. Lampe, C., Johnston, E.: Follow the (Slash) dot: Effects of Feedback on New Members in an Online Community. In: Proc. of the 2005 Int. Conf. on Supporting Group Work, pp. 11–20 (2005)
16. Mason, B.: Issues in virtual ethnography. In: Proc. of Ethnographic Studies in Real and Virtual Environments: Inhabited Information Spaces and Connected Communities, pp. 61–69 (1999)
17. MBOX Documentation. Qmail Mirror Selection (1998), http://www.qmail.org/man/man5/mbox.html
18. Nafus, D., Leach, J., Krieger, B.: Deliverable D16: Gender: Integrated Report of Findings. Free/Libre/Open Source Software: Policy Support, http://www.flosspols.org/

19. U.S. Department of Education, National Center for Education Statistics. The Condition of Education 2011. NCES, Table A-26-2 (2011),http://nces.ed.gov/fastfacts/display.asp?id=72
20. About: Fact Sheet. National Center for Women and Information Technology, http://www.ncwit.org/about.factsheet.html
21. Nonnecke, B., Preece, J.: Lurker Demographics: Counting the Silent. In: CHI 2000, pp. 73–80. ACM Press (2000)
22. Raymond, E.S.: The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary. O'Reilly, Beijing (1999)
23. Robert, K.: Standing out in a Crowd. Keynote Presentation. In: OSCON 2002, Dreamwidth (2002), http://www.oscon.com/oscon2009/public/schedule/detail/10173
24. Robles, G., Scheider, H., Tretkowski, I., Webers, N.: Who Is Doing It? A research on Libre Software developers (2001), http://widi.berlios.de/paper/study.html
25. Soroka, V., Jacovi, M., Ur, S.: We can see you: a study of the community's invisible people through ReachOut. In: Huysman, M., Wenger, E., Wulf, V. (eds.) Proc. of Int. Conf. on Communities and Technologies, pp. 65–79. Kluwer Academic Publishers (2003)
26. Users & Documentation. Parrot VM. Web (August 21, 2011), http://parrot.org/dev/docs/user
27. Wheeler, D.: Why Open Source Software/Free Software (OSS/FS, FOSS, or FLOSS)? Look at the Numbers! (April 16, 2007), http://www.dwheeler.com/oss_fs_why.html

28. Women, Minorities, and Persons with Disabilities in Science and Engineering. U.S. National Science Foundation (2011), http://www.nsf.gov/statistics/wmpd/
29. Ye, Y., Kishida, K.: Toward an Understanding of the Motivation Open Source Software Developers. In: Proc. ICSE 2003, pp. 419–429 (2003)
30. Yum Package Manager, http://yum.baseurl.org/

Chapter 4. The Emergent Qualities of Diversity in Free and Open Source Software Communities: A Critical Review and Theoretical Discussion

Victor Kuechler and Carlos Jensen

Over the last forty years, the free and open source software (FOSS) movement has grown into a remarkably large association of software projects that provide solutions to many diverse home, business, government, and industry users. With such size, one would expect the demographic diversity of FOSS's developer base to mirror the diversity of its users. Several large FOSS developer surveys from early 2000 show that the FOSS developer community is not diverse in its demographic attributes. The discussion surrounding this disparity has persisted with constructive—diversity statements and community initiatives, and destructive—flaming and backlash, results. In this paper we present a critical discussion of the demographic and structural characteristics of the FOSS community that directly and indirectly affect diversity. We also provide a theoretical explanation of FOSS's current homogeneity using group faultlines theory and further analysis through the use of critical systems thinking. Our discussion suggests that FOSS's cultural structure and inherent homogeneity have created projects that are resistant to change, require strong mediation to resolve diversity conflict, and have consequently forced some new contributors to assimilate or leave.

First Monday: Peer-reviewed Journal on the Internet
http://firstmonday.org

4.1 Introduction

The FOSS[3] movement, which emerged from a small, select group of engineers and hobbyists [74], has become a robust association of software projects that span the globe. Many factors have contributed to the evolution and success of this movement, which Steven Weber attributes to its unique development, licensing, and social model and the contributions of highly talented programmers, to name a few [105].

FOSS has seen widespread adoption and use, from novice users and hobbyists to administrators and IT professionals [40], [19], [104], [86]. The number of FOSS projects is growing, and governments and corporations are increasingly adopting FOSS [86]. According to the FSF/UNESCO[4] free software directory, there are now more than 6,000 FOSS programs and tools [25]. A 2005 survey of 512 businesses found that 87% use FOSS [104]. A recent study by Accenture found more than two thirds of large organizations anticipated investing in FOSS in the coming year [40]. In a quantitative study, Wheeler showed that FOSS has a significant market share, and "it is often the most reliable software and in many cases has the best performance" [107].

Despite this growth, several large FOSS surveys conducted between 2000 and 2006 discovered that the FOSS community is approximately 98% male, 27-years-old, and 85% of which are from North America and Western Europe [80], [28], [56], [68], [17]. There were limitations to these surveys, including skewed

---

[3] While key differences between "free software" and "open source software" exist, the distinction is often hazy in the minds of developers. In one survey, 56% of developers claimed they work on open source, while their colleagues, working on the same project, claimed they work on free software [2]. In this paper FOSS refers to both "free" and "open source" software and their associated communities.

[4] Free Software Foundation/United Nations Education, Scientific and Cultural Organization

sampling and language barriers. Nevertheless, these surveys are our best source of data about the FOSS community, and show that the developer community is demographically homogenous. At first glance, this may not seem like a significant issue since the number and scope of FOSS projects and adoption has grown steadily. A 2009 study conducted by Georgia Tech and Red Hat found that FOSS is being adopted globally, with France, Spain, Germany, and Australia topping the list [69]. It is this very growth and expansion that is creating a growing demand for more FOSS developers, and to meet the needs an increasingly diverse user group. As FOSS adoption expands, the developer community is increasingly diluted, increasing the need to expand the developer population. In a corporation, this is the responsibility of a human resource (HR) department, providing employee training in diversity and guidance to managers on how to maintain healthy teams. This function does not exist in the majority of the FOSS projects.

FOSS communities are self-regulating, and struggle to mediate the growing pains within projects, especially related to diversity. This has manifested itself in skirmishes and heated discussions over diversity-related events, conflicts [82], and initiatives [27]. This has led to backlash, potential project forking[5], flaming[6], and increased developer turnover.

Beyond the community's need for contributors, one could argue that embracing diversity is part of the tenets of FOSS. The Open Source Initiative (OSI), a non-profit advocate of open source, requires that an open source license

---

[5] When a project breaks into more than one part to develop in parallel with different development teams.

[6] Flaming is a heated or hostile response to a comment. Flaming often takes the form of ad hominem abuse, and is common in FOSS [35].

"must not discriminate against any person or group of persons" [72], barring projects from restricting who can use software. However, research has shown that demographic attributes affect the way users or developers use tools [9], [84]. One could therefore argue that a homogeneous developer population leads to software that will ostracize users who do not share the same characteristics.

Diversity in FOSS has become a politicized issue, especially the lack of female participation. Discussions about diversity have taken place on blogging sites, FOSS project mailing lists, and at conferences [2]. As a result, diversity awareness has increased in FOSS and the greater software community. But, to the authors' knowledge, a theoretical explanation and discussion of diversity has not taken place. This has led to more focus being placed on the symptoms rather than the root causes of the community's lack of diversity.

For this reason, we present a critical review and theoretical discussion of diversity in FOSS. This discussion is based on two theoretical frameworks: group faultline theory and critical systems thinking. Group faultline theory posits that conflict will arise from faultlines, the alignment of demographic attributes (e.g., age, ethnicity, gender, etc.), and the inherent subgroups formations that they creates. Critical systems thinking, which is a mixed-method approach, posits that any problem-context (e.g., lack of diversity) is a systemic issue with many contributing factors. We believe this two-pronged approach will shed light on the underlying causes and processes driving this lack of diversity, broadening the current discussion and providing a foundation for new initiatives and future research.

In our discussion, we found that group faultlines maintain the homogeneity of the FOSS community. We also find strong evidence that the current structure of FOSS's cultural system has made FOSS projects:

I. resistant to change (e.g., new ways of thinking, new contributors with different backgrounds, etc.)
II. brittle and less willing to compromise (e.g., favors project forking rather than negotiation).
III. require strong governance and mediated dialogue when conflicts arise.
IV. indirectly pressure new contributors to assimilate or leave.

We also posit that small FOSS projects (e.g., niche projects) with high homogeneity rarely deal with diversity conflict, but their growth is inhibited. While these results seem severe, they reveal important implications about the FOSS community and highlight the primary goals of this paper: 1) to show that diversity is a critical issue and should be taken seriously, especially regard of community health, effectiveness, and sustainability; 2) to show that simply increasing the population of diverse members in the FOSS community is only addressing the symptom of the problem, instead of their root cause; and 3) to show that integrating social theory and critical systems thinking will reveal many of the nuanced and systemic issues currently affecting diversity in the FOSS community.

This article is structured as follows: First we present a brief introduction to FOSS. Next, we present demographic data on FOSS developers. We then present group faultlines theory and critical systems thinking as a theoretical framework

for understanding FOSS's current diversity, followed by a discussion of gender faultlines. Next we discuss FOSS's cultural system as two distinct subgroups and several unique structural characteristic. We conclude by discussing the dynamics of FOSS conflict.

4.2 Background

4.2.1 A Brief Introduction to FOSS

FOSS is defined by a unique development and license model. FOSS must be freely redistributable and allow modification and derived works [72]. There are many types of FOSS licenses with varying degrees of legal freedom. FOSS development begins when a project initiator picks a problem that is important to him or her, often referred to as "scratching a developer's personal itch" [77]. This is followed by closed prototyping and eventually the project is opened to community participation [77].

The FOSS community is a collection of independent projects that share this development and licensing model, as well as some core beliefs (see Section 2.2). Some larger projects are organized as sub-communities. For example, Apache, a web server project, is composed of 333 sub-projects [86] that consider themselves "not simply a group of projects sharing a server, but rather of community of developers and users" [98]. Other large FOSS communities include Eclipse (171 projects), Mozilla (77) and Drupal (613) [86].

FOSS development is distributed, and projects rely on computer-mediated communication (CMC), including Internet Relay Chat (IRC) and mailing lists for synchronous and asynchronous communication. They also create and share artifacts remotely through project portals, source control systems, testing tools,

and bug trackers. These tools are frequently used in many forms of collocated and noncollocated software development, and are not unique to FOSS. The importance of these tools depends on the complexity of the project and community organization, but mailing lists and IRC are crucial to building and maintaining group awareness [33]. Hemetsberger & Reinhardt show that mailing list archives are the primary repositories for community knowledge [35].

According to a 2002 study of FOSS projects on Sourcefore.net, the majority of projects only have one developer and the average project has four [53]. In this regard, the most popular and famous projects like Linux distributions, Firefox, OpenOffice.org, and Apache are very atypical.

FOSS is predominantly developed by volunteers, and as a result there is no formal contract, and turnover of developers is expected [70]. Given the small number of developers on a typical project, and the fact that most FOSS developers are volunteers, recruiting newcomers is crucial. Since the majority of code contributions come from a small set of developers, the loss of a core contributor could mean project collapse [54], [29]. It is also important to note there is significant overlap of contributors between projects [51].

