

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

Carlos Jensen  
School of EECS  
Oregon State University  
[cjensen@eeecs.oregonstate.edu](mailto:cjensen@eeecs.oregonstate.edu)

Scott King  
School of EECS  
Oregon State University  
[s\\_king256@yahoo.com](mailto:s_king256@yahoo.com)

Victor Kuechler  
School of EECS  
Oregon State University  
[kuechlej@onid.orst.edu](mailto:kuechlej@onid.orst.edu)

## Abstract

*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.*

## 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 4, we describe the results of this data collection, the amount of data, characteristics of the eight mailing lists, etc. In section 5, we discuss the research questions and present the data, and in section 6 we present our conclusions.

## 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].

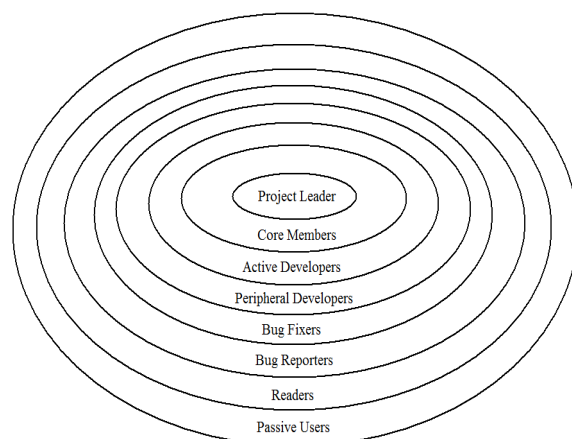
Much has been written about the joining processes in commercial/closed source software (CSS) projects. Sim and Holt [18] studied four employees of a commercial software company joining a new project. They coined the term "software immigrants" to describe these people, as they were not traditional novices 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 [22]**

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 refers 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

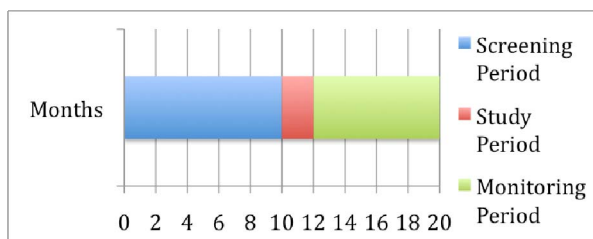
that these would deviate significantly from the above-mentioned.

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.

### 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.<sup>1</sup>
  - 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.

<sup>1</sup> 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” ( $S = 0.93$ ).

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.

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.

## 4. Results

**Table 1. Overview of mailing lists and projects studied.** 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.

Mailing List	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%

**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
<b>TOTAL</b>	<b>643 (86.4%)</b>	<b>101 (13.6%)</b>	<b>744</b>

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 our 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 ballpark 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 ( $p\text{-value} = 0.765$ ).

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.

## **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.

## 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.

## 7. Acknowledgements

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.

## 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](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*, 28<sup>th</sup> 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](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](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>