

Understanding Group Maintenance Behavior in Free/Libre Open Source Software Projects: The Case of Fire and Gaim

Kangning Wei¹

¹School of Management
Shandong University
Jinan, Shandong, 250100
China

kwei@sdu.edu.cn

TEL: 86-18615524338

Kevin Crowston^{2,3}

²School of Information Studies
Syracuse University
Syracuse, NY 13244
U.S.A

³ National Science
Foundation, U.S.A

crowston@syr.edu

Na "Lina" Li⁴

⁴Center for Graduate Studies
Baker College
Flint, MI 48507
U.S.A

nli01@baker.edu

Robert Heckman²

²School of Information Studies
Syracuse University
Syracuse, NY 13244
U.S.A

rheckman@syr.edu

Corresponding author: Kangning Wei

Understanding Group Maintenance Behavior in Free/Libre Open Source Software Projects: The Case of Fire and Gaim

Abstract

In this paper we investigate group maintenance behavior in community-based Free/Libre Open Source Software (FLOSS) development teams. Adopting a sociolinguistic perspective, we conceptualize group maintenance behavior as interpersonal communication tactics—specifically, social presence and politeness tactics—that help maintain relationships among group members. Developer email messages were collected from two FLOSS projects with different development status and content-analyzed to identify frequently-used group maintenance tactics. We then compared the two projects on the group maintenance tactics used, finding differences that reflect changes in the project work practices. Our work contributes theoretically to FLOSS research and has practical implications for FLOSS practitioners.

Keywords: Group maintenance; open source software development; social presence; politeness theory; interpersonal communication

1. Introduction

As an alternative model to commercial knowledge-based collaboration work [1], Free/Libre Open Source Software (FLOSS)¹ has become of increasing interest over the past decade in both the commercial and academic worlds. While FLOSS-licensed software may be developed in the

¹ FLOSS is an umbrella term that covers a diversity of kinds of software and approaches to development. The distinction between free software and open source software is sometimes controversial and there are important differences between these two development communities [2]. However, our focus in this paper is on their development processes, which are acknowledged by participants to be largely similar (Free Software Foundation, <http://www.fsf.org>), hence our use of this umbrella term.

same way as proprietary software (e.g., as in the case of the MySQL database), much of it is developed by teams of organizationally- and geographically-distributed developers, in what has been described as community-based development [3]. This community-based form of FLOSS development is the focus of this paper.

An important feature of the community-based FLOSS development process is that projects depend on voluntary contributions from developers [4]. Sustaining participation—that is, retaining developers' continued contribution and recruiting new contributors—is thus a critical factor for a FLOSS project to continue development [5]. However, it has also been noticed that 80% (or more) of FLOSS projects fade away due to insufficient long-term participation [6]. For example, on SourceForge (<http://sourceforge.net>), one of the most popular FLOSS project portals, over 324,000 projects have been created by 3.4 million developers around the world, but most of these projects are inactive. To sustain participation, it is important (though not sufficient) to sustain social interaction among members in FLOSS projects [7, 8]. Social interaction underlies the software development processes and helps build relations between participants toward shared activities, and thus has positive group-level outcomes [9].

Most prior research on social interaction in FLOSS focuses on the strategies and processes that members use to address particular group work functions such as decision making [10, 11] and coordination [12, 13]. However, how members interact on a daily basis to build and maintain relations that contribute to project continuity is as yet largely unaddressed. We focus on this topic in this paper. The objective of this study is to theorize *group maintenance behaviors in community-based FLOSS projects expressed through daily interpersonal communications*. We define the phenomenon of group maintenance behavior as the pro-social, discretionary and relation-building behavior between members that builds and maintains reciprocal trust and

cooperation [14]. Group maintenance is targeted towards the development and maintenance of social relationships with others [15]. It demonstrates a concern for others as well as the well-being of the group or community [16].

Theories of group behaviour, such as the input-process-output model proposed by Hackman and Morris [17], have emphasized the important role of social relations in effective group outcomes [15]. Group-maintenance behavior is thus known to be important to group function, as it keeps social relations pleasant, increases interdependence among members, facilitates resolution of conflicts, provides encouragement and enables the minority to be heard [18]. Prior group research has found that effective group maintenance increases long-term group performance by fostering various team activities such as higher group cohesion, better information exchange, lower in-group conflict and greater commitment [19-21].

Furthermore, Stewart and Gosain [22] found that since most FLOSS development and communication activities happen online and can be observed by others, they provide an opportunity for potential members to observe others' behaviors and the climate of the project before making a decision to join a project. Thus, positive efforts to maintain good social relationships among FLOSS team members may help recruit new members, in addition to motivating existing members to continue participation. For both these reasons, it is critical to understand group maintenance behaviors in FLOSS teams.

While prior research provides important insights into the nature of group maintenance, the differences between conventional face-to-face groups and virtual and voluntary FLOSS projects suggests that researchers should be cautious in directly applying those findings. Furthermore, reviews of previous research on FLOSS [e.g., 10, 23] indicate that few studies have empirically

examined topics related to group-maintenance behaviors, further emphasizing the nascent state of research on this topic.

Two areas of research that are relevant to this topic are research on socialization, studying how new members join the project and sustain participation, and on ideologies, studying norms shared among members that regulate their interaction behaviors. These two streams of research are related because they examine how members interact with each other to construct positive social relations and contribute to the team.

Research on socialization in FLOSS projects examines the strategies and processes through which new members join an existing FLOSS development project and internalize processes and values that spur participation [6, 24-26]. This body of literature treats socialization as a process and places emphasis on a participant's actions and willingness to understand not just the code base but also the social structure of the project. While these studies provide a good understanding of how new members interact with others and move from a non-participant to a fully-fledged FLOSS developer, they do not explore the specific tactics established members use every day to build and maintain social relations when interacting with others.

A small body of existing FLOSS research examines ideologies (i.e., norms, beliefs and values) shared among project members and their impact on relationship-building behaviors [27]. In general, this literature argues that, shared ideologies help build and maintain social relationships among team members [22, 28, 29]. Most of these studies examine the "big picture" of ideologies shared in the wider FLOSS community and on issues related to software/technical/work processes. However, group maintenance behavior does not take place only through these big events. Rather, it is enabled and manifested in practical everyday interaction [29]. Researchers

have argued that individual interaction can lead to surprising group-level outcomes [30]. For example, Kozlowski and Chao [31] found that social interaction among team members increases team cohesion. In FLOSS projects, these everyday interaction between team members shape the members' behaviour while engaged in FLOSS development activities, which in turn impact group-level outcomes such as reciprocal trust and cooperation (i.e., group maintenance).

In summary, while research has touched on issues related to group maintenance, researchers to date have paid little attention to seemingly mundane activities and everyday interactions that underlie FLOSS development [7, 29]. While prior research provides useful insights into FLOSS projects, it remains unclear what specific tactics are used in everyday interaction to maintain group relations. To address this gap and to understand how the existing literature on group maintenance tactics can be applied to FLOSS projects, in this study, we first address the following research question:

RQ1: What group maintenance tactics do community-based FLOSS members use to maintain relations through everyday interaction?

Secondly, previous research has pointed out that many FLOSS projects fail due to insufficient volunteer participation [6, 32, 33]. Group maintenance behavior helps build and maintain relationships among group members, thus playing an important role in sustaining participation and continuous development. Conversely, changes in exhibited group maintenance behaviors may be an indication of changing group dynamics. Thus, we expect to observe differences in the group maintenance behaviors in a project that sustains development and one that ceases development, reflecting differences in the nature of participation and practice in the projects. More specifically, we expect to observe these different group maintenance behaviors develop

over time as the projects mature and prosper or wind down. Seeing such differences will provide further evidence of the validity of the application of group maintenance concepts in this setting.

Therefore, our second research question is the following:

RQ2: What differences in group maintenance behaviour exist in general and over time between projects that sustain development and those that eventually cease development?

We note that we are not proposing a casual relationship between group maintenance behaviors and project success (e.g., that a lack of group maintenance behaviors causes a project to fail). Rather, we expect that the differences in group maintenance behaviors among projects reflect variations in the group processes that also lead to different project outcomes. In this way, observable group maintenance behaviors can serve as a useful proxy for harder-to-observe group practices.

The rest of the paper is structured as follows. In the next section, we first introduce the two theories that provide the theoretical grounding for our study, social presence and politeness theory from computer-mediated communications. Then we describe our research method, including project selection strategies, data collection techniques and data analysis methods. The results are presented in the following section. Finally, we discuss the implications and the limitations of our study, and make recommendations for future research.

2. Theoretical Development: Social Presence and Politeness Theory

The central element of group-maintenance behavior is the act of building and maintaining relations (e.g., reciprocal trust and cooperation) among group members through everyday interaction. In FLOSS development, developers mainly interact through text-based

communication tools such as developer email lists and discussion fora. Media Richness Theory has pointed out that text-based communication is less rich than face-to-face communication because it disables the conveyance of nonverbal cues such as facial expressions, which is important in helping building relations [35, 36]. As a result, participants adopt alternative strategies to provide paralinguistic and social cues [37]. Specifically, language plays a basic role as relational meaning is conveyed through specific language elements [34].