Since FOSS projects rely on volunteers, users are viewed as potential contributors. Ye and Kishida studied "role transition" of users (using the software, motivating developers), bug reporters (reporting and documenting problems, submitting feature requests), and active developers (contributing code). They found that users transition between roles following an "onion model", with core contributors at the center and a larger group of passive users at outer layers [113], [36]. Each layer in this model represents the progression of knowledge, status, and skill within the community. More recently, Jergensen et

al. studied the overlap of developers across FOSS projects and found evidence to suggest that an "onion patch" model is a more suitable. Knowledge gained in one project transfers to other projects, which allows some contributors and developers to transition between roles more quickly [51]. Jensen and Scacchi, in a study of Mozilla, Apache and NetBeans, found more complex role transition paths. Users gravitate toward roles suited to their skills and interests; not everyone wants to be a core contributor [50].

FOSS projects are also meritocracies, where "participants gain influence over a project through the recognition of their contribution" [26], and especially code contributions. A common adage within these communities is "shut up and code." This mentality is reinforced Jim Whitehurst, president and CEO of Red Hat, when he notes that "in the end, nothing matters but the code. The code wins" [108]. Similarly, Roy Fielding notes that "The more work you have done, the more you are allowed to do" [24].

Beyond this notion of meritocracy, others have examined the connection between social interaction and status in FOSS. Lampe and Johnson found that participation of new posters to Slashdot was positively correlated with receiving feedback from others [58]. The same effect was found in a study of Freenet's mailing lists [29]. Bird et al. discover that social status is a motivator for becoming a developer [5]. Steward states in his analysis of large FOSS projects "that in the process of status attainment, community members tend to evaluate a participant's reputation according to publicly available social references" (e.g., comments and code submissions) [90]. A study conducted by Xu et al. of FOSS developers found that community identity and obligation have a strong effect on participation, and that trust and shared goals, in turn, support obligation and

identification [112]. Ghosh et al. also found that developers participate for social reasons [30].

With this is mind, joining a FOSS project is truly a process of socialization. Through an ethnographic approach, Ducheneaut finds "successful contribution to an OSS project is much less about technical expertise than about the construction of identities. Despite the rhetoric surrounding Open Source, which basically argues that 'anybody can contribute,' it seems instead that only those few participants who have managed to define and present themselves as 'software craftsman' eventually reach the status of developer" [18].

Status and feedback are not the only drivers of FOSS participation. Lakhani and Wolf found that personal enjoyment and "how creative a person feels when working on the project, is the strongest and most pervasive driver." They also report an "interplay between extrinsic (payment, need for software, community acknowledgement) and intrinsic (enjoyment, obligation) motivation: neither dominates nor destroys the efficacy of the other" [56]. The volunteer nature of FOSS logically places motivation at the heart of open source communities.

4.2.2 Characterizing Diversity

The body of diversity research is extensive, and consequently, there are many definitions of diversity with varying contexts. To complicate this further, Mor-Barak also notes that the term "workforce diversity" (and presumably other concepts of diversity) does not have the "human resource connotation it has acquired in other parts of the world" when translated to other languages and cultures, which suggests that some definitions may not be transferable [67]. Therefore, a universal definition of diversity is problematic. In this paper, we

define diversity demographically because our treatments of diversity are related to demographic information from surveys, which is the only available data on the FOSS developer community at large.

Demographic attributes are divided into two dimensions: primary or relation-oriented attributes, and secondary or task-oriented attributes (See Table 8) that are "more mutable, less visible to others around us, and more variable in the degree of influence they exert on our individual lives" [44], [65] [47].

Table 8. Demographic Attributes

Dimension	**Description**
Primary or relation-oriented	
Gender	Male or female
Ethnicity	Culture, language, ancestry, race
Mental or physical ability	Ability to learn or retain knowledge or skills, or perform a mental or physical act[a]
Age	Broad age groups
Socio-economic	Class and educational opportunities
Sexual orientation	Emotional, romantic and sexual behaviors, or affiliation
Secondary or task-related	
Education	Formal education and training
Organization role and rank	Real or perceived status and influence based on power or achievement
Tenure	Previous work experience and achievements
Work style	Method of completing work
Communication style	Aggressive, passive, or assertive in the context of different media (email, face-to-face, etc.)
Geographic location	Not necessarily tied to ethnicity but rather current location
Background	Life history, experiences & perspectives
Motivation and personal outlook	Extrinsic (recognition, financial incentives) and Intrinsic (Altruism, personal achievement, attitude)

These dimensions form the basis of demographic diversity. However, not all are relevant to virtual communities like FOSS, especially since interactions in FOSS are mostly text-based. The demographic attributes we address in this paper include gender, age, ethnicity, education, tenure, geographic location, and motivation. We have chosen these demographics because this information is readily available in the survey data currently published on FOSS developers.

4.2.3 FOSS Developer Surveys

Demographic data for FOSS developers were collected in five large surveys, all from roughly the same period in time (2001-2006). Each survey had slightly different focus and employed different methodologies, but they provide a baseline for understanding the composition of the community.

We have several concerns about the survey data, which should be noted. The first is that researchers disproportionately sampled from large projects (see Table 2). It is easier to recruit from a handful of large projects than many small projects. Sampling primarily targeted English speaking cultures, and areas where portals, such as Sourceforge, are commonly used. Recruitment for these surveys was primarily through mailing lists, and all surveys were written in English, though some were translated to a few other languages. FLOSSPOLS focused predominantly on European projects.

Seven years have passed since this data was collected, which is a significant amount of time in computing. A growing number of studies have looked at individual communities, but only a handful has looked at multiple projects. There is a wealth of data about the demographics of individual projects, but we have not used that data in this paper. This data could add to our

understanding of diversity in FOSS, but we have no way of determining the idiosyncrasy of the data. For instance, while most surveys find a lack of women in FOSS, it is possible to find projects with a large female participation, such as the Dreamwidth, which has a 75% female participation [79]. Table 9 gives a basic overview of the five main surveys.

Table 9. Geographic locations of FOSS Developers

Study	**Year**	**# of Respondents**	**Geographic Location** (% of N by country or area)
Nafus et al. [68][7]	2006	361 developers 128 employers. 1,541 contributors	Employer Survey: 36 - Germany 26 - France 16 - United Kingdom 8 - Spain 6 - Belgium <9 - Other
David et al. [17]	2003	1,588 developers	53 - Western Europe 27 - North America, 8 - Russia/ East Europe 5 - East Asia, 3 - Oceania 3 - Latin America 1 - Middle East / Africa
Ghosh et al. [28]	2002	2,700+ developers	71 - Europe or Russia 13 - United States 17 - Rest of the world
Lakhani and Wolf [56]	2002	1,684 Sourceforge developers	47 - Americas 42 - Europe

[7] Derived from Deliverable D10 (Skills Survey Interim Report); D16 (Gender: Integrated Report Findings); and D17 (Gender: Policy Recommendations). D10 draws from developer and employer surveys targeting Europe. D17 includes ethnographic results for parts of Europe (2004/2005) and an online survey.

			11 - Rest of the world
Robles et al. [80]	2001	5,600 developers[8]	Miscellaneous data worldwide

4.2.4 Comparing Results on FOSS and Diversity

4.2.4.1 Primary Diversity in FOSS

4.2.4.1.1 Gender

Table 10. Female Participation in FOSS

Study	Female Ratio
Nafus et al. (2006)	1.5%
David et al. (2003)	1.6%
Ghosh et al. (2002)	1.1%
Lakhani and Wolf (2002)	2%
Robles et al. (2001)	1.4%

A consistent finding across surveys is low participation of women, with findings ranging from 1.1% to 2%. The rate at which women are severely underrepresented in FOSS is startling given that women are 25.6% of the U.S. IT

[8] Authors used several data sources: SourceWell, a database of more than 1,000 FOSS applications; The Debian Database, with information on more than 700 developers worldwide; Codd, an application that associates code contributions to programmers; and lastly, Widi (who is doing it), an online questionnaire posted to several FOSS mailing lists. While this presents a wealth of data, it makes it difficult to compare to other surveys, or determine the generalizability of the results.

workforce [111], and 17.7% of bachelor's degrees in engineering and CS are awarded to women.

Previous research has identified possible reasons for the lack of gender diversity in FOSS [68], [63]. Flaming is one practice that creates an uninviting culture for women is. Nafus et al. notes that flaming is more detrimental to women, eroding their self-confidence [53]. In addition, the tools used by the community, and the way that information is organized can introduce gender bias. Burnett et al. find that while men like to tinker (experiment with features), women prefer to stick to features they know or consult documentation, and significant difference exist in the way men and women use and create with software [9]. Undocumented tools and the loose organization of FOSS may thus present a barrier to women. Nafus et al. [68] report that women and men participate differently; women are more active in activities seen as less valued or visible: "Men are overrepresented in activities such as coding, testing, as well as reading and writing bug reports. All these activities are closely related to the production of source code. On the other hand women are more likely to engage in the less technical aspects of F/LOSS production. They engage more in documenting, providing graphics and sounds, moderating mailing lists and organizing workshops. In most projects these activities are considered to be less crucial and are, therefore, less prestigious" [68]. Whether women gravitate to such roles or are pushed to accept such roles is worth further study.

4.2.4.1.2 Ethnicity

There is little data on the ethnicity of FOSS contributors. Ghosh et al. claim that though we don't know the ethnicity of developers, knowing where they live gives us insights. Since FOSS demands specific skills and computing knowledge,

the United States and Western Europe (particularly Germany and the U.K.) have become hubs for FOSS. Developers are twice as attracted to work on FOSS projects in the United States compared to Europe and three times more than France and India [29].

4.2.4.1.3 Age

Table 11. Age Diversity in FOSS

Study	David et al.	Ghosh et al.	Lakhani and Wolf	Robles et al.
Average age	27	27.1	30	27

Research on the Wikipedia community has found a strong correlation between gaps in age and experience, and increased hostility, lower productivity and high drop-out rates [14]. Bird et al. note that, in FOSS, there is a constant competition among developers to "race to the top," vying for reputation [5].

2.4.2. Secondary Diversity in FOSS

4.2.4.1.4 Education Diversity

Table 12. Educational Diversity in FOSS

Education Level	David et al.	Ghosh et al.	Robles et al.
PhD	36.7%	9%	4.8%
Master		28%	11.9%
Bachelor's degrees	36.4%	33%	28.6%
Vocation/ Apprentice	6.2%	11%	29.5%

High school	19.4%	17%	23.4%
Elementary	1.3%	2%	1.6%

While the numbers differ slightly between surveys, the community appears to be divisible into thirds: A third of developers are pursuing or have graduate degrees, a third bachelor's degrees, and a third vocational or high school diplomas. This is different from the educational distribution in the United States, where only 10.6% of the population has gone beyond a bachelor's degree, 30% have a bachelor's degree, and 59.4% have the equivalent of an associate's degree or less [13]. It is also important to note that the FOSS population has many younger adults, and that some are likely still pursuing educational goals. Although no survey breaks participation down by academic fields, or the number of contributors with training in computing, we assume based on the number of FOSS developers currently working in IT that this is a significant subgroup.

4.2.4.1.5 Organizational Role and Rank

Table 13. Organization Role and Rank in FOSS

Employment	David et al.	Ghosh et al.	Lakhani and Wolf	Robles et al.
Work in IT	65.3%	83%		
Paid to FOSS	27.4%		30%	20.7%
Employed	67.8%	79%		
Students	28.8%	17%		

With the growing interest in FOSS there has been an influx of professional developers. David et al. found that 63% of respondents were “employed by a private firm or organization that produces proprietary software” [17]. Of employed respondents, 70% had employers who are aware of their FOSS involvement [17]. Corporations (e.g. Oracle, IBM, and Google) support many FOSS projects, either financially or through developers, or they buy and release software under a FOSS license. By providing support through hosting, companies like Microsoft (Codeplex), Google (Google Code), and IBM (Eclipse foundation) also support FOSS.

