The impact of agile practices on communication in software development

Article *in* Empirical Software Engineering · June 2008

DOI: 10.1007/s10664-008-9065-9 · Source: DBLP

Minna Pikkarainen
OsloMet – Oslo Metropolitan University

Pekka Abrahamsson
Tampere University

Roles and Politeness Behavior in Community-Based Free/Libre Open Source Software Development

Kangning Wei
School of Management
Shandong University
Jinan, Shandong, 250100 China

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

U. Yeliz Esreyel
Faculty of Economics and Business
University of Groningen
PO Box 800
9700 AV Groningen, The Netherlands

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

Roles and Politeness Behavior in Community-Based Free/Libre Open Source Software Development

Abstract

Community-based Free/Libre Open Source Software (FLOSS) development relies on contributions from both core and peripheral members. Prior research on core-periphery has focused on software coding-related behaviors. We study how core-periphery roles are related to social-relational behavior in terms of politeness behavior. Data from two FLOSS projects suggest that both core and peripheral members use more positive politeness strategies than negative strategies. Further, core and peripheral members use different strategies to protect positive face in positive politeness, which we term respect and intimacy respectively. Our results contribute to FLOSS research and politeness theory.

Keywords: Open source software development; Core-periphery structure; Politeness behavior

1. Introduction

In recent years, Free/Libre Open Source Software (FLOSS) projects have received much attention as successful examples of open innovation [1]. Many of these projects are developed in a community-based form, that is, developed and maintained by teams of independent volunteer developers who are organizationally- and geographically-distributed. In community-based FLOSS projects, FLOSS teams are largely decentralized and self-organized, without a formal hierarchy and with non-coercive leadership structures [2]. This kind of FLOSS team has attracted great interest among researchers who seek to understand this novel model of organizing, often with an interest in transferring the model to other self-organizing settings [e.g., 3, 4].

Though community-based FLOSS projects do not have formal hierarchies that are imposed by external forces, members have different levels of participation in FLOSS development and so naturally take on different roles [5]. A widely accepted view of roles in community-based FLOSS teams is the core-periphery structure [6-8]. For example, Crowston and Howison [9] see community-based FLOSS teams as having an onion-like core-periphery structure, in which the core category includes core developers and the periphery includes co-developers, active users and other registered users including newcomers. Rullani and Haefliger [10] described periphery as a “cloud” of members that orbits around the core members of open source software development teams.

Generally speaking, access to core roles is based on technical skills demonstrated through the development tasks that the developer performs [11]. Core developers usually contribute most of the code and oversee the design and evolution of the project, which requires a high level of technical skills [9]. Core developers are usually also the top contributors to the projects and so have been a primary focus of FLOSS research [12]. Peripheral members, on the other hand, contribute at a lower level, e.g., by submitting patches such as bug fixes (e.g., co-developers), which provide an opportunity to demonstrate skills and interests, or just providing use cases and bug reports as well as testing new releases without contributing code directly (e.g., active users), which require less technical skills [9].

Despite the difference in technical contributions to the projects, both core and peripheral members are important to the success of the project. It is evident that, by making direct contributions to the software developed, core members are vital to the project development. Peripheral members, even though they contribute only sporadically, provide bug reports, suggestions and critical expertise, which are fundamental for innovation [10]. In addition, the periphery is the source of new core members [13, 14], so maintaining a strong periphery is important to the long-term success of the project. Amrit and van Hillegersberg [7] examined core-periphery movement in open source projects and concluded that a steady movement toward the core is beneficial to a project, while a shift away from the core is not.

Distinct from the notion of status, roles are defined by the activities performed by the members [6]. Up to now, the few discussions of differences in core/periphery contributions have mainly focused on coding-related behaviors such as innovation [15] and division of labor [10]. However, developers do more than just coding [6]. It is important for a participant to learn both social and technical aspects of a FLOSS project before making contributions [16]. FLOSS projects cannot succeed without group efforts. Therefore, both core and peripheral members need to interact and communicate virtually with each other, engaging in social-relational behaviors in addition to task-oriented behaviors such as coding. Consideration of these non-task activities is important because effective interpersonal communication plays a vital role in the development of online social interaction [17]. Members find social support, companionship and a sense of belonging in the context of online communities [18].