In this research, we adopt a linguistic analysis to develop deeper insights about group maintenance behavior in community-based FLOSS development. Specifically, we build on two theoretical bases that have been used to study relationship building via text-based communication: theories of social presence [38, 39] and politeness theory from computer-mediated communications (CMC) [34]. These two theories provide complementary theoretical insights that identify interpersonal communication tactics that individuals perform to build and maintain relationships, and collectively provide a basis to identify potential group maintenance tactics applicable to FLOSS development teams.

2.1 Social presence

Social presence explains “the sense of being with another” [40]. It has been studied extensively in computer-mediated communication and various definitions and categorizations have been provided to describe this phenomenon [See 40 for a review of different definitions]. A widely accepted definition was proposed by Short et al. [41], which defined social presence as “the degree of salience of the other person in a mediated interaction and the consequent salience of the interpersonal interaction” (p. 65).

Social presence is important because prior research has demonstrated the impact of social presence on outcomes such as group cohesion [e.g. 42], trust [e.g. 43], social identity [e.g. 44] and project participation [e.g. 45], which are the direct outcomes of group maintenance as we have defined it. In general, a positive relationship has been found between social presence and these outcomes [46], that is, a high degree of social presence perceived by team members is a contributor to a cohesive and sustained community. These results suggest the importance of communication tactics that build social presence as an important component of group maintenance and so project outcomes. For example, Shen and Khalifa [45] found that online community users' sense of social presence had strong positive impacts not only on their motivations to participate, but also on their level of contribution.

Social presence is also interesting because of the particular features of the FLOSS setting. In a virtual environment such as community-based FLOSS development, members communicate mainly through lean media such as email and discussion fora, where text is often the only way to express a message. At first glance it would seem difficult for team members to build high degree of social presence in this setting since social presence is more easily established in the absence of ambiguous and equivocal informational cues (i.e., in richer media) [47]. To increase social presence, these participants must enact communicative tactics such as using emoticons and paralanguage through their written communication that compensate for the lack of other cues in the lean medium [38].

Many researchers have studied the tactics individuals used to build social presence in CMC settings. A majority of these studies has focused on online educational environment [e.g., 48, 49]. For example, Rouke et al. [39] provided a template for assessing social presence in asynchronous text-based computer conferencing learning settings through content analysis of conferencing

transcripts. Linguistic symbols were identified to indicate social presence such as use of emoticons, humor, inclusive pronouns and vocatives (i.e., referring to participants by name or address to a specific person). The insights gained from this literature review are used to develop our measures of social presence.

2.2 Politeness theory

A second theoretical base for our study is politeness theory. Politeness theory explains how people phrase communications in a way that take into consideration the feeling of the others in relational communication [50], thus contributing to the development of social relations. Other CMC researchers studying relational communication have used politeness theory. For example, Morand and Ocker [34] state “The specific tactics of politeness can be reliably observed and thus quantitatively measured; as such they can be used in the assessment of relationalities within CMC, at a linguistic level of analysis” (p. 5). There are three main elements to politeness theory: face, face threatening acts (FTA) and politeness strategies [51]. The first two of these elements provide the theoretical argument for the importance of the final element, the communication strategies. Politeness theory thus provides a complementary theoretical base for identifying additional group maintenance behaviors.

Face is the central element in politeness theory and is defined as the positive value individuals claim for the public self they present [34]. Because face is emotionally charged and is inherently vulnerable when engaging others in interaction, people strive to maintain face in social settings and communications [52]. However, the identity that one claims can only be validated by others and so is dependent on others. It thus becomes within everyone’s interest to maintain the group by maintaining the face of those they interact with [53]. Face is therefore viewed as “a social rather than a psychological construct” [54]. And it is within these social situations that people

continuously interact in ways that preserve, bolster, or show consideration for the face of others [52]. Thus, politeness theory emphasizes interactional support work directed toward others' face [34]. Face is constructed of two wants: autonomy of action (also known as negative face) and the need for validation (also known as positive face) [51]. Negative face is exemplified by wanting to be left alone, independence from others, self-direction, and freedom from restrictions created by others; meanwhile positive face includes want of respect, membership in a valued community, and a reputation for competence and fairness [55].

Despite the need to support both the negative and positive face of others, there are instances when one may have to “make requests, disagree, and offer advice or criticism to others” [55]. These instances are known as face threatening acts (FTAs), and can either be directed toward the speaker or the hearer, and can threaten both types of face [51].

Finally, politeness strategies are linguistic acts that take the forms of positive tactics (to encourage positive face) and negative tactics (to encourage negative face) to redress or mitigate any threats to others' face engendered by an FTA [34]. Examples of positive politeness tactics include use of colloquialisms or slang, vocatives, agreement, inclusive pronouns and sympathy that express positive face, e.g., membership in a project. Examples of negative politeness tactics include use of hedges, indirect inquiries, subjunctives, honorifics, apologies, formal verbiage, passive voice, and rationale for FTAs that preserve negative face, e.g., not giving direct instructions to preserve self-direction [34, 52]. By supporting, preserving or restoring face, both kinds of communication tactics help maintain relationship among group members. Collectively, these politeness strategies are thus additional elements in the understanding of group maintenance behaviors.

In summary then, these two bodies of theories suggest specific communication behaviors that individuals might exhibit that serve to build and support relationships among group members. The setting of Rouke et al.'s study of social presence [39] is comparable to FLOSS projects, where text-based messaging is the major communication channel for distant learning. We therefore adapt Rouke et al.'s template to examine social presence as a kind of group maintenance tactic in FLOSS teams. In online settings, politeness strategies are also visible in the group discussions. We adapted Morand and Ocker's [34] approach to examine politeness strategies as a kind of group maintenance behaviour in FLOSS teams.

3. Research Method

As noted above, prior research has identified a set of possible group maintenance behaviors that are exhibited in conventional teams and for which evidence has been found in online settings. However, given the differences between those settings and FLOSS teams, it is not clear which if any of the behaviors would be applicable to FLOSS teams. Given the current state of the research on this topic, we adopted an exploratory case study approach [56]. We studied group maintenance behavior through a content analysis of email archives from two FLOSS teams. This strategy provides a deep understanding of the applicability of the theories in these settings, though at the expense of generalizability, a tradeoff that we return to in the conclusion. The following section describes our case selection strategy, data collection method, coding scheme development and coding process, and data analysis methods.

3.1 Case selection

Several criteria were settled to assist in selecting ideal FLOSS projects for this study. First, to control unwanted variance brought by systematic factors such as different types of software

developed and different ages of the projects, we wanted to focus on projects that were developing similar software. Second, to answer research question 2, we needed to explore a project that continued to have active development activities and a project that lasted for a fairly long period but ceased development activities eventually, so that we could examine whether any differences exist in group maintenance behavior between these two categories of projects. Needing to know the fate of the projects led us to a retrospective analysis, picking projects with a known outcome and looking at the activities of projects leading up to that point.

Keeping these criteria in mind, we selected two projects from SourceForge.net for this study: Gaim and Fire. These two projects met the first criterion because they both developed the same kind of software: multi-platform Instant Messaging (IM) clients. A user of Gaim or Fire can run a single program and chat with users across multiple services. The two projects were thus similar in terms of their project goals, nature of tasks, and potential users. The two projects met the second criterion as well: Gaim continues operating as an active project today (although it is now known as Pidgin), while Fire ceased development in early 2007.

3.2 Data collection

Since most (if not all) FLOSS activities are archived, FLOSS projects provide a unique setting in which to examine group maintenance behaviors over time. In this study, we analyzed the email messages from the public mailing lists of the two projects. The fact that Fire ceased development in early 2007 limited us to collect messages posted before early 2007 for both projects. As a result, we decided to select messages from Fire between June 2002 and December 2005, which allow us to have enough messages to observe group maintenance behavior before it ceased development. Messages from Gaim were selected between June 2002 and February 2006. Analyzing email messages over such extended periods provided us with a sampling frame to

examine an essentially complete record of group maintenance behavior over time. While software development has certainly evolved since the time our data were generated, our study examines everyday interaction that is a fundamental social process of how people interact with each other, a phenomenon that does not change rapidly over time. Furthermore, many FLOSS projects still use the same communication technologies, email and discussion fora. Therefore, analysis of this data should generalize to current FLOSS development.

Since there were thousands of messages exchanged during the selected period for each project, it was infeasible to analyze all of them using the intensive manual coding method described below. Instead, we sampled messages from each project to study. We wanted a sample of at least 300 messages, large enough to provide sufficient power for our statistical tests, but small enough to be tractable for analysis. Because our second research question involved a comparison over time, we developed a hierarchical sampling strategy to uniformly cover the history of the projects.

We started by dividing the messages from each list into 360 sequential “chunks” of messages. In each list, all chunks had the same number of messages but as a result did not cover the same time duration. We then randomly selected one message from each chunk of messages. However, there were periods when large amounts of spam messages were sent to the lists and archived in the repositories. As well, the lists received messages sent by various automated tools. Because these messages do not represent the behaviors of developers, they were filtered out of the sample. During the coding process we replaced messages identified as automated or spam with the nearest usable messages from the archive. If all the messages in a chunk were spam, then the chunk was removed from analysis. Using this process, 360 messages were selected from the Gaim mailing list. These messages spread from June 2002 to February 2006, covering 45 months. 336 messages were selected from the Fire mailing list from June 2002 to December 2005,

covering 43 months; the other 24 chunks had only spam or automated messages that were dropped, reflecting decreased level of activity in the project. The descriptive statistics for the sample messages are illustrated in Table 1.