4.2.4.1.6 Tenure

Gosh et al. found that only 8% of respondents started working on FOSS before 1990. The mean starting year was 1996 and involvement significantly increased after 1998 [28]. David et al. report that 50% of respondents joined FOSS after 1998, and only 31.6% before 1997 (mean 1996). They found that "the majority of respondents used [FOSS] regularly or occasionally for recreation or in their coursework" before joining FOSS [17].

4.2.4.1.7 Motivation

Given that most FOSS developers are volunteers, understanding motivations is crucial to understanding who contributes. Motivations have been studied by economists, psychologists and computer scientists alike, finding that FOSS contributors are driven by both intrinsic and extrinsic motivations [57]. Although these have distinct attributes, in many cases they overlap with hidden complexity.

Intrinsic motivation (self-motivation) is based on internal rewards. Intrinsic motivation could include boosting one's sense of accomplishment, feelings of community belonging, or the desire to gain new knowledge or skills. Extrinsic motivations are external to the body [62], and can include the hope of recognition from the community, job opportunities, travel or payment.

Ross-Ackerman asserts that the intrinsic motivation most important to FOSS is altruism—selfless giving for the benefit of others, where the giver receives satisfaction from helping or living up to a moral commitment [81]. Altruism drives Wikipedia—"knowledge shared for all"—and the Humanitarian FOSS (HFOSS) movement, which includes Sahana, a disaster management and preparedness system; OpenMRS, a medical records system; the GNOME Accessibility project; and the One Laptop Per Child (OLPC) project.
Bitzer et al. and Hars and Ou [34] found that contributors are driven by both intrinsic and extrinsic motivations, benefitting from using the software they create, the fun of the work, and the gift culture of the community they operate within [6].

Lakhani and Wolf found the intellectual challenge of programming to be the strongest reason for contributing to FOSS, followed by improving programming skills. One third of respondents listed the product, "open source code" as a reason for participation. Lakhani and Wolf [57], and Robles et al. [40] also find that knowing they were meeting a need was a significant motivator. Ghosh et al. find that almost 80% of developers joined FOSS because "they wanted to learn and develop new skills," while half said they "wanted to share knowledge and skills with other developers." Gosh et al. also report that 30% "wanted to participate in new forms of cooperation," and from a more ideological

perspective, "wished to improve software products for other developers" and that "software should not be a proprietary product" [28].

David et al. find that 77.8% of respondents considered it "important" or "very important" to "give back to the community after using free software themselves." 52% were "motivated to interact with like-minded programmers," and 68% wanted to become better programmers, a result that supports the work of Ghosh et al. A majority are also driven by ideology, wanting to promote FOSS and "provide an alternative to proprietary software" [17].

Also, the fact that many corporations and organizations are looking to participate in or adopt FOSS adds strong extrinsic motivation to current and future professional developers looking to either meet a mandate from their employer, or to acquire marketable skills.

4.3 Theoretical Foundations

4.3.1 Group Faultline Theory

In 1998, Lau and Murnighan [59] proposed a treatment for understanding work group conflict as the results of *group faultlines*, which divide a group's members on the basis of one or more demographic attributes. When multiple demographic attributes align, the strength of the faultline between members who associate with one demographic group others increases, which can complicate group sensemaking, affect group and subgroup dynamics, hinder political processes, and promote conflict. When faultlines exist, they create subgroups based on demographic alignment [46]. Every alignment of demographic attributes is considered a "dormant" faultline until it becomes salient enough within a group to create a perceived subgroup, known as "faultline activation".

Research "suggests that both dormant and active faultlines influence group processes and outcomes" [97]. In contrast, two studies also found that faultlines decrease conflict [97], [60].

The basis of group faultlines theory, as well as the justification for its application in virtual communities, is grounded within the domain of social identity theory, including social identification, self-categorization, and similarity attraction. Social identity theory, originally developed by Tajfel and Turner [93] and supported by many others, posits that human identities are purely social constructs [48]. *Identity*, or the "self", is relational and contextual. As we perceive others around us, we relate our "selves" through social categorization, which Turner et al. call "self-categorization" [100]. Additionally, Byrne's similarity attraction theory [10] posits that humans are attracted to people who are similar to themselves. Social categories are defined by the salient shared characteristics, and self-categorization inherently draws upon the notion that one does (in-group) or does not (out-group) perceive themselves as a member of a social group, with varying levels of association [101]. It's important to note that social categories are established, in contrast to other social categories within our societies, and we are therefore born into them [38].

Thatcher and Patel note, in their review of faultlines literature that "self-categorization and social identity theories explain why individuals classify themselves and other team members based on salient characteristics. The similarity-attraction paradigm explains why individuals are likely to align with similar individuals resulting in subgroup formation" [97].

Researchers have pursued and studied the effects of faultlines to test Lau and Murnighan's assumptions. In Thatcher and Patel's review of 59 published

articles on faultlines, they found support for the claim that strong faultlines negatively influence group performance, group satisfaction, group learning, and knowledge exchange, to name a few [97]. Thatcher and Patel also note that most studies of diversity and faultlines control for group size and that one study found that in very large groups "it is very unlikely that subgroups can be homogenous across multiple attributes" [97]. They conducted a study that confirmed that very large projects will likely not have strong faultlines [95]. Both Lau and Murnighan also propose that very large groups are unlikely to have strong faultlines. While this may seem to disqualify FOSS communities, it's important to note that on average, most projects are small [53].

Thatcher and Patel note that "Faultlines are either empirically inferred or created in lab settings" [97]. Faultline measurement, particularly faultline strength, is most commonly calculated statistically using a multivariate clustering approach found in Thatcher et al. [96], while recent empirical measures include variance decomposition approaches and latent class analysis [97]. In this paper, we can only posit that faultlines are a strong explanation for a lack of diversity in FOSS communities based on existing survey data. Also, application of faultline theory to FOSS hinges on whether or not group members can perceive similarities and differences in demographic characteristics.

FOSS communities are virtual communities that share many of the characteristics of communities of practice (CoP). First defined by Lave and Wenger [61], a community of practice is a community characterized by a shared domain of interest, whose members regularly interact, using shared resources, tools, and methods of problem solving. Wenger also notes that "communities of practice are groups of people who share a concern or a passion for something

they do and learn how to do it better as they interact regularly" [106]. The FOSS community has formed as a network of people who share similar interests and commitments to developing FOSS, engaging in active discussion to share information and learn [57]. Similarly, just as IRC is an accepted tool for communication within FOSS communities, FOSS practitioners also share other tools, methodologies, ideologies, and experiences. Most, if not all, collaboration is coordinated through CMC.

In a comparison of CMC and face-to-face (f2f) communication, Etzioni notes that CMC is superior in reaching more people over large distance, but less secure in presenting accurate interpersonal knowledge, noting that "identity is rarely authenticated" [23]. CMC is inherently abstracted from physical space, but the identities of communicators are still negotiated and inferred [20], even though physical proximity and non-verbal cues do not exist [16]. Tidwell and Walther evaluated how computer-mediated communicators develop impressions of interpersonal characteristics through self-disclosure, question-asking, and uncertainty reduction [94] as a response to the assertion of past CMC research that claims lack of nonverbal cues hinder personal impressions. They conclude that "CMC does not appear to hinder uncertainty reduction, merely perceptions of ability to do so" and that "CMC interactants work to overcome limitations of the channel as they get to know one another" [1].

Since identity is socially constructed and elements of identity can be perceived in the medium of CMC and self-categorization occurs, it is logical that the perception of demographic attributes can be inferred in a virtual community whether it's explicitly shared or observationally inferred.

While group faultline theory is only one explanation of workgroup conflict, it is suitable in the study of FOSS communities for several reasons. First, our primary source of information about FOSS developers comes in the form of demographic information, which is the basis of group faultlines theory. Second, identity is still perceived in an online space, and therefore faultlines still form. Third, faultline theory is an alignment theory, meaning that it is uniquely equipped to deal with effects of multiple demographic attributes, rather than the effect of an individual attribute. Fourth, and perhaps most importantly, faultlines theory can help reveal conflict areas within FOSS that may be less obvious to the community. This is done by exploring faultlines activation, represented by diversity conflict within the community.

4.3.2

Peter C. Jackson notes that "When systems practitioners bring together various systems ideas and techniques in an organized way and employ them to try to improve a problem situation, they are said to be using a 'system methodology' […]" [42]. In this paper, we will be borrowing concepts from a variety of system methodologies and tools to highlight the complex nature of diversity conflict in FOSS communities. This is known as Critical Systems Thinking (CST). All systems methodologies have limitations, many of which include a lack of social context [41], which is what gave rise to CST. Below we highlight the logical progression of systems methodologies that led to CST, as well as provide the foundation for future work.

Although it can trace it's lineage back to Aristotle, systems thinking originated in the 1940s and 1950s, with a key publication on "cybernetics" by Norbet Wiener [109] and research by von Bertalanffy [103] Boulding [8], and

Bogdanov [7]. These laid the groundwork for systems thinking as we know it today. Although much can be said about the development and progress of the systems movement across a range of disciplines, for this paper we will focus on its application to sociological and organizational management.

Jackson defines a system as "a complex whole the functioning of which depends on its parts and the interactions between those parts" [43]. Therefore, "systems thinking" is a problem-solving method for understanding the relationships of components within a system, be it biological, sociological, mechanical or otherwise. Systems can be studied through a lens of reductionism—interpreting systems as the sum of their parts; or holism—interpreting systems as more than the sum of their parts [43]. Jackson notes that "holism gained a foothold in many different academic disciplines[9], benefitting from the failure of reductionism to cope with problems of complexity, diversity and change in complex systems" [43]. Systems are also generally characterized as either hard—having a definable problem with clear inputs and outputs; or soft—difficult to define, with less tangible factors (cultural, philosophical, etc.). Social and cultural systems are "soft systems".

In response to this, Peter Checkland designed the Software Systems Methodology (SSM) to specifically address real-world, ill-defined, fuzzy problems [11]. SSM is a seven-step process that helps contextualize a problem, define the relevant systems, conceptualize the activities needed to realize the nature of the problem, and finally develop actions to improve the problem situation [11].

---

[9] For an in-depth review of the history of systems thinking, see Jackson [41].

SSM contains several key methods to help characterize soft problems. First and foremost is the mnemonic CATAWOE, which is used to create a "root definition" which defines a human activity system (i.e., "holon" in systems language) that is relevant to the problem situation (see Table 7). This is important because all problem situations feature "human beings in social roles trying to take purposeful action" [12]. All relevant systems should be identified and considered when trying to understand a problem-context and create an initiative to improve it.

Table 14. CATWOE as a Representation of FOSS Development [11]

Mnemonic		**Explanation** [12]	**FOSS Development Community**
C	Customers	Victims or beneficiaries of T	FOSS developers and users
A	Actors	Those who carry out the activities of T	FOSS developers and contributors
T	Transformation	Transformation of some entity into a changed form of that entity, conversion of input and output	Write software code to create a usable, effective FOSS software
W	*Weltanschauung*	The "*worldview*" that makes T meaningful in context	Community-developed software can create efficient, usable and cost-effective software
O	Owner	Those who can stop T	FOSS developers and contributors
E	Environment	Elements outside the system which it takes as given	Open development model and license, accepted collaborative tools and methods

Each of these components helps define the assumptions made about the holon. There are many holons that are relevant to the problem situation, so a root definition will change when CATWOE components change, especially "W" and "T" since they describe the purpose and value of the system.

Below is an example of a simple root definition of the FOSS community as a holon for developing software (generated from Table 14):

> *A developer- and user-owned system to create software, by employing open development practices and accepted collaborative tools, leveraging community participation, in order to create efficient, usable and cost-effective software.*

Checkland notes that "no human activity system is *intrinsically* relevant to any problem situation; the choice is always subjective" [12]. In the root definition example above, we can assume that FOSS development as a system plays a large role in how the community operates, and therefore is relevant to a problem like diversity, at least in terms of the assumption that diversity is a systemic problem. While CATWOE is only one component of Checkland's SSM, there are many other tools and methodologies. To address this issue Jackson and Keys created the System of Systems Methodology (SOSM) to "relate different systems methodologies to each other on the basis of the assumptions they made about the nature of problem situations or 'problem-contexts'" [41].

Jackson and Keys' SOSM is a continuum that considers the complexity of problems. Problems can be simple—limited number of elements and interactions, and governed by "well understood laws of behavior" or complex—large

number of highly interactive elements, which are open to "turbulent environments" [41]. SOSM also considers the relationship of the values and interests of the parties involved as they relate to the problem situation, which are characterized as *unitary*, *pluralist*, or *conflictual/coercive*. Unitary relationships exist when the parties involved are in "genuine agreement about their objectives" and have shared values and interests. Pluralist relationships exist when parties involved have divergent values and interests, but "share enough in common so that it is worthwhile" to remain members of a group that make up the system. Relationships can be conflictual or coercive "if their interests diverge irreconcilably and power comes to bear so that some group or groups gets its own way at the expense of those who are coerced" [41].

With this in mind, we will characterize the problem context of diversity in FOSS as "complex" and "coercive." We support this claim with evidence from our discussion section. In terms of understanding diversity, the FOSS community is a mix of varying perspectives that deny the existence of a diversity problem, realize that it exists, or accept it while maintaining a passive stance. These differing perspectives are manifested in FOSS community member comments like: "I'm all for removing (artificial) barriers, but I still don't see the reason for having more women in FOSS."[10] Matt Zimmerman, a technical leader in the Ubuntu community, posted on an Ubuntu forum suggesting they adopt a diversity statement [114]. While many people agreed, some were passive "I like your intentions on this, but I think it's rather naïve to believe that such a statement would improve the kind of chronically socially inept people that

[10] https://plus.google.com/+gnome/posts/jUSheJrbttC

typically cause those kind of problems." There are many other instances where overt initiatives to improve diversity have begun heated arguments on how to exactly fix the problem of the diversity.

When a problem context is coercive, there are factors that coerce the perceptions of the problem. Because diversity is a social issue, there are many external factors that cannot be controlled or defined, which attest to its complexity. Jackson and Keys note that in complex systems, "not all of the attributes of the parts of the system will be directly observable; actions within the system will be probabilistic in nature, and systems will evolve over time [43]. The SOSM shows us that problem contexts are different in meaningful ways, which implies that problem contexts require different problem-solving methodologies [41].

Coercive problem contexts require an *emancipatory systems approach* which "is suspicious of the current social order and seek to radically reform it. They see society, as presently constituted, as benefitting some groups at the expense of other groups which are suggesting domination or discrimination" [41]. Emancipatory systems approaches include Beer's Team Syntegrity approach [3] and Ulrich's Critical Systems Heuristics [102]. In this paper we are not trying to solve the diversity problem using an emancipatory systems approach. Instead, we are highlighting its complex and systemic nature.

In this paper, we use elements of systems thinking to examine the relevant systems within FOSS that can influence perceptions of diversity. The benefits of this approach are threefold. Systems thinking can explain the inner workings of how faultlines are activated, and cause conflict. Systems thinking can help characterize the perceptions and values of members of FOSS community in

relation to diversity. Systems thinking can also conceptualize diversity conflict as the interaction of different subsystems (e.g., ideological, methodological, etc.). Checkland reinforces this when he says: "[…] any actual social system observed in the world will be a mixture of a rational assembly of linked activities (i.e., holons) and a set of relationships such as occur in a community (i.e., a natural system)" [11].

4.4 Faultlines in the FOSS Community

As noted above, faultlines can be inferred empirically or created in lab settings. In Lau and Murnighan's original presentation of faultlines, they note that "group faultlines increase in strength as more attributes are highly correlated, reducing the number and increasing the homogeneity of the resulting subgroups" [59]. This means that when a subgroup is formed, faultlines exist at the boundaries of the subgroup. Lau and Murnighan also demonstrate the variable strength of faultlines by creating hypothetical, archetypal groups composed of members with varying degrees of demographic diversity. Based on the perception of differences between the hypothetical groups, they were able to estimate the faultline strength existing in each group. For this reason, we can suggest that faultlines exist where there is a skewed polarity in the community in terms of demographic attributes.

4.4.1 Faultlines Inferred

As noted in Section 2, it is plain to see that the FOSS community is demographically homogenous. Although we could suggest all potential subgroups that exist in the community, it only seems reasonable to focus on the

demographic attributes that have been verified by multiple surveys: gender, tenure, and education.

- 98% of the FOSS community is male (gender)
- 76% of the FOSS community works in IT (tenure).
- 55% of the community has at least a Bachelor's degree (education).

By multiplying these percentages, we can also estimate the probability of different demographic attributes aligning. For example, there is a 74% chance that a FOSS community member will be male and work in IT. There is also a 40% chance that a community member will be male, work in IT and have achieved at least a bachelor's degree.

Although there is less triangulation, we can also see that:

- 80% of developers joined the FOSS community to learn and developer new skills
- 68% of developers wanted to become better programmers
- 77% of developers considered it important to give back to the community after using FOSS.

There are some important limitations to our ability to determine potential faultlines in the community. First, the current large-scale FOSS developer surveys can only provide limited insight into the demographic landscape of FOSS communities, in part due to outdated data and inconsistent sampling. Second, using probability is a potential way to infer the demographic characteristics of subgroups. This method of inferring faultlines assumes that social identity theory and similarity-attraction theory will create subgroups of

demographically related members 100% of the time. We know this is not true, because not all demographic characteristics are salient all of the time, let alone in a virtual community.

We can use these percentages as a guide to estimate where faultlines could form or currently exist. In this regard, we are not trying to prove anything except to suggest that faultlines exist in FOSS communities and can explain its current homogeneity.

4.4.2 Faultlines Discussed

From the list of demographic characteristics listed above, we see that gender stands out as the most evident and corroborated faultline. In this section we will discuss the gender faultline, and evidence of activation in FOSS communities. Michael Schwern, an open source developer, notes in a keynote presentation at YAPC::NA[11] that "[…] demographic diversity is the canary in the coal mine. Gender is simply easy to track. It's the most obvious thing in front of our face." He also notes that "If gender is out of whack, if demographic diversity is out of whack, then there's a good chance that other things are as well" [84]. In other words, although gender is a salient issue in FOSS, it is only the most visible of the issues facing this community.

Unsurprisingly, over the last ten years, gender has become a hot-button issue in the FOSS community, which suggests that gender differences have become more salient and the faultline has been activated and has let to events and conflict within the community. Before we begin to look at potential signs of gender faultline activation, it's important to point out that faultline triggers are

---

[11] "Yet Another Perl Conference, North America"

experiential and not necessarily easy to pin down. As an example, one study "found that faultlines based on location were triggered in virtual teams when team members tried to find a common time for a chat meeting" [97].

Chrobot-Mason et al. posit that faultline triggers take the form of one of the following: differential treatment, different values, assimilation, insult or humiliative action, or simple contact [97]. An example of each one of these triggers has occurred in the FOSS community and GeekFeminism, a blog and wiki that "exists to support encourage, and discuss issues facing women in geek communities" [6] has recorded some of the more public gender-related conflicts. Examples include advertisements relating reliability to sex acts with women; FOSS conference presentations where women are depicted in humiliating ways; presenters who have made sexist remarks; or women who have been sexually harassed. Although these are rare events, they are very public, and many women experience FOSS communities as hostile on a daily basis with flaming or crude humor in mailing lists and IRC channels [68], [49].

Kuechler et al. found that participation of women in FOSS mailing lists declined at a disproportional rate compared to men over time, and that this problem "is likely not technical, because most women drop out of FOSS early in their project membership" [55]. Women make up a very small minority of all contributors to FOSS, and when an action takes place that brings to light this gender disparity, it is not hard to see that women feel out of place, or as Kirrily Roberts notes in a keynote talk, it feels like "walking into a sports bar on game night wearing the wrong team's jersey. It can be the most friendly sports bar in the universe, but you're still going to feel pretty awkward" [79].

In support of this, Lau and Murnighan posit that:

> “Relative subgroup sizes and/or a disparity in subgroup power also can generate a variety of group dynamics. The presence of few subgroup members suggests a lack of social power and less internal support, leading to reduced confidence and less frequent opinion voicing. When members of less powerful subgroups do voice their opinions, they are likely to experience more opinion suppression than the members of more powerful subgroups. The suppressing or hiding of minority opinions makes the group process seem smooth, since overt conflict is kept to a minimum” [59].

They also note that:

> “[…] a single newcomer tends to experience strong pressures to conform unless he or she shares some demographic similarities with at least one existing group member. Thus, the first member of a noticeable societal minority who enters an established group of similar societal majority members may not be in a strong position to alter existing group norms” [45].

Unsurprisingly, some in the FOSS community are reluctant to admit that a gender problem exists. To illustrate this we can look to Nafus et al. [68] who found that "66% of men and 85% of women agreed that more female participants would be better for the whole F/LOSS community." The same study found that most men "see themselves as neither sexist nor hostile towards women." By contrast, "almost all women have observed or experienced discriminatory behavior against themselves or other women by men in the general F/LOSS community, but only a third of all men reported to have perceived discriminatory behavior against women" [68]. Interestingly, one study finds that 72% of female respondents said “they felt outnumbered but only 24% said that caused them to feel alienated” [73]. These results show that even when

demographic differences are salient, the faultlines that are formed do not always guarantee conflict will occur, but may have other effects, like alienation.

Initiatives to improve diversity and bring accountability to the community may also be indicative of the activation of gender faultlines. Women in FOSS created "chix groups" to build peer networks and provide mentorship for women, starting with LinuxChix[12] in 1998. They replicated this approach across many communities, including the DebianWomen[13] group, which maintains a webpage with access to mentors;[14] Dreamwidth, where 75% of contributors in 2008 were female [79]; ChickTech, "an organization dedicated to retaining women in the technology workforce;"[15] the ADA Initiative, a "nonprofit organization supporting women in open technology and culture;"[16] GitHub, that recently donated a private repository to women learning FOSS [32], and Railsbridge, an organization that puts on free workshops and events to promote diversity in technology.[17]

The success of these initiatives is in part attributed to the fact that they were created to promote gender diversity. Developers were recruited based on the needs of the project rather than FOSS experience, which many lacked at the time. Language and tools were chosen for their ease of use, and an environment was designed to be supportive and focused on teaching rather than competing [79]. This is different from standard FOSS projects. In an effort to recruit more women, many projects now have diversity statements, including Dreamwidth

---

[12] http://www.linuxchix.org/
[13] http://www.dreamwidth.org/
[14] http://www.debian.org/women/
[15] http://www.chicktech.org/
[16] http://adainitiative.org/
[17] http://workshops.railsbridge.org/

(2009), Python (2009), Plone (2010), Ubuntu (2011), Debian (2012), and Padre (2012). While more research ad initiatives are needed, some efforts appear to be having an effect.