Table 1. Descriptive Statistics of the Sample Messages

Projects	Number of Messages	Message Length (number of words)			
		Min	Max	Mean	Std. Deviation
Fire	336	6	1429	96	125
Gaim	360	5	1037	114	120
Fire & Gaim	696	5	1429	105	123

3.3 Analysis approach

Given the nature of our data, namely textual email messages, we adopted a qualitative data analysis approach. It is sometimes assumed that the goal of qualitative analysis is to uncover latent or hidden meanings in a text, e.g., to understand individuals' concepts of their social worlds from their communications, that is, that qualitative research is always interpretivist. But qualitative research can, in fact, adopt any research perspective: positivist, interpretivist or critical [57]. In our study, we assume that social processes of the groups studied are accurately reflected in the texts that they produce as part of their work together. Rather than hidden meanings, we look for explicit evidence of particular behavioural patterns of the participants. Our approach is thus essentially positivist, despite its reliance on qualitative data.

Specifically, for this study, we identified the communicative tactics individuals used to maintain team relations in the two projects though quantitative content analysis of their messages. In this data analysis approach, we seek to identify units of text in the email messages that are examples of particular theoretical categories (in our case, of particular group maintenance behaviors) and

label those text segments with a tag for the category, a process referred to as coding². Coding is guided by a coding scheme that provides definitions of theoretical categories of interest and guidance for how these categories are exhibited in the text. The result of the coding process is a text annotated (or tagged) with codes for the categories exhibited [58]. Quantitative content analysis allows researchers to determine specific frequencies of relevant categories and examine the relationships involving these categories using statistical methods [59]. Specifically, we compare the frequencies of different group maintenance behaviors in two different projects.

As the unit of coding, we adopted the thematic unit, defined as “a single thought unit or idea unit that conveys a single item of information extracted from a segment of content” or the “unit of meaning” [60]. Such units vary in size from an emoticon or punctuation, to a word, a phrase, a part of a sentence, a sentence, or even a few sentences that capture the meaning.

The initial analysis of the textual data was done by two research assistants, referred to as coders. Two or more coders are needed to be able to compare the separately analyzed data and isolate mistakes or errors in judgement [61]. Although two coders is only the minimum requirement to determine reliability, for practical reasons, this number of coders is commonly used [e.g. 62, 63]. For example, to study citizen-driven information processing through Twitter services, Oh, et al. [63] investigated three social crises and used two coders for each crisis to code the studied variables. The two coders working on our project had a basic understanding of software development processes. Though they had not worked as software developers, they had two years of experience studying FLOSS development practices, which were the focus of the analysis. Since we were not focusing on programming, but on the social interaction of the people, their

² Appendix 2 gives examples of the text segments labeled with categories. For example, a sentence “Well thanks a lot for you hard work!” was identified from the email messages and was coded and labeled as “appreciation”.

knowledge of FLOSS development and the projects was sufficient for them to code the collected messages for the concepts of interest, even when technical terms were used.

3.4 Coding scheme development

We developed an initial coding scheme both deductively and inductively to identify and categorize group maintenance tactics from the messages to be analyzed. Deductively, an initial coding scheme for the exhibited group maintenance tactics was derived from the two theories discussed above, social presence theory and politeness theory. The coding scheme for tactics to establish social presence was adapted from Rourke et al. [39]. In their study, social presence in an asynchronous online learning setting was assessed in three categories, affective, interactive and cohesive responses. The coding scheme for politeness tactics was adapted from Morand and Ocker [34] and included both positive and negative politeness tactics. After comparison, we found that indicators for interactive and cohesive responses from Rourke et al's study were similar to the positive politeness tactics from Morand and Ocker's study, both indicating group closeness. Therefore, for our coding scheme we combined these three sets of codes together in a category named "positive politeness". Affective responses were renamed "emotional expression". The negative politeness category's name was adopted from Morand and Ocker [34].

Inductively, the initial coding scheme was further revised and refined by the two coders and the authors through pilot coding an independent sample of approximately 400 messages³ from the Gaim and Fire mailing lists. The coders first independently coded a subset of the messages and their results were discussed among the two coders and the authors, to correct the coders'

³ These messages were independently sampled for the purpose of refining the coding scheme, and so could overlap with the messages in the sample used for the main study. Because of concerns about the representativeness of the sample and the possible introduction of biases in coding during the coding scheme development process, these were not used for the analysis reported here. Rather a new sample was drawn as described in the text and coded with the scheme developed.

misunderstandings and to identify problematic codes. The scheme was adjusted based on difficulties encountered by the coders. The revised scheme was then used to code more messages and the coding scheme was discussed and revised again. The iterative process repeated until all disagreements were addressed and a reliable coding scheme was achieved, as described below.

We made the following changes to the coding scheme during this development and refinement process. First, fifteen of the indicators from prior work were removed. Twelve indicators were removed because they occurred very infrequently or not at all in the messages in our sample, an example of the need to customize the theorizing for this setting. Examples included “self-disclosure” from Rourke et al [39] and “notice hearer's admirable qualities or possessions, show interest, exaggerate”, “ellipsis”, “claim common view”, “give reasons”, and “assert reciprocal exchange or tit for tat”, from Morand and Ocker [34].

The other three indicators removed were “continuing a thread”, “quoting from others’ messages” and “asking questions” from Rourke et al. [39]. Although these had high occurrences in the sampled messages, we believe that in this setting, these indicators do not capture the desired behavior of maintaining relationships. Rather, “continuing a thread” and “quoting from others” are simply a matter of using asynchronous software (i.e., email) features to carry on a conversation, while “asking questions” is a common behavior in human communication. Inclusion of these indicators would inflate the apparent occurrence of group maintenance behaviors, so they were removed from the coding scheme.

Second, two indicators were added based on observation of the messages. Participants in both projects often used jargon (e.g., abbreviations of technical terms) or metaphors in their emails. Similar to the usage of colloquialisms and slang (codes already included in the scheme), these

terms communicate commonalities among team members. People who understand and use group-specific jargon or metaphors are perceived to be closer to other group members. In contrast, those who do not understand the jargon terms may feel further distance to other members. Hence, the usage of “jargon/metaphor” indicates positive politeness among group members. This indicator was added to the Positive Politeness category in the coding scheme. The other added indicator was “participation”, which indicates behaviors that encourage group members to participate in discussion. As we discussed earlier, member participation is important to sustained development of FLOSS projects. However, participation in FLOSS development is voluntary. So messages that encourage others to participate are perceived as trying to build relationships with others. This code was also added to the positive politeness category.

Third, four pairs of indicators were combined because they were found conceptually to be similar in nature or serve the same purposes, and practically, to often co-occur in a thematic unit and to be hard to distinguish in coding. For example, “using hedges” and “using subjunctive” [34] were combined into one indicator named “Hedges/Subjunctives”. “Vocatives” and “referring explicitly to others’ messages” [39], were combined into a single indicator named “vocatives”.

Finally, “complimenting, expressing appreciation” [39] was split into two indicators, “complimenting” and “appreciation”, because they were felt to convey different meanings. “Complimenting” refers to praising others or message contents but it does not necessarily express thankfulness for the work. On the other hand, “Appreciation” refers to expressing gratefulness for other’s effort but it does not necessarily indicate a compliment about the work.

Appendix 1 details the evolution of our coding scheme, showing the codes that were removed, added, combined and split. Appendix 2 shows the list of tactics used in group maintenance with

definitions and examples. The final coding scheme contains 15 indicators in three categories, emotional expression (2 codes), positive politeness (9 codes) and negative politeness (4 codes).

3.5 Coding reliability

We assessed inter-rater reliability between the two coders using simple percent agreement rate [64]. Cohen's Kappa is often used instead of simple agreement to correct for the effect of chance agreement. In our study though, the codes are infrequent at the level of thematic units (that is, only a few words or sentences in a message are examples of group maintenance behaviors), so the probability of chance agreement is low, obviating the need for the correction. During the coding scheme refinement process, the inter-rater reliability between the two coders reached 80% in the second half of pilot-coding process and 85% in the last 1/5 of the pilot-coding process. This level of agreement is generally considered acceptable for the research [64]. Therefore, after the reliability of the coding scheme was established, each of the two coders independently coded half of the messages sampled for this study.

3.6 Statistical analysis

Statistical analyses were performed on the coded data to answer the research questions. The unit of analysis for the analysis is the message. The measure of group maintenance behavior we analyzed is the frequency of code occurrence, which is defined as the count of a particular code per message. Messages contain multiple thematic units and so can have multiple instances of a particular code (i.e., multiple examples of a particular group maintenance behavior) in a message. To explore the possible effects of differing length of messages, we ran the same tests using density, the count of a particular indicator in a message divided by the number of words in the message [39]. However, we did not find meaningful differences between the results of these two

approaches, so for simplicity of presentation, we present results just for the frequency of indicator occurrence.