In this section we have argued that faultlines have been activated due to the lack of diversity in FOSS. This is important for future research on FOSS diversity conflict in several ways. Faultline activation shows that demographic differences are affecting the FOSS community. In more specific terms, activation shows that subgroup formation and inter-subgroup dynamics within a project are bringing to light demographic differences. This can help researchers and community members identify the roots of conflict within the community and guide initiatives. Faultlines also suggest that conflict resolution is not as simple as identifying problem members of the community who are calling attention to differences between members, but instead, directing attention to factors that have shaped the status quo and current homogeneity of the community. In this regard, subgroups will only change if the demographic composition of the community changes. Future research will need to look at the specific subgroup formation that has occurred in FOSS, since demographic data can only suggest the strength of faultlines, as Lau and Murnighan did [59]. Since there are many different demographic attributes within the FOSS community, there is little doubt that other faultlines exist.

4.5 A Critical Systems Approach to Understanding Diversity Conflict in FOSS

"They don't know it, but they are doing it" [21] -Karl Marx

Systems thinking picks up where faultlines leaves off by explaining the forces and dynamics that contribute to conflict materialization and faultline activation. The FOSS community, as a cultural system, is composed of at least two subsystems: a system of ideology, and a system of methodology. There are also other structural characteristics of the FOSS cultural system that affect the way the community operates. These include: high turnover of developers, reliance on CMC, geographic separation, and language and cultural barriers (see Figure 10). We draw a distinction between coercive factors and structural characteristics. Coercive factors influence how community members perceive the problem-context, while structural characteristics are attributes of the community that can create and exacerbate conflict.

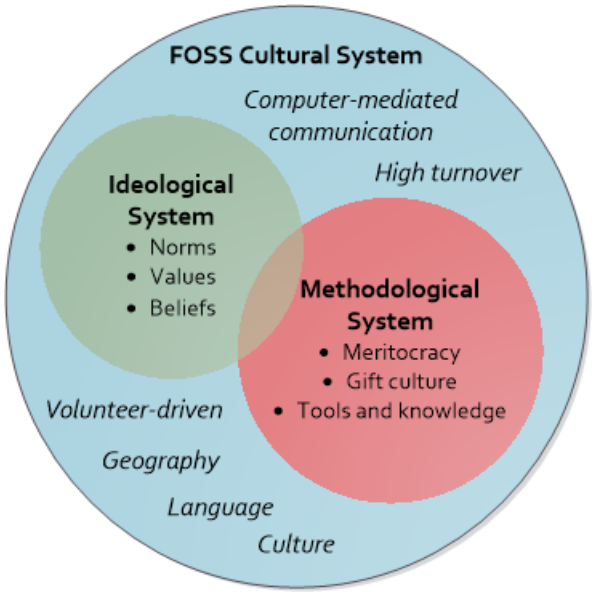

Figure 10. FOSS Cultural System and its Subsystems

4.5.1 Coercive Factors

4.5.1.1 A System of Ideology

An ideology is the "body of doctrine, myth, belief, etc., that guides an individual, social movement, institution, class or large group" [39]. An ideology therefore plays a central part in how people think and behave. Although defining an ideology is no simple matter, it can be done by examining the values, norms, and beliefs of the FOSS community.

Stewart and Gosain found that some ideological components enhance trust and communication within teams, which increased team effectiveness [91]. They promote the idea that ideology shapes behavior and that behaviors that align with the community's accepted norms and values will "enhance cognitive trust, communication quality, and encourage identification with the project team" [91]. For the sake of simplicity we have compiled a list of some of the norms, values, and beliefs that exist in FOSS (see Table 15).

Table 15. FOSS Norms, Values, and Beliefs

Norms	Values	Beliefs
Forking is frowned upon [91]	"Sharing information is important" [91]	Code quality is better when using FOSS development methods [91]
Distribution of software code changes should use the proper channels [91]	Helping others is important [91]	"Outcomes are better when code is freely available" [91]
Named credit should not be removed without the author's consent [91]	"Technical knowledge is highly valued" [91]	"Outcomes are better when information is freely available" [91]
Ownership is acquired by initiating a project, having it publically bequeathed to you, or by reviving a "dead" project [105]	"There is a value on learning for its own sake" [91]	"The more people working on the code, the more quickly bugs will be found and fixed" [91]
"Authority derives from responsibility" [105]	"Voluntary cooperation is important" [91]	"Practical work is more useful that theoretical discussion" [91]

"Seniority rules in disagreements over who has responsibility for a piece of code" [105]	Reputation gained by participation in FOSS is valuable [91]	"Status is achieved through community recognition" [91[
"[…] Disputes can and should be resolved 'objectively' […] the best dispute resolution mechanism is simply to 'let the code decide." [105]		"FOSS is a philosophy and way of life" [64]
		Proprietary software is theft and hoarding [64]

FOSS grew out of a male-defined, academic and corporate etiquette from the 1960s. The ideals and ethos of FOSS is still very much anchored in this context. FOSS is also strongly situated in not only a social movement of software development (i.e., open collaboration, rebellion against proprietary control), but also a political movement that emphasizes the intellectual rights of software developers and their freedom to choose how they use and share software (e.g., anti-authoritarianism). Although a correlation between ideology and the effectiveness of FOSS teams has been established, the FOSS ideology could also affect other elements of FOSS, including perceptions of diversity.

4.5.1.2 A System of Methodology

Next, we discuss the methodological subsystem that operates in FOSS as a combination of meritocracy, gift culture, and tools and knowledge. As we noted in our introduction of FOSS, meritocracy is the core of many FOSS projects. We have noted that FOSS projects are structured like an onion model [113] or onion patch model [51]. The transition between roles within this structure predicates on a contributor's status and reputation within the project. As examples, the project initiator has immediate reputation within the project because he started it, the

core contributors have earned their place through submitting code pastches, actively participating in the community mailing list, helping newcomers, etc.

Raymond [75] and Rheingold [78] characterized FOSS as a gift culture, characterized by free sharing of information, ideas, advice, source code, etc. Berquist and Ljungberg point out that there are nuanced rules that must be followed for the "gift exchange" to be successful [4]. For example, a person requesting help must be perceived as an "earnest enquirer" and demonstrate a measure of "netiquette" [4]. They also note that "in order not to get flamed, inexperienced users often headline their question 'newbie' followed by a topic that is developed further in the message. The norms and values of the community are internalized through these kind of narrative practices" [4]. Humility is also used as "a strategy to get acceptance from the community" [4]. Berquiest and Ljungberg note that "Practices such as flaming are not only used to exclude people who overrule the core values of the community," but also "flaming newbies happens so newbies can learn how to flame" [4]. Flaming can be a consequence of improper etiquette or substandard contributions, which suggests that only gifts that the community appreciates results in increased status within the community.

The gift culture also forms power relationships between givers and receivers: "virtual community gift giving is managed through acknowledgement: the giver is 'paid' by the community by receiving a certain amount of fame and respect" [4]. Through this socialization process, social structure and hierarchy is imposed on the community and only a select few achieve high rank.

While the gift culture influences the community, the community also borrows from outside of the community. Most people are shaped by the behavior

they found successful in the past. 76% of the FOSS community works in IT, and 55% of contributors have at least a Bachelor's degree. This presumably includes training in coding practices, data structures, algorithms, theories of computation, or software testing. In this respect, IT workers will be familiar with working with like-minded and similarly skilled people, which creates efficiency. Working with interdisciplinary teams has been shown to lead to friction [71].

4.5.2 Structural Characteristics

4.5.2.1 Volunteer-driven

Since participating in FOSS is voluntary, many researchers have sought to understand the motivations that influence participation (see Section 2.4.2.4). Clary and Snyder believe that voluntary participation, in general, can be attributed to: acting upon a personal value, seeking social interaction, searching to understand the world or exercise skills, or attaining career-related experience [15]. We can see the evident parallel to FOSS developer motivations: to benefit people through software, shared knowledge and skills, and offering alternatives to proprietary software [56].

Volunteering can lead to ethnographic differences. The U.S. Bureau of Labor Statistics shows that the average number hours spent volunteering in 2012 differed across ethnicity, age, marital status, employment status, etc. One U.S. study examined differences on volunteering between native-born and immigrant African Americans, Asians, Hispanics and Whites. In a survey of 59,888 respondents, Sundeen et al. found that different ethnic groups are driven by different individual and social factors [92]. The National Centre for Social

Research and the Institute for Volunteering Research, in a study of nonprofit volunteering in England between 2005 and 2006 [66], found that groups deemed at risk of social exclusion tend to volunteer less than others (32% to the average 42%). These groups included minority ethnic groups, individuals with disabilities, no qualifications, and long-term illnesses [66].

4.5.2.2 High Turnover of Contributors

High turnover is potentially a symptom of diversity conflict and an explanation for it. The Pareto principle states that 80% of work is produced by only 20% of the workforce.[18] If we assume the Pareto Principle holds true in FOSS, we can assume that the small minority of core developers do most of the work. Seaman and Basili also found that when team member in a collocated software development were not in regular contact with each other, it often increased meetings time due to rehashing global issues that had already reached previous resolution [83]. This could also lead to impatience and frustration. Many FOSS communities have come to expect high turnover [70]. This influences how current developers interact with new people interested in contributing. There may be unwillingness to invest time and effort in mentoring. Without interactions between novices and experts, "newbies" may not be motivated to contribute. Surveys have found that communication volume between developers is high, though it occurs between small groups of developers [28]. For new contributors, it may be hard to break into these tight-knit social networks.

---

[18] http://www.investopedia.com/terms/1/80-20-rule.asp

4.5.2.3 Computer-Mediated Communication

CMC is the norm in FOSS, with text-based IRC and mailing lists taking the place of face-to-face interactions. Sproul and Kiesler show that 93% of intent is articulated through tone and facial expression [88]: “When technological change creates new social situations, traditional expectation and norms lose their power” [88]. Sproul and Kiesler also explain that “when communication lacks the dynamic personal information of face-to-face communication or even of telephone communication, people focus their attention more on the words in the message than on each other” [88].

Private discussions occur in FOSS, but often time mailing lists are used, which broadcast to anyone subscribed to them. Similarly, IRC is a public forum and can be read by people not participating in the conversation. Since text-based communication loses many of the nuanced cues of face-to-face interaction, it is no surprise if ambiguities and miscommunication occur, which could offend or upset individuals. This is especially true for newcomers who may not know or understand the norms of the community.

Jessup et al. found that: "de-individuation occurs when social cues that distinguish individuals are missing. The effect on people is less of a sense of individuality among group participants, detaching individuals from his or her comments that can lead to a reduction in normal restraints on behavior" [52]. This de-individuation may enable the negative behavior we see in FOSS. One study found that e-mail reduces social cues and empowers people to behave more aggressively [89].

4.5.2.4 Geography, Language, and Culture

Language is an important barrier to diversity. FOSS is not restricted by geography because of the online nature of projects. However, English is the "working language of hacker culture (FOSS)" [76]. David et al. find that 95% of FOSS developers speak English [17], and Raymond notes that "If your writing is semi-literate, ungrammatical, and riddled with misspellings, many hackers (including myself) will tend to ignore you. While sloppy writing does not invariably mean sloppy thinking, we've generally found the correlation to be strong—and we have no use for sloppy thinkers" [76]. While FOSS participation is open to anyone anywhere, knowledge of English is determined by geography, culture, and access to education. While a number of large projects translate their interfaces, coverage is irregular and requires extra effort. The code however is still commented in English, and documentation and discussions often default to English. Even though English is a widely used language, for the majority of the world it is a second language. According to Internet World Stat, roughly 536 million people speak English, or 27% of Internet users [99].

There are important FOSS projects not anchored in the west. Some examples include Ruby (Japan), Sahana (Sri Lanka), and Ushahidi (Kenya). These projects default to English for communication, coordination, and documentation. Other projects, such as Ubuntu and the One Laptop per Child project, started in the west, and though they draw on western developers, they have local teams across the world. These teams often work using their local languages. Finally there are a growing number of national FOSS repositories, many hosting regional versions of FOSS projects, largely isolated from their foreign language brethren.

Language and geographic diversity come at a cost, as teams must work harder to reach understanding and agreement. Research has shown that the ability to negotiate understanding, reach consensus, and establish trust across cultural barriers is difficult. As Olsen and Olsen report [71], even when everyone speaks English, communicating across cultures over a narrow channel such as email and IRC can be fraught with problems. Beyond a lack of cues (e.g., body language, etc.), time zones, mental models, and geography all affect coordination [22]. Collaboration across time zones can be difficult, which explains why many software developers work best with others in adjacent time zones [87].

Hofstede showed that cultures approach social interactions and conflict is in accordance with cultural dimensions, including *power distance*: how cultures cope with inequality; *uncertainty avoidance:* how cultures cope with uncertainty; *individualism versus collectivism*: the relationship of the individual with her or his primary group; and *masculinity versus femininity:* the emotional implications of being born of a particular gender [37]. Since FOSS development is anchored in the United States and Western Europe, low- and high-power distance conflicts are particularly interesting. These occur when cultures with greater respect for hierarchy such as Brazil, China, India, and Singapore have to work with cultures with less respect for hierarchy, such as the United States. Gibbs recognizes this issue and notes that "New ways of understanding and managing cultural differences are needed to explain the complex culture of global teams and enable productive collaboration" [31].

4.5.3 Systems Thinking Discussed

As we noted in Section 3.2, different systems methodologies exist, and their applicability hinges on the problem context. We have presented the lack of

diversity in FOSS communities as a coercive problem context, according to the SOSM [41]. This characterization assumes that coercive factors exist within the FOSS community that shapes how people perceive the diversity problem. We have already discussed the implications of several of the perceivable structural characteristics within FOSS's cultural system, but we believe that a system of ideology and methodology are coercive factors within the community, but we do not fully understand how they affect diversity.

For example, we have seen that ideology is positively correlated with team effectiveness [91]. Because ideology is strongly situated in the heart of FOSS, there is no doubt that it shapes perceptions and behaviors. Since we know this to be true, we can link ideology to motivations, which research suggests is the crux of FOSS participation. At a very high level, we can simply suggest that a person's alignment with this FOSS ideology may motivate or discourage participation. Similarly, if one closely aligns with this FOSS ideology, one may be less inclined to agree with different or conflicting norms, values, and behaviors. In this regard, newcomers must learn and adapt to the FOSS ideology in order to gain acceptance. We could argue that when one agrees with and accepts this ideology, one has successfully assimilated into the culture. When one does not, one is publically flamed in an attempt to force assimilation or abandonment. The established socialization of FOSS newcomers attests to this [18].

Similarly, current development methods adopted by the community play a coercive role in the perceptions surrounding diversity. We can see by the implementation of the "default" meritocratic model of FOSS that the value of a contributor is equal to the quantity and quality of their code. Thus, reputation is purely based on ability. One could argue that one's ability to develop FOSS is a

product of both knowledge and familiarity with current development tools, cultural norms, or language. Therefore, newcomers are inherently placed in a disadvantageous position, and those who do not share the core community's background and experience doubly so. This is especially true for those who do not have an underlying knowledge of computing (i.e., anyone who does not work in IT or is not formally educated in computer science), a proficiency with English, as examples.

We can also point to another feature of the FOSS model that possibly affects diversity conflict. The simple fact that the adage "shut up and code" exists in FOSS suggests that writing code is more important than resolving conflict, at least initially. This is to say that "squabbling" over issues that are not directly related to furthering the project, are not valued. Conflicts can also arise during problem solving, design, and task prioritizing, and different demographic backgrounds can lead to different conclusions or recommendations. As we discussed, women tend to work on issues that do not garner as much social capital [68]. Therefore, fostering diversity will require the FOSS community to see worth differently, looking beyond programming abilities.

We can only make guesses about the effect these factors have on diversity, and further in-depth studies, using systems methodologies, should be conducted to corroborate our assumptions. This research should discover all of the systems that are relevant to the problem context of diversity in FOSS communities, and the effects of these systems, including ideology and methodology. Qualitative information will need to be gathered through interviews, observations, and data mining over a wide variety of FOSS projects.

4.6 Systems Thinking and Faultlines

Thus far we have presented diversity conflict as a product of faultlines and subgroup formation, and a cultural system with several unique structural characteristics and two distinct subsystems. Both faultlines theory and systems thinking expose new directions to diversity research in FOSS, and broaden the scope of potential new initiatives designed to understand, improve, mediate, and prevent diversity conflict. This is important because recent efforts to improve diversity have focused on increasing the populations of minority groups (e.g., gender and ethnic minorities) in FOSS, but this only addresses the symptom, not the root cause.

Since all people are the product of their environment, it's not hard to see that one dynamic can have many effects. For example, we can see from our discussion of socialization and the act of flaming that the intent behind some flaming is not inherently bad, but instead a didactic tool used to teach newcomers the norms and values of the community. But, since this is usually done on project mailing lists, it becomes public to its current subscribers. Since text-based CMC lacks much of the intent, only members of the community will understand what is occurring, while an outsider may see this as harsh criticism, sexism, racism and so forth. This may explain why FOSS has brittle communities. Researchers and the community have failed to see that all conflict and events that occur in the FOSS community are products of its social, cultural, and demographic structure.

To expand on this, we can see through faultline theory that subgroups have undoubtedly formed, many of which were established early in the life of the FOSS movement [74]. Lau and Murnighan also note that subgroup structure

is difficult to change once it has been established [59]. This, in itself, supports the idea that FOSS will continue to maintain its homogeneity. In addition, when considering the strong ideology of FOSS coupled with its accepted methods, you will often find that newcomers who do not agree with the status quo will only have one choice with three consequences: accept the new values and methods (assimilate), reject them and leave, or attempt to bring new ideas and methods into the community and be met with flaming and community resistance.

In the previous section, we oversimplify the problem by suggesting that only two distinct ideologies and methodologies exist (newcomer vs. established culture), but we must also address the concept that FOSS is a decentralized and independent structure. Although FOSS projects are all united under the banner of FOSS, which includes a common licensing and development model, most projects are independent and likely have their own culture: "Just as the ideology of an organizational culture is influenced by the larger national culture in which it exists, organizational subunits may have their own unique subcultures, varying in their ideological content" [91]. Hofstede's power-distance may explain why minority members of the predominant subgroup remain silent to avoid disrupting the community.

As researchers and the FOSS community pursue change toward increased diversity, many nuanced variables will need to be addressed. Motivations shape your willingness to accept or resist change. If you believe that FOSS is a technical movement that pays homage to an early hacker dynasty, then you may be more inclined to support a community that represents those notions. For change to occur, current developers must be motivated to work toward diversity and

persevere through challenges that come with implementing diversity. In organizations with paid employees, there are monetary incentives to accept change. In FOSS, incentives are not tied to financial gain, making it difficult to motivate contributors to accept new ways of working.

When attempting implement diversity change, whether through addressing a systems issue, as we have argued, or by simply adding new members of differing demographic backgrounds, conflict will still arise with varied and uncontrollable effects that can be both positive and negative (See Figure Z). Pulling from Wolstenholme's System Enquiry methodology [110], we can visualize the effects of conflict using a modified system dynamics diagram (see Figure 11).

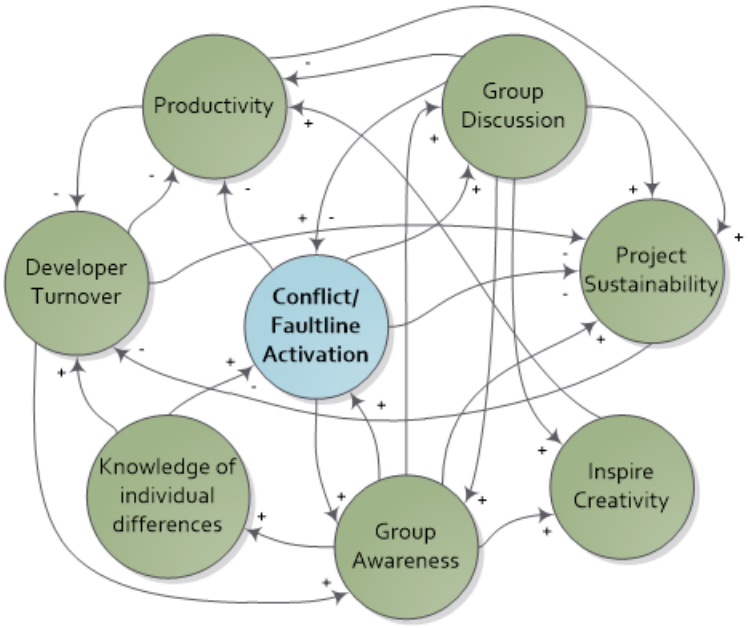

Figure 11. Dynamic Effects of FOSS Conflict

In light of Figure 11, it seems reasonable to suggest that even when an initiative is in place to improve diversity, the effects could actually increase

conflict. For example, when we try to increase the number of community members with minority demographic attributes within the FOSS community (e.g., non-male, and non-white), we could actually be adding fuel to the diversity fire. Why? Because increasing the number of members who are different from the majority subgroup within the community could inherently increase the number of conflicts. We cannot prove this in this paper, but if diversity conflict is the product of faultline activation, then we will see more conflicts arise as minority subgroup sizes increase.

We would also propose that when conflict arises in the FOSS community, it will require strong mediation because the root of diversity conflict is also dynamic. Our systems thinking explanation for diversity conflict suggests that subsystems and structural characteristics are also playing a role in conflict, not just demographic differences. This means that those involved in conflict are not only negotiating their demographic differences, but also their ideological differences, and so forth. This is to say that conflict can become much more heated, and multifaceted. Although this may be subconscious, many IT conferences (FOSS in particular) have become much more explicit about equality and respect and have implemented the means to control many types of conflict. Interestingly, there is little evidence that suggest this level of mediation has crossed over into the FOSS community in the virtual sense.

In summary, we can see that the use of systems thinking and faultlines theory are mutually beneficial. Faultlines can help us understand diversity conflict as the product of demographic differences and subgroup formation [41], while systems thinking can identify the underlying cultural structure that contributes to the conflict. We have discussed that diversity conflict surrounding

gender has become a hot-button issue in the FOSS community, but this is the proverbial tip of the iceberg. Many other factors are influencing diversity conflict and we must fully understand them before we can create lasting change in the FOSS community.

4.7 Conclusion

While much is still unknown about the diversity problems in FOSS, and while the structure and culture of FOSS presents inherent challenges to increasing diversity, this paper reveals that a) a significant diversity problem exists in FOSS, b) diversity conflict is a systemic issue with many complicating factors, and c) marrying the concept of faultlines and systems thinking will help identify characteristics of the community that help or hinder diversity conflict and propose a method to address and alleviate it.

We believe that the importance of diversity extends beyond FOSS and the communities they serve. Many of the challenges facing diversity in the FOSS community are common to distributed collaboration, as the work of Olsen and Olsen suggests [71]. A richer understanding of the impact of diversity and potential tradeoffs will benefit many information technology workers, where remote collaboration and virtual teams are common.