The analysis proceeded in several steps. First, we averaged the numbers of indicator occurrences per message at the project level for the two projects to assess the general patterns of communication tactics used for group maintenance in FLOSS projects (RQ1). Second, to answer RQ2, we conducted a series of Mann-Whitney U tests comparisons between Fire and Gaim to identify between-group similarities and differences in group maintenance behaviors. We used a non-parametric test because the counts of codes were not normally distributed. Because we were making multiple comparisons (15 comparisons, one for each indicator), it seemed possible that one or more comparisons might achieve significance by chance. To correct for this effect, we applied a Bonferroni correction to the usual cut-off alpha of 0.05, which resulted in a required alpha of 0.003 to declare statistical significance. This approach is conservative and does reduce the power of the statistical tests, an issue we discuss below.

To identify when differences in group maintenance behaviors between Gaim and Fire arose (RQ2), we compared the frequencies of group maintenance behaviors in messages from two periods for each project. Because the two projects initially were about equally active, we expected Fire and Gaim to show similar group maintenance patterns at the beginning of the study period. However, we expected the projects to develop differences over time as Fire began to wind down while Gaim continued development. To examine this change, we compared separately the frequencies of group maintenance behaviors from Fire and Gaim in the first 1/3 of the messages (i.e., the first 112 messages from Fire and the first 120 messages from Gaim) and in the last 1/3 of the messages (i.e., the last 112 messages from Fire and the last 120 messages from Gaim), omitting the middle transitional period.

4. Findings

In this section, data analysis results are presented that address the two research questions. We first report the general pattern of group maintenance behaviors that appeared in the two projects. Then we compare group maintenance behaviors between Fire and Gaim.

4.1 RQ1: What group maintenance tactics do community-based FLOSS members use to maintain relations through everyday interaction?

Our results indicate that messages from FLOSS projects did exhibit a level of group maintenance behaviours, especially on a certain set of indicators. Table 2 displays the means, medians, ranges and percentages of the occurrence frequencies of the group maintenance indicators found in the selected periods of Gaim and Fire. The 360 Gaim messages included a total of 3748 occurrences of group maintenance indicators, while the 336 Fire messages included 2861 occurrences. From the table we can see that the use frequencies of different tactics were diverse, with some more highly used (when aggregated at the message level) while some others rarely used. Tactics that were used in more than 10% of messages are highlighted in bold type in Table 2.

10 tactics (out of 15) were used in more than 10% of messages. Jargon/metaphor (mean=4.63 occurrences per message), hedges/subjunctives (mean=1.48), vocatives (mean=0.78), emoticon/capitalization/punctuation (mean=0.51) and inclusive pronouns (mean=0.63) were the five most frequently observed group maintenance behaviors. Among them, three were from the positive politeness category, one from emotional expression, and one from negative politeness.

Table 2. Descriptive Statistics of Occurrence Frequencies of Indicators per Message across Gaim and Fire

Category	Indicator	Mean	Median	Range	Percentage*
Emotional Expression	Emoticon/capitalization/punctuation	0.51	0	0–15	30.5
	Humor	0.10	0	0–3	8.2
Positive Politeness	Slurring/colloquialisms	0.35	0	0–6	24.7
	Jargon/metaphor	4.63	3	0–94	92.4
	Vocatives	0.78	0	0–15	35.2
	Inclusive Pronouns	0.63	0	0–17	26.4
	Phatics	0.31	0	0–3	25.0
	Complimenting	0.08	0	0–3	6.9
	Agreement	0.04	0	0–2	3.5
	Participation	0.05	0	0–1	5.3
	Appreciation	0.21	0	0–2	18.7
Negative Politeness	Disclaimers	0.14	0	0–3	11.1
	Rational for FTA	0.09	0	0–4	7.8
	Hedges /subjunctives	1.48	1	0–37	58.1
	Formal Verbiage	0.08	0	0–4	6.9

*The percentage of messages that contain the given indicator out of the total number of messages in the sample.

4.2 RQ2: What differences in group maintenance behaviour exist in general and over time between projects that sustain development and those that eventually cease development?

Two analyses were carried out to answer the second research question. First, we compared Fire and Gaim overall to reveal a general pattern of whether and how Fire and Gaim differed in the group maintenance behaviors exhibited. The results indicate that the frequencies of certain group maintenance behaviors in Fire and Gaim did differ. Thus, our next step was to explore when the differences emerged by comparing Fire and Gaim in two different periods. The results confirmed our expectation that Fire and Gaim would show similar group maintenance patterns at the beginning of the projects, but that differences would emerge over time.

Comparison between Fire and Gaim

Table 3 displays the means and ranges of occurrence frequencies of each group maintenance indicator within Fire and within Gaim. The table also shows the level of statistical significance of

the difference in frequency according to a Mann-Whitney U test for each indicator. The test compares the mean rank of units from the two samples in a combined sorted list. The unit of analysis for this test is the message. The null hypothesis tested is that the frequency of occurrence of the indicator is drawn from the same distribution in the two samples of messages.

Table 3. Descriptive Statistics of Indicators Occurrence Frequency per Message for Fire and Gaim and Comparisons between Fire and Gaim

Category	Indicator	Mean		Range		Mann-Whitney U Test		
		Fire	Gaim	Fire	Gaim	Mean Rank		p value
						Fire	Gaim	
Emotional Expression	Emoticon/capitalization/punctuation	0.46	0.56	0–6	0–15	342	354	0.331
	Humor	0.04	0.16	0–2	0–3	333	363	0.000*
Positive Politeness	Slurring/colloquialisms	0.35	0.36	0–6	0–5	348	349	0.920
	Jargon/metaphor	3.80	5.41	0–27	0–94	317	378	0.000*
	Vocatives	0.84	0.73	0–10	0–15	348	349	0.993
	Inclusive Pronouns	0.68	0.58	0–9	0–17	361	337	0.049
	Phatics	0.36	0.27	0–3	0–2	361	337	0.039
	Complimenting	0.13	0.03	0–3	0–2	364	334	0.000*
	Agreement	0.02	0.05	0–1	0–2	345	352	0.134
	Participation	0.05	0.06	0–1	0–1	348	349	0.771
	Appreciation	0.31	0.11	0–2	0–2	382	317	0.000*
Negative Politeness	Disclaimers	0.10	0.18	0–2	0–3	338	358	0.015
	Rational for FTA	0.09	0.09	0–4	0–2	346	350	0.576
	Hedges/subjunctives	1.16	1.77	0–24	0–37	319	376	0.000*
	Formal Verbiage	0.10	0.06	0–2	0–4	357	341	0.019

* $p \leq 0.003$ is considered significant (Bonferroni correction for 0.05)

Most indicators of group maintenance behaviors (10 out of 15) demonstrated similar patterns for both groups. However, significant differences between Fire and Gaim were found for five group maintenance indicators. In the emotional expression category, messages in Gaim displayed significantly more frequent use (mean rank=363, $p=0.000$) of humor tactic than those in Fire (333). In the positive politeness category, messages in Fire displayed more frequent usage of appreciation (mean rank=382, $p=0.000$) and complimenting (mean rank=364, $p=0.000$), but less use of jargon/metaphor (mean rank=317, $p=0.000$) than those in Gaim (317, 334 and 378

respectively). In the negative politeness category, messages in Fire displayed significantly less use of hedges /subjunctives (mean rank=319, $p=0.000$) than those in Gaim (376).

Comparisons between Fire and Gaim within periods

The previous comparison included behaviors from the entire life of the two projects. To examine when the differences emerged during the evolution of the projects, we conducted Mann-Whitney U Tests comparing Fire and Gaim in the two selected periods. Table 4 displays the means and ranges of the occurrence frequencies of group maintenance indicators in the first and second periods and the results of the comparisons. The results are consistent with our expectations: in the first period, Fire and Gaim showed hardly any difference except for one indicator, humor. It appeared that members of Gaim used more humor (mean rank=124⁴, $p=0.003$) than Fire (109) in the first period. But in the second period, Fire and Gaim showed differences in four indicators. Messages in Fire displayed more appreciation (mean rank=131, $p=0.000$) and formal verbiage (mean rank=125, $p=0.000$), but fewer disclaimers (mean rank=110, $p=0.002$) and hedges /subjunctives (mean rank=98, $p=0.000$) than Gaim (103, 109, 134 and 123 respectively).

We also made across-time comparisons for Fire and Gaim separately, expecting to find that members in Fire changed their group maintenance behaviour over time, while those in Gaim did not. As expected for Gaim, the results did not show significant changes for most indicators, except for humor, which declined (i.e., messages from Gaim in the first period seemed exceptional in the higher use of humor). However, the analysis for Fire also did not show significant changes in the frequencies of behaviors, except for a decline in the frequency of vocatives. The lack of significant differences reflect a lack of power in the tests due to the

⁴ Note that the magnitude of the mean ranks in the test depends on the sample size; they are not comparable across the different tests.

reduced sample size in the subgroup comparisons and the more conservative decision threshold due to the Bonferroni correction. These negative results make the differences between Fire and Gaim in the second period all the more striking, as they indicate that the two projects evolved in different directions, leading to the seeming paradox of the two projects becoming different while not individually changing in statistically significant ways. It seems that the projects changed at a level below the power of the tests to detect individually, but detectable when compared to each other.