Acknowledgements

We would like to thank Dr. Javier Calvo-Amodio for his exceptional insight into systems thinking, and Becka Morgan for assisting us on an early draft.

References

1. Ashford, B.E., Mael, F., (1989) “Social Identity Theory and the Organization”, *Academy of Management Review* 14: 20—39.

2. Aurora, V., (2010) “The dark side of open source conferences”. Lwn.net. http://lwn.net/Articles/417952/

3. Beer, S., 1995, *Beyond Dispute: The Invention of Team Syntegrity*. Wiley and Sons, Chichester.

4. Bergquist, M., Ljungberg, J. The power of gifts: organizing social relationships in open source communities. *Information Systems Journal*, 11(4):305–320, 2001. http://innovate.ucsb.edu/wp-content/uploads/2010/06/bergquist-power-of-gifts.pdf

5. Bird, C., Gourley, A., Devanbu, P., Gertz M., Swaminanthan, A., "Mining Email Social Networks." Proc. of the 3rd Int'l Workshop on Mining Software Repositories, pp. 137-143, 2006.

6. Bitzer, J., Schrettl W., and Schroder, P., "Intrinsic Motivation in Open Source Software Development." Journal of Comparative Economics, vol. 35, no. 1, pp. 160-69, 2007.

7. Bogdanov, A., (1984) *Essays in Teknology: The General Science of Organization*. Translated by G. Gorlik, Intersystems Publications, California.

8. Boulding, K. “General Systems Theory: The Skeleton of Science.” *Management Science*, 2, 3, pp.197-208

9. Burnett, M., Wiedenbeck, S., Grigoreanu, V., Subrahmaniyan, N., "Gender in End-User Software Engineering." Proc. 4th Workshop on End-User Software Engineering, 2008

10. Byrne, D. E. 1971. The attraction paradigm. San Diego, CA: Academic Press.

11. Checkland, P. (1981). *Systems thinking, systems practice*. Wiley.

12. Checkland, P., Scholes, J., (1990). *Soft system methodology in action*. Wiley.

13. "Census Bureau Reports Nearly 6 in 10 Advanced Degree Holders Age 25-29 Are Women" (2010) U.S. Department of Commerce. http://www.census.gov/newsroom/releases/archives/education/cb10-55.html

14. Chen, J., Ren Y., Riedl, J., "The effects of diversity on group productivity and member withdrawal in online volunteer groups." Proc. of the 28th Int'l Conf. on Human Factors in Computing Systems, pp. 821-830, 2010.

15. Clary, E. G., & Snyder, M. (1999). The Motivations to Volunteer: Theoretical and Practical Considerations. Current Directions in Psychological Science, 8(5), 156–159.

16. Code, J.R., Zaparyniuk, N.E. (2009). Social Identities, Group Formation, and the Analysis of Online Communities. In S. Hatzipanagos, & S. Warburton (Eds.), *Handbook of Research on Social Software and Developing Community Ontologies* (pp. 86-101). Hershey, PA: Information Science Reference

17. David, P., Watermann, A., Arora, S., "FLOSS-US: The Free/Libre/Open Source Software Sur-vey for 2003." Technical Report. Standford Institute for Economic and Policy Research, Standford, USA, 2003. http://www.stanford.edu/group/floss-us/

18. Decheneaut, N., "Socialization in an Open Source Software Community: A Socio-Technical Analysis". Computer Support Cooperative Work (2005) 14: 323-363. pg. 362

19. Deshpande, A., and Riehle, D., "The Total Growth of Open Source." Proc. 4th Conf. Open Source Systems, pp. 197–209, 2008.

20. Diamandaki, K., (2003). "Virtual ethnicity and digital diasporas: Identity construction in cyberspace." Global Media Journal, vol. 2, issue 2. http://lass.purduecal.edu/cca/gmj/sp03/graduatesp03/gmj-sp03grad-diamandaki.htm

21. Engels, F., Marx, K., Moore, S. (2010). Capital: A Critical Analysis of Capitalist Production.

22. Espinosa, J.A., Kraut, R.E., Slaughter, S.A., Lerch, J.F., Herbsleb, J.D., and Mockus, A, “Shared Mental Models, Familiarity, and Coordination: A Multi-Method Study of Distributed Software Teams”, Proc. of the 23rd Int'l Conf. on Information Systems, pp. 425-433, 2002.

23. Etzioni, A., Etzioni, O., Face-to-Face and Computer-Mediated communities, a comparative analysis. The Information Society, 15(4):241–248, 1999.

24. Fielding, R.T., (1999). Shared leadership in the Apache project. Communications of the ACM 42, 4, 42-43.

25. Free Software Directory. Accessed May 20, 2011. http://directory.fsf.org/

26. Gardler, R., Hanganu, G., (2010) “Meritocratic Governance Model.” http://www.oss-watch.ac.uk/resources/meritocraticGovernanceModel

27. Geek Feminism Wiki. March 2008. Accessed May 20, 2011. http://geekfeminism.wikia.com/wiki/Geek_Feminism_Wiki

28. Ghosh, R.A., Glott, R., Krieger, B., Robles, G., "Free/Libre and Open Source Software: Survey and Study, Part 4: Survey of Developers." June 2002. Accessed May 20, 2011. www.flossproject.org/report/

29. Ghosh, R.A., David P.A., "The nature and composition of the Linux kernel developer community: a dynamic analysis." Stanford Inst. of Economic Policy Research working paper, 2003.

30. Ghosh, R.A., "Understanding Free Software Developers: Findings from the FLOSS Study", in J. Feller, B. Fitzgerald, S. Hissam and K. Lakhani (eds.). Making Sense of the Bazaar: Perspectives on Open Source and Free Software, Cambridge: MIT Press. 2004.

31. Gibbs, J.L., "Culture as Kaleidoscope: Navigating Cultural Tensions in Global Collaboration." Proc. of the 2009 Int'l workshop on Intercultural Collaboration, pp. 89-98, 2009.

32. “GitHub donates private repositories to women learning open source software.” (2013). The Ada Initiative.

http://adainitiative.org/2013/04/github-donates-private-repositories-to-women-learning-open-source-software/

33. Gutwin, C., Penner, R., Chneider, K., "Group Awareness in Distributed Software Develop-ment." Proc. of the 2004 ACM Conf. on Computer Supported Cooperative Work, pp. 72-88, 2004.

34. Hars, A., and Ou, S.S., "Working for free? Motivations for participating in open-source projects," International Journal of Electronic Commerce (6:3), 2002, pp. 25-39.

35. Hemetsberger, A., Reinhardt, C., "Learning and Knowledge-building in Open-source Com-munities: A Social-experiential Approach." Management Learning, vol. 37, pp. 187-214, 2006.

36. Herraiz, I., Robles, G., Amor, J.J., Romera T., Gonzalez Barahona, J.M., "The Process of Joining in Global Distributed Software Projects." Proc. of the Int'l Workshop on Global Software Development for the Practitioner, pp. 27-33, 2006.

37. Hofstede, G.H., Culture's Consequences: International Differences in Work-related Values. Beverly Hills: Sage Publications, 1984.

38. Hogg, M.A., Abrams, D., (1988). “Social Identifications: A Social Psychology of Intergroup Relations and Group Processes”. London: Routledge.

39. ideology. (n.d.). *Dictionary.com Unabridged*. Retrieved May 15, 2013, from Dictionary.com website: http://dictionary.reference.com/browse/ideology

40. "Investment in Open Source Software Set to Rise, Accenture Survey Finds." Online Posting. Au-gust 5, 2010. Accenture. Accessed May 20, 2011. http://newsroom.accenture.com/article_display.cfm?article_id=5045

41. Jackson, M.C. (2000). *Systems approaches to management*. Springer.

42. Jackson, M.C. (2007). *Systems thinking: Creative holism for managers*. Wiley.

43. Jackson, M.C., Keys, P., “Towards a System of Systems Methodology” *The Journal of the Operational Research Society*, Vol. 35, No. 6 (Jun., 1984), pp. 473-486

44. Jackson, S.E., Joshi, A., Erhardt, N.L., "Recent Research on Team and Organizational Diversity: SWOT Analysis and Implications." Journal of Management, vol. 29, no. 6, pp. 801-830, 2003.

45. Jackson, S.E., Stone, V.K., Alvarez, E.B. (1992). Socialization amidst diversity: The impact of demographics on work team oldtimers and newcomers. Research in Organizational Behavior, 15: 45-109. In: Lau, D.C., Murnighan, J.K., “Demographic Diversity and Faultlines: The Compositional Dynamics of Organizational Groups.” Academy of Management Review. Vol. 23, No. 3, 325-340. 1998.

46. Jehn, K.A., Bezrukova, K. “The faultline activation process and the effects of activated faultlines on coalition formation, conflict, and group outcomes” Organizational Behavior and Human Decision Processes. 112 (2010) 24-42

47. Jehn, K.A., Bezrukova, K., Thatcher, S. (2007) “Conflict, Diversity, and Faultlines in Workgroups.” In: Carsten, K.W. De Dreu., Gelfand, M.J.,The Psychology of Conflict and Conflict Management in Organizations. SIOP Organizational Frontiers Series. CRC Press.

48. Jenkins, R. (2004). Social Identity. New York, NY: Routledge.

49. Jensen, C., King S., and Kuechler, V., "Joining Free/Open Source Software Communities: An Analysis of Newbies' First Interactions on Project Mailing Lists," Proc. of the 44th Hawaii Int'l Conf. on System Sciences, pp. 1-10, 2011.

50. Jensen, C., Scacchi, W. "Role Migration and Advancement Processes in OSSD Projects: A Comparative Case Study." Proc. of the 29th Int'l Conf. on Software Engineering, pp. 364-374, 2007.

51. Jergensen, C., Sarma, A., Wagstrom, P., 2011. The onion patch: migration in open source ecosystems. In Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software

engineering (ESEC/FSE '11). ACM, New York, NY, USA, 70-80. DOI=10.1145/2025113.2025127 http://doi.acm.org/10.1145/2025113.2025127

52. Jessup, L.M., Connolly, T., Tansik, D., "Toward a Theory of Automated Group Work: The De-individuating Effects of Anonymity." Small Group Research, vol. 21, no. 3, pp. 333-348, 1990.

53. Krishnamurthy, S., "Cave or Community? An Empirical Examination of 100 mature Open Source Projects". First Monday, vol. 7, no. 6, 2002.

54. Krogh, G., Spaeth, S., and Lakhani, K., "Community, joining, and specialization in open source software innovation: a case study." Research Policy, vol. 32, no. 7, pp. 1217-1241, 2003.

55. Kuechler, V., Gilbertson, C., Jensen, C., “Gender Differences in Early Free and Open Source Software Joining Process,” in in Open Source Systems: Long-Term Sustainability, vol. 378, I. Hammouda, B. Lundell, T. Mikkonen, and W. Scacchi, Eds. Springer Berlin Heidelberg, 2012, pp. 78–93.

56. Lakhani, K.R., Wolf, R.G., "The Boston Consulting Group Hacker Survey." 2002. ftp3.au.freebsd.org/pub/linux.conf.au/2003/papers/Hemos/Hemos.pdf

57. Lakhani, K.R., Wolf, R.G., "Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects." MIT Sloan Working Paper, no. 4425-03, 2003.

58. Lampe, C., Johnston, E., "Follow the (Slash) dot: Effects of Feedback on new Members in an Online Community." Proc. of the 2003 Int'l ACM SIGGROUP Conf. on Supporting Group Work, pp. 11-20, 2005.