Table 4. Descriptive Statistics of Indicator Occurrence Frequencies per Message for Fire and Gaim and Comparisons between Fire and Gaim in the First and Second Periods

Indicator	First Period				Second Period				First Period			Second Period		
	Mean		Range		Mean		Range		Mean Rank		p	Mean Rank		p
	Fire	Gaim	Fire	Gaim	Fire	Gaim	Fire	Gaim	Fire	Gaim		Fire	Gaim	
Emotional Expression														
Emoticon/ capitalization/ punctuation	0.38	0.66	0-4	0-15	0.41	0.50	0-6	0-9	110	123	0.080	115	118	0.629
Humor	0.08	0.25	0-2	0-3	0.02	0.09	0-1	0-3	108	124	0.003*	115	118	0.279
Positive Politeness														
Slurring/ colloquialisms	0.21	0.40	0-2	0-5	0.38	0.29	0-6	0-4	113	120	0.296	120	114	0.349
Jargon /metaphor	4.13	5.63	0-23	0-94	3.79	5.26	0-24	0-29	110	123	0.142	107	125	0.043
Vocatives	0.95	0.88	0-10	0-15	0.69	0.64	0-10	0-6	123	111	0.128	112	121	0.188
Inclusive Pronouns	0.64	0.53	0-9	0-14	0.54	0.73	0-6	0-17	125	109	0.017	115	118	0.625
Phatics	0.28	0.27	0-2	0-2	0.41	0.22	0-3	0-2	117	116	0.806	126	108	0.007
Complimentin g	0.11	0.03	0-2	0-1	0.12	0.03	0-3	0-2	120	113	0.044	120	113	0.034
Agreement	0.04	0.06	0-1	0-2	0.02	0.04	0-1	0-1	117	116	0.933	115	118	0.290
Participation	0.10	0.05	0-1	0-1	0.04	0.03	0-1	0-1	119	114	0.160	117	116	0.921
Appreciation	0.18	0.13	0-2	0-2	0.34	0.08	0-2	0-1	120	114	0.250	131	103	0.000*
Negative Politeness														
Disclaimers	0.13	0.18	0-2	0-2	0.02	0.15	0-1	0-3	114	118	0.441	110	123	0.002*
Rational for FTA	0.13	0.13	0-2	0-2	0.04	0.08	0-1	0-1	116	117	0.838	114	119	0.129
Hedges /subjunctives	1.40	1.42	0-13	0-15	0.93	2.20	0-13	0-37	119	114	0.584	98	134	0.000*
Formal Verbiage	0.08	0.10	0-2	0-4	0.20	0.03	0-2	0-1	116	117	0.917	125	109	0.000*

* $p \leq 0.003$ (Bonferroni correction)

5. Discussion

The goal of this study was to advance our understanding of the nature of group maintenance behavior in community-based FLOSS development teams. Our results suggest that FLOSS team members do exhibit certain types of group maintenance tactics in their communication that are similar to but not identical to other kinds of virtual teams. As well, differences in the frequencies of these behaviours were found between a project that continued development and one that ceased development, overall and during later periods in the projects in particular. These differences reflect the outcome and illuminate the underlying processes in the teams. In the following sections, we discuss our findings in turn.

5.1 Major findings

RQ1: What group maintenance tactics do community-based FLOSS members use to maintain relations through everyday interaction?

By examining occurrence frequencies of group maintenance tactics and the percentages of messages that used these tactics, our results suggest that FLOSS team members frequently use a number of communication features in the form of social presence and politeness tactics that serve to maintain team social relations.

Emotional expression. Intuitively, one might expect a low degree of emotional expressions in FLOSS development discussions for several reasons. First, the interaction in FLOSS development is typically task-oriented [65]. Research has found that individuals express fewer emotions in task-oriented contexts than in social-emotional contexts when using CMC [e.g. 66], e.g., by using fewer emoticons. Second, members are geographically distributed and some have never met each other, suggesting a low level of personal connection. And third, the communication medium is lean and not conducive to emotional expression.

However, our study contradicts this expectation. We found that FLOSS team members frequently used a number of communicative tactics that expressed emotions in their online discussion. The most common tactic we found was the use of emoticons, capitalization and punctuation, which were present in almost 1/3 of the sampled messages (30.5%). For example, one developer in Fire used an emoticon “;-)” to simulate a smile (i.e., a facial emotional expression) when talking about his computer: *“I will check if that crash also happens on my g4 powerbook which is much slower (and of course has only one cpu ;-))”*. In another example, a developer from Gaim used punctuation (“???”) to strongly express his eagerness to know an answer: *“Also, how do you send a message to someone??? I see a useful file Conversation_IM.c which has likely looking calls, but I have no idea what the arguments are.”* By using these tactics, individuals can communicate via email in ways that resemble the use of body language, facial expressions or even speech tones used in face-to-face communication [67]. Such emotional tactics enable members to convey social information that mere words may not be able to carry, which in turn can increase mutual understanding and maintain interpersonal relationships among members [67].

Positive and negative politeness tactics. Our results show that members of the FLOSS projects studied used both positive and negative politeness tactics, but with an emphasis on the former. Positive and negative politeness tactics serve two different functions: positive politeness tactics are those that intend to bond people together, while negative politeness tactics reduce threats to face by respecting others’ autonomy and keeping others from getting too close. Our findings indicate that although members were somewhat careful to respect the autonomy of the others (i.e., by using negative politeness tactics such as hedges/subjunctives to reduce the threat of the acts), they seemed more likely to attempt to build and sustain a sense of group commitment by using

positive politeness tactics. Indeed, among the five most frequently used tactics, three were from the positive politeness category: jargon/metaphor (found in 92% of messages), vocatives (35%) and inclusive pronouns (26%).

Jargon/metaphor, that is, abbreviations or use of technical terms, could be seen in almost every message in the sample. Here are a few examples (with bold type indicating jargon). “*What should it be Arabic letter on my **EUC-K*R** environment, even without **utf8-aware**?*” “*I cannot make the fix myself because I do not have **cvs**.*” “*I am working on a pure **Gtk2.0/Gnome2.0** platform*”. We argued earlier that jargon/metaphor builds connections by communicating commonalities among team members. Team members seek common ground by expressing shared background in knowledge and expertise, thus contributing to reducing social distance [68]. However, the high frequency of jargon/metaphor use in these two projects made us think another functional possibility that jargon/metaphor might serve: as a necessary technical skill for team members to join in the discussion of software development. Future research should therefore distinguish jargon as generally-used programming terms from jargon as project-specific language.

Vocatives means referring to participants by name or address to a specific person, as showed in the following examples: “***Glen**’s suggestion of I Seek You is alright*” and “*I trust **Simon**’s code far more than an unknown.*” The use of vocatives again indicates familiarity among team members. FLOSS members also frequently used inclusive pronouns such as “we”, “us” and “our”, as in these messages: “***We** will certainly NOT be using “libyahoo2”*” and “*when I get some spare time, I’m going to make a UI for it (and for MSN) more befitting to conferencing than **our** plain-ol chat UI.*” Using these tactics created intimacy among team members and helped build solidarity in the teams. This observation is consistent with Bagozzi and Dholakia

[9]'s argument about the importance of we-intention in Linux user groups, i.e., when individuals think themselves as “us” or “we” and so attempt to act in a joint way.

In the negative politeness category, hedges/subjunctives was the most frequently used tactic, found in more than half of the messages (58%). Here are a few examples. “*I **would** guess that means that...*” “*This **likely** needs to be changed as well.*” “*It **would be** really nice to ditch the old model of hundreds of #ifdef’s and instead go with a layered approach.*” By employing this and other negative politeness tactics, members redressed the force of a FTA and kept proper distance from others to minimize the chance of intrusion and offense.

Given the various discontinuities that characterize virtual collaboration environments [69], one might expect members would try to reduce distances created by these discontinuities, that is, use positive rather than negative politeness strategies. For example, in a study of politeness strategies in collaborative email exchanges between students, Vinagre [70] found that only 3.6% of the examined messages used negative politeness tactics, while 94.3% used positive politeness tactics. Our observation of a relatively high use of negative politeness tactics contrasts with this study. One possible explanation is that the membership in FLOSS development teams is fluid and voluntary, with new members joining and old members quitting at any time. Further, these members might not have met before and probably will not meet in person even after they join the team. Under these circumstances, people are reluctant to impinge on each other’s time. Therefore, people need not only actions that can take them together and foster closeness, but also actions that keep their independence and freedom from imposition. This observation adds support to Morand and Ocker [34], who proposed that the maintenance of harmonious social relations in CMC depends not only upon the exchange of positive messages but also on negative politeness.

RQ2: What differences in group maintenance behaviour exist in general and over time between projects that sustain development and those that eventually cease development?

We compared communications from Fire and Gaim to identify differences in the use of group maintenance tactics. Our results suggested that the two projects did show differences in the frequency of use of a number of group maintenance tactics (Table 3). Considering the codes that were frequently used as showed in Table 2, an interesting observation is that one might expect developers in Gaim to use more positive politeness tactics than Fire, since it continued development. However, our results indicated that in general developers in Fire used more positive politeness tactics (i.e., complimenting and appreciation) and less negative politeness tactics (i.e., hedges/subjunctives) than in Gaim. It seems that to maintain group cohesiveness, developers in Fire used more tactics that help create cohesive relationships while those in Gaim used more tactics that tried to keep distance from each other to protect each member's independence. Since Fire ceased development later on, we suspect that this observation might mainly happen in the second period when Fire and Gaim showed difference in development status.

A further across-period examination confirmed our expectations that Fire and Gaim were initially mostly similar in the use of tactics to build and maintain relationships among members, but over time, as Fire slowed its production, differences in group maintenance behaviors developed. In the first period we investigated, Gaim only demonstrated more usage than Fire of one type of group maintenance behavior, humor. However in the second period, Gaim appeared to use more hedges/subjunctives and disclaimers but less appreciation and formal verbiage than Fire. This pattern of initial similarity and later difference suggests that both group maintenance behaviors and team performance are reflective of underlying group processes.

If we only examine tactics that were frequently used (those boldfaced in Table 2), this analysis showed a same trend as the results from the overall comparison between Gaim and Fire⁵. That is, it seemed that members in Gaim used more negative politeness tactics (i.e., hedges/subjunctive and disclaimers) but less positive politeness tactics (i.e., appreciation) than those in Fire. These results suggest that as the project developed, members in Gaim exhibited more use of tactics to keep distance from other and respect others' independence, while members in Fire seemed to use more tactics to bond people together.

One possible explanation might be that as Fire began to cease production, fewer and fewer people participated in development (for reasons that we do not explore in this research). As a result, the remaining members were more familiar with each other and experienced higher interdependence [71]. Therefore, it became unnecessary for members to frequently use negative politeness tactics that keep proper distance from one another. The higher use of appreciation in Fire may reflect members' expressions of appreciation for the work that was still being contributed, perhaps in the hopes of encouraging additional contribution. However in Gaim, the project was still growing and attracting new members to join the discussion. So members would use negative politeness tactics to avoid offending others and to show politeness, especially to unfamiliar newcomers. Thus the use of these visible linguistic tactics provides insight into the processes within the projects. In particular, our findings suggest the important role of negative politeness tactics in maintaining relationships among FLOSS team members over time.

⁵ Due to different sample sizes, there are differences about the individual indicators that showed significant differences in the two analyses.

6. Implications, Limitations and Conclusions

6.1 Theoretical Implications

This study begins to examine the important question of group maintenance behavior in FLOSS development projects. To our knowledge, no empirical work has been done to understand how project members maintain social relations with one another through everyday communication in FLOSS development. By content analysis of email messages from the public mailing lists of two FLOSS projects, this study provides initial insights into the specific nature of group maintenance behaviors in this setting. As well, it demonstrates that members in projects with different development status (i.e., continued production vs. ceased development) show different patterns of group maintenance tactics. For example, the project that continued production seemed to keep constant or increased use of negative politeness tactics while the project that ceased development grew to use fewer of them. Such in-depth examination enhances our understanding of social/emotional behaviors in FLOSS projects, which stands in contrast to prior research that has mainly focused on task-related behaviors [e.g., 26, 72, 73].

6.2 Methodological Implications

A majority of prior FLOSS research has used either narrative case study or quantitative survey methods [10]. In contrast, this study demonstrates a different approach, namely a linguistic analysis of everyday online communications. The FLOSS setting is heavily reliant on text communication and so provides a unique opportunity to observe and identify different group behaviors through detailed linguistic analysis of what actually happens in everyday communication, the venue for practice in FLOSS projects [29]. This approach thus provides a way to answer von Hippel and von Krogh's call for more research attention to interpretation of subtle matters in FLOSS project activities from contextual and behavioral perspectives [7].

6.3 Practical Implications

This study also has practical implications for FLOSS practitioners. FLOSS practitioners should be aware of the importance of various communication tactics for maintaining interpersonal relationships, as expressed in everyday online communication. Our results suggested members frequently employed a number of tactics for emotional expression, positive politeness and negative politeness. These tactics not only foster closeness, solidarity and cohesion in groups, but also show respect for one another's independence and freedom from imposition. Hence, the descriptive analysis of group maintenance tactics in this research provides a guideline of how to interact with others in a text-based environment for FLOSS practitioners.

The results further suggest that as a project began to cease development, members seemed to use different types of group maintenance tactics from those in a project that continued development. So FLOSS practitioners, especially those who have leadership roles, might want to monitor members' communication styles and language used, which might give them hints about the development status of the project from a social-emotional perspective, to complement other indicators of project activity and success. In addition, project leaders might want to encourage the use of particular tactics. For example, if leaders felt that their project needs more participants, they might encourage the use of more negative politeness tactics that should make the project more attractive to potential volunteers.

6.4 Limitations and Future Work

A major limitation of our research is the small sample of projects. Fire and Gaim are only two out of hundreds or thousands of active FLOSS projects and represent a small sample from which to draw conclusions, leading to questions about the generalizability of our findings.

Understanding two projects in detail was necessary at this initial stage of research to develop a reliable and valid coding scheme and to do the coding needed to provide initial evidences for our research questions. Still, the small sample suggests that results should be interpreted cautiously.

To address this limitation, future research should apply the framework of this research to communications from a larger and more representative sample of FLOSS projects. Of course, to increase the sample size will require methodological innovations to reduce the cost of the necessary linguistic analysis. A particularly promising approach is the application of computerized analysis approaches [e.g., 74, 75-77]. For example, Crowston, et. al [78] applied natural language processing technique to automate content analysis and achieved good performance on a number of codes in a pilot case study. The current study provides a tested content analysis codebook and a sample of coded data, both of which are necessary prerequisites for future automated analyses.

A further limitation is that the present study focuses on FLOSS teams that develop the same kind of software (i.e., instant messaging clients) and from the same development platform (SourceForge). It does not consider potential effects of project-level factors (e.g., project size, leadership types, member roles, software types or different technologies used) on group maintenance behaviors. These factors have been found in past research to have implications for FLOSS projects. For example, research has found that shared leadership in FLOSS projects encourages voluntary participation and facilitates the growth and development of the project [79, 80], suggesting additional factors to consider when developing a research model. Research might also consider the impact of more detailed features of the project work on the exhibited behaviors, such as the proportion of new members participating in the project, the rate of code production or the timing of releases, which are significant milestones in the life of a project.

Finally, our study did not examine the consequences of group maintenance behaviors beyond their relation to project continuance. Prior research has hypothesized that these behaviors serve to increase group performance by increasing group cohesion [e.g. 42], trust [e.g. 43], social identity [e.g. 44] and project participation [e.g. 45]. The FLOSS setting provides an opportunity to explicitly test some of these connections. While a number of the identified consequent factors (e.g., cohesion and trust) are difficult to identify from the email messages we studied, other outcomes, such as the level of contribution, are more apparent. Group maintenance behaviors could be linked to subsequent levels of contribution to the group for a more fine-grained analysis. As well, future research should seek to develop new measures for factors such as social identity that leave observable traces in the developers' communications in order to test those theoretical linkages. Finally, as group maintenance behaviors can be assessed unobtrusively, they may be a useful addition to studies that primarily employ other methods, e.g., combining an assessment of group maintenance behaviours from content analysis with assessments of trust or group cohesion obtained from interviews or surveys. Again, reducing the cost of the necessary content analysis is necessary for this opportunity to be widely used.

Despite these limitations, this study provides a first exploration into group maintenance behaviors embedded in everyday communication in community-based FLOSS projects. The results revealed that group maintenance tactics are frequently used in members' communications, and different tactics seem to reflect the development status of the projects (i.e., projects with continuous development vs. projects that ceased development). It is hoped that this in-depth examination will provide insights into the distinctive nature of group maintenance behaviors in community-based FLOSS development.

Acknowledgements

The authors would like to acknowledge the contribution from Michael Scialdone, James Howison and Xue Xiao during the collection and analysis of the data. We are grateful for the comments of Nora Misiolek and Hala Annabi.

This research was partially supported by the U.S. National Science Foundation (IIS Grant 04-14468 and HSD Grant 05-27457). Kangning Wei is partially supported by the Humanity and Social Science Youth Foundation, Ministry of Education of P. R. China (No. 11YJC630219). Kevin Crowston is supported by the U.S. National Science Foundation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

- [1] Y. Awazu and K.C. Desouza, *Open knowledge management: Lessons from the open source revolution*. Journal of the American Society for Information Science and Technology, 2004. **55**(11): p. 1016.
- [2] C.M. Kelty, *Two Bits: the Cultural Significance of Free Software*. 2008, Durham, NC: Duke University Press.
- [3] G.K. Lee and R.E. Cole, *From a firm-based to a community-based model of knowledge creation: The case of the Linux kernel development*. Organization Science, 2003. **14**(6): p. 633-649.
- [4] G. Hertel, S. Niedner, and S. Herrmann, *Motivation of software developers in Open Source projects: an Internet-based survey of contributors to the Linux kernel*. Research Policy, 2003. **32**(7): p. 1159-1177.
- [5] I. Chengalur-Smith, A. Sidorova, and S.L. Daniel, *Sustainability of free/libre open source projects: A longitudinal study*. Journal of the Association for Information Systems, 2010. **11**(11): p. 5.
- [6] Y. Fang and D. Neufeld, *Understanding Sustained Participation in Open Source Software Projects*. Journal of Management Information Systems, 2009. **25**(4): p. 9-50.
- [7] E. von Hippel and G. von Krogh, *Open source software and the "private-collective" innovation model: Issues for organization science*. Organization Science, 2003. **14**(2): p. 209-223.
- [8] B. Xu, D.R. Jones, and B. Shao, *Volunteers' involvement in online community based software development*. Information & Management, 2009. **46**(3): p. 151-158.
- [9] R.P. Bagozzi and U.M. Dholakia, *Open source software user communities: A study of participation in Linux user groups*. Management Science, 2006. **52**(7): p. 1099--1115.
- [10] K. Crowston, K. Wei, J. Howison, and A. Wiggins, *Free/Libre Open Source Software: What we know and what we do not know*. ACM Computing Surveys, 2012. **44**(2).