59. Lau, D.C., Murnighan, J.K., “Demographic Diversity and Faultlines: The Compositional Dynamics of Organizational Groups.” Academy of Management Review. Vol. 23, No. 3, 325-340. 1998.

60. Lau, D., Murnighan, J. K., (2005). Interactions within groups and subgroups: The dynamic effects of demographic faultlines. Academy of Management Journal. Vol. 48, no. 4, 645-659

61. Lave, J.,Wenger, E., Situated Learning : Legitimate Peripheral Participation. Cambridge University Press, 1 edition, September 1991.

62. Li, Y., Tan, C., Teo, H., Mattar, A.T., "Motivating Open Source Software Developers: Influence of Transformational and Transactional Leaderships." Proc. of the 2006 ACM SIGMIS CPR Conf. on Computer Personnel Research, pp. 34-43, 2006.

63. Lin, Y., "Gender Dimensions in Floss Development," Mute Magazine. November 23, 2005. Accessed December 10, 2010. http://www.metamute.org/en/node/5596

64. Ljungberg, J., Open source movements as a model for organizing. *European Journal of Information Systems*, 9(4), 2000.

65. Loden, M., Implementing Diversity. Chicago: Irwin Professional, 1996.

66. Low, N., Butt, S., Paine, A.E., Smith, J.D., "Helping Out: A national survey of volunteering and charitable giving." National Centre for Social Research and the Institute for Volunteering Research, 2007.

67. Mor-Barak, M.E., Managing Diversity: Toward a Globally Inclusive Workplace. Thousand Oaks: Sage Publications, 2005.

68. Nafus, D., Leach, J., Krieger, B., "Deliverable D16: Gender: Integrated Report of Findings." Free/Libre/Open Source Software: Policy Support. Accessed May 20, 2011. http://www.flosspols.org/

69. Noonan, D.S., Baker, P.M., Moon, N.W., (2008) "Open Source Software Softtware Potential Index (OSPI): Development Considerations." *Context* (2008).

70. Oh, W., Jeon, S., Membership Herding and Network Stability in the Open Source Community: The Ising Perspective. Management Science, 53(7):1068-1101, July 2007.

71. Olson, G., Olson, J., "Distance Matters." Human-Computer Interaction, vol. 15, no. 2, pp. 139-78, 2000.

72. Open Source Initiative. Accessed May 20, 2011. http://www.opensource.org/

73. Powell, W.E., Hunsinger, D.S., Medlin., B.D., (2010) "Gender Differences Within the Open Source Community: An Exploratory Study." *Journal of Information Technology Management*, Volume 21, Number 4.

74. Raymond, E. S. (1999). A brief history of hackerdom. In C. DiBona, S. Ockman, and M. Stone (Eds.), *Open Sources: Voices from the Open Source Revolution*. Cambridge, Massachusetts: O'Reilly and Associates.

75. Raymond, E.S. (1999a) Homesteading the Noosphere. In: The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary, Raymond, E.S. O'Reilly and Associates, Sebastopol, CA. [www.tuxedo.org/~esr/writings/homesteading/home steading.html].

76. Raymond, E.S., (2001) How to be a hacker. http://www.catb.org/esr/faqs/hacker-howto.html

77. Raymond, E.S., (1999). The Cathedral and the Bazaar (1st ed.). Tim O'Reilly (Ed.). O'Reilly & Associates, Inc., Sebastopol, CA, USA.

78. Rheingold, H. (1994) Virtual Community: Homesteading on the Electronic Frontier. Minerva, London.

79. Robert, K., "Standing out in a Crowd." Keynote Presentation, OSCON 2002, San Diego, CA, July 22, 2002. http://www.oscon.com/oscon2009/public/schedule/detail/10173

80. Robles, G., Scheider, H., Tretkowski, I., Webers, N., "Who Is Doing It? A research on Libre Software developers." August 27, 2001. Accessed May 20, 2011. http://widi.berlios.de/paper/study.html

81. Ross-Ackerman, S., "Bribes and gifts." in: A. Ben-Ner and L. Putterman (eds.), Economics, Values, and Organization, Cambridge: University Press, pp. 296–328, 1998.

82. Schroder, C., (2009) "Mark Shuttleworth's Community Has No Women" *Linux Today*. http://www.linuxtoday.com/blog/2009/09/mark-shuttlewor-1.html

83. Seaman, C.B., Basili, V.R., (1997). An empirical study of communication in code inspections. In *Proceedings of the 19th international conference on Software engineering* (ICSE '97). ACM, New York, NY, USA, 96-106.

84. Setlock, L.D., Fussell, S.R., Ji, E., Culver, M., "Sorry to Interrupt: Asian Media Preferences in Cross-cultural Collaborations." Proc. of the 2009 Int'l Workshop on Intercultural Collaboration, pp. 309-312, 2009.

85. Schwern, M., (2012) "Who else are we driving away? What ideas are we losing? Keynote, Yet Another Perl Conference, North America. http://adainitiative.org/2012/09/michael-schwern-who-else-are-we-driving-away-what-ideas-are-we-losing/

86. Skok, M.J., "2013 Future of Open Source-7th Annual Survey Results". 2013. http://www.slideshare.net/mjskok/2013-future-of-open-source-7th-annual-survey-results

87. Spinellis, D., "Global software development in the freebsd project." Proc. of the 2006 Int'l Workshop on Global Software Development for the Practitioner, pp. 73-79, 2006.

88. Sproull, L., Kiesler, S., Connections: New Ways of Working in the Networked Organization. Cambridge, MA: MIT, 1995.

89. Sproull, L., Kiesler, S., "Reducing Social Context Cues: Electronic Mail in Organizational Communication." Management Science, vol. 32, pp. 1492-1512, 1986.

90. Steward, D. "Social Status in an Open-Source Community." American Sociological Review, vol. 70, no. 5, pp. 823-842., 2005.

91. Stewart, K.J., Gosain, S., (2006). "The Impact of Ideology on Effectiveness in Open Source Software Development Teams," MIS Quarterly, (30: 2).

92. Sundeen, R.A., Garcia C., Raskoff, S.A., "Ethnicity, Acculturation, and Volunteering to Organizations: A Comparison of African Americans, Asians, Hispanics, and Whites." Nonprofit and Voluntary Sector Quartely, vol. 38, no. 6, pp. 929-955, 2009.

93. Tajfel, H., Turner, J. C., (1986). The social identity theory of intergroup behavior. In S. Worchel and W. G. Austin (Eds.), Psychology of intergroup relations: 7-24. Chicago: Nelson-Hall Publishers.

94. Tidwell, L.C., Walther, J.B., "Computer-Mediated Communication Effects on Disclosure, Impressions, and Interpersonal Evaluations, Getting to Know One Another a Bit at a Time." *Human Communication Research*, Vol. 28, No. 3, July 2002.

95. Thatcher, S.M.B., in press. Moving beyond a categorical approach to diversity: The role of demographic faultlines. New York: Oxford University Press.

96. Thatcher, S.M.B., Jehn, K.A., Zanutto, E., (2003). Cracks in diversity research: The effects of faultlines on conflict and performance. Group Decision and Negotiation, 12, 217–241.

97. Thatcher, S.M.B., Patel, P.C., "Group Faultlines: A Review, Integration, and Guide to Future Research" Journal of Management, vol. 38, no. 4, 2012 pp. 997.

98. "The Apache Software Foundation." (2013). www.apache.org

99. "Top Ten Languages in the Internet, 2010" Internet World Stats. http://www.internetworldstats.com/stats7.htm

100. Turner, J. C., Hogg, M.A., Oakes, P.J., Reicher, S.D., Witherell, M.S., (1987). Rediscovering the Social Group: A Self-Categorization Theory. New York: Basil Blackwell

101. Turner, J., Oakes, P., (1986). "The significance of the social identity concept for social psychology with reference to individualism, interactionism and social influence". British Journal of Social Psychology 25 (3): 237–252.

102. Ulrich, W., (1987). "Critical Heuristics of Social Systems Design," *Euro. J. Opl. Res.*,31, 276.

103. von Bertalanffy, L., (1950). "The theory of open systems in physics and biology", in: Systems Thinking, F.E. Emery, ed. Penguin, Harmondsworth, pp. 60-85.

104. Walli, S., Gynn, D., Von Rotz, B., "The Growth of Open Source Software in Organizations." Optaros. 2005. http://dirkriehle.com/wp-content/uploads/2008/03/wp_optaros_oss_usage_in_organizations.pdf

105. Weber, S., The Success of Open Source. Harvard University Press, 2004

106. Wenger, E. "Communities of Practice: A Brief Introduction." June 2006. http://www.ewenger.com/theory/index.htm

107. Wheeler, D., "Why Open Source Software/Free Software (OSS/FS, FOSS, or FLOSS)? Look at the Numbers!" April 16, 2007. Accessed May 20, 2011. http://www.dwheeler.com/oss_fs_why.html

108. Whitehurst, J., (2012) "The best idea wins: Jim Whitehurst on the Red Hat Meritocracy." Opensource.com. http://opensource.com/business/12/5/best-idea-wins-jim-whitehurst-red-hat-meritocracy

109. Wiener, N., (1948) *Cybernetics*. Wiley, New York.

110. Wolstenholme, E.F. 199, *Systems Enquiry: A System Dynamic Approach. Wiley, Chichester.*

111. "Women, Minorities, and Persons with Disabilities in Science and Engineering." US National Science Foundation, 2011. http://www.nsf.gov/statistics/wmpd/

112. Xu, B., Jones, D.R., Donald, R., "Volunteer's Participation in Open Source Software Development: A study from the Social Relational Perspective." The DATA BASE for Advances in Information Systems, vol. 41, no. 3, pp. 69-84, 2010.

113. Ye, Y., Kishida, K.," Toward and understanding of the motivation of open source software developers." Proc. of the 25th International Conference on Software Engineering, pp. 419-429, 2003.

114. Zimmerman, M., (2011). "A diversity statement for Ubuntu." http://mdzlog.alcor.net/2011/02/07/a-diversity-statement-for-ubuntu/

Chapter 5. General Conclusion

Over the last six years, FOSS research has slowed, and much of the current research has focused on FOSS adoption in a variety of domains—education, government, business, and industry. This naturally means that the FOSS community will be composed of new members with new relationships with new domains that are learning to understand the FOSS development model and the most effective way to participate in community-centric software development.

Understanding the FOSS joining process and current composition of the FOSS community will help shed light on issues newcomers face when joining a project. More importantly, it will help researchers and the FOSS community at large to create effective initiatives that will manage the growing pains associated with increased diversity within the community.

The three manuscripts that make up this thesis represent a step forward in understanding the FOSS joining process and diversity conflict. In Chapter 2, we discovered that the FOSS community's response to newcomers on mailing lists is positively correlated with future participation. In the Chapter 3, we found that female participation in FOSS is disproportionately lower than men. This suggested that some characteristic of FOSS is negatively affecting women, and it is likely non-technical issue.

In Chapter 4 we respond to this by presenting a theoretical discussion of the lack of diversity in FOSS. The current demographic composition of the FOSS community shows that FOSS is exceedingly homogenous. One explanation for this is that faultlines exist within the community and have inadvertently forced newcomers to assimilate or leave. Upon further investigation using critical system thinking, we presented strong evidence that other factors are affecting

diversity, including an established and accepted ideology and methodology, as well as a several structural characteristics.

Although we presented assumptions about how these factors and characteristics affect the FOSS community, we proposed that future research be undertaken to update the current demographic surveys. We also suggested the research community use systems thinking methodologies and faultlines theory to understand diversity conflict and how to effectively alleviate it.