- [11] C. Jensen and W. Scacchi. *Collaboration, Leadership, Control, and Conflict Negotiation and the Netbeans.org Open Source Software Development Community*. in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*. 2005. Big Island, HI, USA.
- [12] M. Den Besten, J.-M. Dalle, and F. Galia, *The allocation of collaborative efforts in open-source software*. *Information Economics and Policy*, 2008. **20**(4): p. 316-322.
- [13] K. Crowston and B. Scozzi, *Bug fixing practices within free/libre open source software development teams*. *Journal of Database Management (JDM)*, 2008. **19**(2): p. 1-30.
- [14] M. Ridley, *The Origins of Virtue: Human Instincts and the Evolution of Cooperation*. 1996, New York: Viking.
- [15] J. Birnholtz and S. Ibara. *Tracking Changes in Collaborative Writing: Edits, Visibility and Group Maintenance*. in *the 2012 ACM Conference on Computer Supported Cooperative Work. CSCW 2012*. 2012. Seattle, WA, USA.
- [16] G. Yukl, *Leadership in Organizations*. 2005, Upper Saddle River, NJ: Prentice Hall.
- [17] J.R. Hackman and C.G. Morris, *Group tasks, group interaction process, and group performance effectiveness: A review and proposed integration*, in *Group Processes, volume 8 of Advances in Experimental Social Psychology*, L. Berkowitz, Editor. 1978, Academic Press: New York. p. 45-99.
- [18] D.G. Bowers and S.E. Seashore, *Predicting Organizational Effectiveness with a Four-Factor Theory of Leadership*. *Administrative Science Quarterly*, 1966. **11**(2): p. 238-263.
- [19] M.E. Warkentin, L. Sayeed, and R. Hightower, *Virtual Teams versus Face-to-Face Teams: An Exploratory Study of a Web-based Conference System*. *Decision Sciences*, 1997. **28**(4): p. 975-996.
- [20] D. Tjosvold, M. Poon, and Z.-y. Yu, *Team Effectiveness in China: Cooperative Conflict for Relationship Building*. *Human Relations*, 2005. **58**(3): p. 341-367
- [21] K.Y. Ng and L.V. Dyne, *Antecedents and Performance Consequences of Helping Behavior in Work Groups*. *Group & Organization Management*, 2005. **30**(5): p. 514-540.
- [22] K.J. Stewart and S. Gosain, *The Impact of Ideology on Effectiveness in Open Source Software Development Teams*. *MIS Quarterly*, 2006. **30**(2): p. 291-314.
- [23] A. Aksulu and M.R. Wade, *A Comprehensive Review and Synthesis of Open Source Research*. *Journal of the Association for Information Systems*, 2010. **11**(11): p. Article 6.
- [24] G. von Krogh, S. Spaeth, and K.R. Lakhani, *Community, joining, and specialization in open source software innovation: a case study*. *Research Policy*, 2003. **32**(7): p. 1217-1241.
- [25] S.K. Shah, *Motivation, governance, and the viability of hybrid forms in open source software development*. *Management Science*, 2006. **52**(7): p. 1000-1014.
- [26] N. Ducheneaut, *Socialization in an Open Source Software Community: a Socio-Technical Analysis*. *Computer Supported Cooperative Work*, 2005. **14**(4): p. 323-368.
- [27] W. Scacchi, *Free/Open Source Software Development: Recent Research Results and Methods*. *Advances in Computers*, 2007. **69**: p. 243-295.
- [28] M. Bergquist and J. Ljungberg, *The Power of Gifts: Organising Social Relationships in Open Source Communities*. *Information Systems Journal*, 2001. **11**(4): p. 305-320.
- [29] E. Monteiro, T. Østerlie, K.H. Rolland, and E. Røyrvik, *Keeping it going: The Everyday Practices of Open Source Software*, 2004, Working paper. Department of Computer and Information Science, Norwegian Univ. of Science and Technology, Trondheim, Norway.
- [30] J. Barney and T. Felin, *What are Microfoundations?* *The Academy of Management Perspectives*, 2013. **27**(2): p. 138-155.
- [31] S.W. Kozlowski and G.T. Chao, *The dynamics of emergence: cognition and cohesion in work teams*. *Managerial and Decision Economics*, 2012. **33**(5-6): p. 335-354.
- [32] K. Crowston, H. Annabi, and J. Howison. *Defining Open Source Software project success*. in *Proceedings of the 24th International Conference on Information Systems (ICIS 2003)*. 2003.
- [33] S. Krishnamurthy, *Cave or community: An empirical examination of 100 mature Open Source projects*. *First Monday*, 2002. **7**(6).

- [34] D.A. Morand and R.J. Ocker. *Politeness Theory and Computer-Mediated Communication: A Sociolinguistic Approach to Analyzing Relational Messages*. in *Proceedings of the 36th Hawaii International Conference on System Sciences* 2003. Big Island, HI, U.S.A.
- [35] S.S. Kahai and R.B. Cooper, *Exploring the core concepts of media richness theory: The impact of cue multiplicity and feedback immediacy on decision quality*. *Journal of Management Information Systems*, 2003. **20**(1): p. 263-300.
- [36] W.K. Tan, C.H. Tan, and H.H. Teo, *Conveying information effectively in a virtual world: Insights from synthesized task closure and media richness*. *Journal of the American Society for Information Science and Technology*, 2012. **63**(6): p. 1198-1212.
- [37] J.B. Walther, *Interaction Through Technological Lenses Computer-Mediated Communication and Language*. *Journal of Language and Social Psychology*, 2012. **31**(4): p. 397-414.
- [38] R. Garrison, T. Anderson, and W. Archer, *Critical thinking in a text-based environment: Computer conferencing in higher education*. *The Internet and Higher Education*, 2000. **2**(2-3): p. 87-105.
- [39] L. Rourke, T. Anderson, D.R. Garrison, and W. Archer, *Assessing Social Presence in Asynchronous, Text-based Computer Conferencing*. *Journal of Distance Education*, 1999. **14**(2): p. 50-71.
- [40] F. Biocca, C. Harms, and J.K. Burgoon, *Toward a More Robust Theory and Measure of Social Presence: Review and Suggested Criteria*. *Presence: Teleoperators and Virtual Environments*, 2003. **12**(5): p. 456-480.
- [41] J. Short, E. Williams, and B. Christie, *The Social Psychology of Telecommunications*. 1976, New York: John Wiley.
- [42] D.S. Stein, C.E. Wanstreet, H.R. Glazer, C.L. Engle, R.A. Harris, S.M. Johnston, M.R. Simons, and L.A. Trinko, *Creating shared understanding through chats in a community of inquiry*. *The Internet and Higher Education*, 2007. **10**(2): p. 103-115.
- [43] P.B. Lowry, D. Zhang, L. Zhou, and X. Fu, *Effects of Culture, Social Presence, and Group Composition on Trust in Technology-Supported Decision-Making Groups*. *Information Systems Journal*, 2010. **20**(3): p. 297-315.
- [44] K.N. Shen, A.Y. Yu, and M. Khalifa, *Knowledge contribution in virtual communities: accounting for multiple dimensions of social presence through social identity*. *Behaviour & Information Technology*, 2010. **29**(4): p. 337-348.
- [45] K.N. Shen and M. Khalifa, *Exploring Multidimensional Conceptualization of Social Presence in the Context of Online Communities*. *International Journal of Human-Computer Interaction*, 2008. **24**(7): p. 722-748.
- [46] C. Pavitt and E. Curtis, *Small group discussion: A theoretical approach*, 1994, Gorsuch Scarisbrick (Scottsdale, Ariz.).
- [47] Y. Yoo and M. Alavi, *Media and Group Cohesion: Relative Influences on Social Presence, Task Participation, and Group Consensus*. *MIS Quarterly*, 2001. **25**(3): p. 371-390
- [48] G. Cui, B. Lockee, and C. Meng, *Building modern online social presence: A review of social presence theory and its instructional design implications for future trends*. *Education and information technologies*, 2013. **18**(4): p. 661-685.
- [49] P. Rogers and M. Lea, *Social presence in distributed group environments: The role of social identity*. *Behaviour & Information Technology*, 2005. **24**(2): p. 151-158.
- [50] P. Brown and S. Levinson, *Politeness: Some Universals in Language Usage*. 1987, Cambridge, United Kingdom: Cambridge University Press.
- [51] A.J. Meier, *Passages of Politeness*. *Journal of Pragmatics*, 1995. **24**(4): p. 381-392.
- [52] D.A. Morand, *Dominance, Deference, and Egalitarianism in Organizational Interaction: A Sociolinguistic Analysis of Power and Politeness*. *Organization Science*, 1996. **7**(5): p. 544-556.
- [53] T. Holtgraves, *Social psychology, cognitive psychology and linguistic politeness*. *Journal of Politeness Research. Language, Behaviour, Culture*, 2005. **1**(1): p. 73-93.

- [54] T. Holtgraves, *The linguistic realization of face management: Implications for language production and comprehension, person perception, and cross-cultural communication*. *Social psychology quarterly*, 1992. **55**(2): p. 141–159.
- [55] K.W. Duthler, *The politeness of requests made via email and voicemail: Support for the hyperpersonal model*. *Journal of Computer-Mediated Communication*, 2006. **11**(2): p. 500–521.
- [56] R.K. Yin, *Case Study Research: Design and Methods*. 2003, Thousand Oaks, CA: SAGE Publications.
- [57] M.D. Myers, *Qualitative Research in Information Systems*. *Management Information Systems Quarterly*, 1997. **21**(2): p. 241-242.
- [58] M.B. Miles and A.M. Huberman, *Qualitative Data Analysis: An Expanded Sourcebook*. 2nd ed. 1994, Thousand Oaks: Sage Publications.
- [59] D. Riffe, S. Lacy, and F.G. Fico, *Analyzing media messages: Using quantitative content analysis in research*. 2005: Psychology Press.
- [60] R. Budd, R.K. Thorpe, and L. Donohue, *Content analysis of communication*. 1967, New York: Macmillan.
- [61] K. Krippendorff, *Content analysis: An introduction to its methodology*. 2012: Sage.
- [62] V.J. Duriau, R.K. Regeer, and M.D. Pfarrer, *A content analysis of the content analysis literature in organization studies: Research themes, data sources, and methodological refinements*. *Organizational Research Methods*, 2007. **10**(1): p. 5-34.
- [63] O. Oh, M. Agrawal, and H.R. Rao, *Community Intelligence and Social Media Services: a Rumor Theoretic Analysis of Tweets During Social Crises* *MIS Quarterly*, 2013. **37**(2): p. 407-A7.
- [64] K.A. Neuendorf, *The Content Analysis Guidebook*. 2002, Thousand Oaks, CA: Sage Publications.
- [65] K. Crowston, K. Wei, J. Howison, and A. Wiggins, *Free/Libre Open Source Software Development: What we know and what we do not know*. *ACM Computing Surveys*, 2012. **44**(2): p. Article 7.
- [66] D. Derks, A.E.R. Bos, and J. von Grumbkow, *Emoticons and social interaction on the Internet: the importance of social context*. *Computers in Human Behavior*, 2007. **23**: p. 842-849.
- [67] D. Derks, A.H. Fischer, and A.E.R. Bos, *The role of emotion in computer-mediated communication: A review*. *Computers in Human Behavior*, 2008. **24**(3): p. 766-785.
- [68] J.-R. Park, *Linguistic Politeness and Face-Work in Computer Mediated Communication, Part 2: An Application of the Theoretical Framework*. *Journal of the American Society for Information Science and Technology*, 2008. **59**(14): p. 2199-2209.
- [69] M.B. Watson-Manheim, K. Crowston, and K.M. Chudoba. *a New Perspective on "Virtual": Analyzing Discontinuities in the Work Environment*. in *35th Hawaii International conference on System Sciences*. 2002. Big Island, Hawaii.
- [70] M. Vinagre, *Politeness strategies in collaborative e-mail exchanges*. *Computers & Education*, 2008. **50**: p. 1022-1036.
- [71] J.G. March, *Exploration and exploitation in organizational learning*. *Organization science*, 1991. **2**(1): p. 71-87.
- [72] G. Kuk, *Strategic Interaction and Knowledge Sharing in the KDE Developer Mailing List*. *Management Science*, 2006. **52**(7): p. 1031-1042.
- [73] A. Mockus, R.T. Fielding, and J.D. Herbsleb, *Two case studies of open source software development: Apache and Mozilla*. *Acm Transactions on Software Engineering and Methodology*, 2002. **11**(3): p. 309-346.
- [74] F. Thung, D. Lo, and L. Jiang. *Automatic Defect Categorization*. in *2012 19th Working Conference on Reverse Engineering (WCRE)*. 2012. IEEE.
- [75] P.K. Prasetyo, D. Lo, P. Achananuparp, Y. Tian, and E.-P. Lim. *Automatic classification of software related microblogs*. in *Software Maintenance (ICSM), 2012 28th IEEE International Conference on*. 2012. IEEE.
- [76] D. Surian, D. Lo, and E.-P. Lim. *Mining collaboration patterns from a large developer network*. in *Reverse Engineering (WCRE), 2010 17th Working Conference on*. 2010. IEEE.

- [77] F. Thung, T.F. Bissyandé, D. Lo, and L. Jiang. *Network Structure of Social Coding in GitHub*. in *Software Maintenance and Reengineering (CSMR), 2013 17th European Conference on*. 2013. IEEE.
- [78] K. Crowston, E.E. Allen, and R. Heckman, *Using Natural Language Processing Technology for Qualitative Data Analysis*. *International Journal of Social Research Methodology*, 2012. **15**(6): p. 523-543.
- [79] J. Mateos-Garcia and W.E. Steinmueller, *The institutions of open source software: Examining the Debian community*. *Information Economics and Policy*, 2008. **20**(4): p. 333-344.
- [80] R.T. Fielding, *Shared leadership in the Apache Project*. *Association for Computing Machinery. Communications of the ACM*, 1999. **42**(4): p. 42-43.

Appendix 1. Actions Taken in Coding Scheme Development Process

Actions Performed	Indicators from Literature	New Indicators	Reasons for the Change		
Removed	self-disclosure ¹	N/A	Very low or no occurrence in the messages.		
	notice hearer's admirable qualities or possessions, show interest, exaggerate ²				
	Ellipsis ²				
	claim common view ²				
	give reasons ²				
	assert reciprocal exchange or tit for tat ²				
	give something desired ²				
	be conventionally indirect ²				
	give deference by using honorifics ²				
	impersonalize the speaker and hearer by avoiding the pronouns (I and you) ²				
	use the past tense to create distance in time ²				
	nominalize to diminish speakers' active participation ²				
	continuing a thread ¹			N/A	More a result of using software (i.e., email) features or normal communication than a purposeful behavior of maintaining interaction.
	quoting from others' messages ¹				
asking questions ¹					
Added	N/A	Jargon /metaphor participation	New group maintenance behaviors observed in FLOSS context.		
Combined	vocatives ¹	vocatives	They are very similar in nature, or they serve the same purposes, often co-occur in a thematic unit, and are hard to be distinguished within one particular thematic unit.		
	referring explicitly to others' messages ¹	Slurring /colloquialisms			
	phonological slurring ²	Hedges /subjunctive			
	colloquialisms or slang ²	disclaimers			
	use hedges ²				
	use subjunctive ²				
	use words or phrases that minimize the imposition ²				
Apologize ²					
Split	Complimenting, expressing Appreciation ¹	Complimenting	They might convey different meanings if treated as a whole.		
		Appreciation			

¹ Items from [39].

² Items from [34].

Appendix 2. Coding Scheme of Group Maintenance Behaviors

Indicator	Definition	Examples
Emotional Expression		
Emoticon /capitalization /punctuation	Expressions of emotion or emphasis using emoticons, conspicuous capitalization and repetitious punctuation, exclamation point, underlining, italic fonts, or any other.	:) ;-) “EVERYONE ON THE LIST” “!!!”
Humor	Expression of humor using teasing, cajoling, irony, understatements, sarcasm.	“The only way to keep your health is to eat what you don’t want, drink what you don’t like, and do what you’d rather not’. -- Mark Twain”
Positive Politeness		
Colloquialisms /slang	Spelling out phonological slurring, using colloquialisms or slang; beyond group specific; used to show familiarity.	“Saturdayish” “BTW”
Jargon /metaphor	Use of group-specific jargon, language, or metaphors.	“Why is this a .mm file? what is .mm again? I know .m is ObjC”
Vocatives	Referring to participants by name, or specifically addressing to an individual.	“As sean* said” “Martin, ...”
Inclusive Pronouns	Incorporating writer and recipient(s)	“we”, “us”, “let’s”, “our”
Phatics	Personal greetings and closures	“Hi”, “regards”,
Complimenting	Complimenting others or message content.	“You guys have done an awesome job”
Agreement	Expressing agreement with others’ previous statements	“Agreed” “I suppose.” “Correct.”
Participation	Encouraging others to participate	“Any comments welcome.”
Appreciation	Expressing appreciation for other’s effort	“Well thanks a lot for you hard work!”
Negative Politeness		
Disclaimers	Use of disclaimers prior to an FTA; self-depreciation as a distancing tool; may include apologies as explanations; express reluctance	“dumb fire question#1: which MSNService.nib “file” is the real one?” “Sorry if I’m terribly ignorant somehow.. I’m just getting into this stuff.”
Rational for FTA	Stating an FTA as a general rule to minimize impact or as to not single out an individual; Explaining the reasons behind an action that might threat someone’s face.	“In general we want to avoid forking the MSN library with our own changes so any changes there need to be sent on to Meredydd.”
Hedges /subjunctives	Use of words/phrases/subjunctives to diminish force of act; Use of hesitation in disagreement (ie. “well...”)	“um...” “I’m not sure what the problem is...” “it would be nice to at least..”
Formal Verbiage	Using formal wording choices	“please send the file to ...”

*All names quoted in this table are pseudonyms to protect subject privacy.