For FLOSS development in particular, the health of the community is an important factor that impacts performances of FLOSS projects [9], as it is challenging to sustain a project with voluntary members over the long term [2, 19]. For example, Barcellini et al. [6] identified a socio-relational role in open source software communities, which is associated with activities (e.g., praise others for their contributions, express agreements or reduce conflict) to facilitate interpersonal relationships. Social-relational issues have been seen as a key component of achieving design effectiveness [6] and enhancing online group involvement and collaboration [20]. Therefore, it is important to understand how members of community-based FLOSS teams build and maintain relationships with each other. While there is recognition of the importance of social behaviors, we still have limited knowledge about how roles—defined by task distinctions—are related to such behaviors.

In FLOSS settings, collaborative work primarily takes place via information technologies such as asynchronous (e.g., email lists or discussion fora) and synchronous communication tools (e.g., Internet Relay Chat (IRC)) [5], systems for sharing and reviewing software (e.g., Concurrent Version System (CVS), Subversion or Git), bug trackers, project documentation systems and so on [6]. Our study explicitly focuses on the first type of information technologies, namely communication tools, since they are the main communication channels that enable social-relational interaction between core and peripheral members for their development effort, which is the focus of this paper.

Prior studies of relations in FLOSS have mostly examined patterns of interactions among participants using network-based analysis [e.g., 21, 22, 23]. However, little research has explicitly examined the content of the interaction, that is, the content of the messages sent to discussion fora or email lists by participants [6]. As a first attempt to study social-relational behavior in community-based FLOSS development teams, Wei, et al. [24] analyzed group maintenance behaviors used by members to build and maintain reciprocal trust and cooperation in their everyday interaction messages. This research found that members use a variety of group maintenance strategies composed of emotional expressions and politeness strategies. In this paper, we extend the work of Wei, et al. [24] by investigating the link between the task-oriented structure of core-periphery and social-relational behaviors. We focus on one specific type of group maintenance strategies in this study, namely politeness strategies, which are linguistic strategies used to save or promote the speaker's self-image in a communicative act [25]. In particular, we examine the following research question:

> *Do core and peripheral members in community-based FLOSS development teams engage in politeness behaviors differently? If so, how?*

The rest of the paper is organized as follows. In section 2, we introduce politeness theory as our theoretical background. Based on a review of prior research on core and peripheral members in FLOSS development, we develop our hypotheses regarding the differences between core and peripheral members in the use of politeness behaviors. Section 3 describes the research method used to examine the hypotheses and section 4 presents the results. Finally, we discuss the results and their implications and conclude the paper with limitations and future research in section 5.

2. Theoretical Background and Hypothesis Development

2.1 Politeness Theory and Politeness Strategies

In both face-to-face and virtual communications, the tone of communications is an important factor in how messages are received and interpreted and in how they advance both task and relationship building. One theoretical lens to explain this kind of behavior is politeness theory, which describes how people phrase communications in a way that takes into consideration the feeling of the others [26], thus contributing to the development of social relations. Researchers have found that politeness theory is especially useful in analyzing relational communication in computer-mediated communication (CMC) contexts, as pointed out by Morand and Ocker [25]: "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).

Politeness theory is built on two concepts: face and face threatening acts [27]. Face is the central element in politeness theory and is defined as the positive value individuals claim for the public self they present [25]. Because face is emotionally charged and is inherently vulnerable when engaging others in interaction, people strive to maintain face in social settings and communications [28]. Face is constructed of two wants: autonomy of action (also known as negative face) and the need for validation (also known as positive face) [27]. 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 wanting respect, membership in a valued community and a reputation for competence and fairness [29].

However, face—the value that one claims for one's self—can only be validated by others and so is dependent on others. Thus everyone with an interest in maintaining the group also needs to be concerned with maintaining the face of those with whom she or he interacts [30]. Face is therefore viewed as "a social rather than a psychological construct" [31]. And it is within these social situations that people continuously interact in ways that preserve, bolster or show consideration for the face of others [28]. Thus, politeness theory emphasizes interactional support work directed toward others' face [25], which are known as politeness strategies.

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" [29]. These instances are known as face threating acts (FTAs), and can either be directed toward the speaker or the hearer, and can threaten both types of face, positive and negative [27]. Therefore, two forms of linguistic politeness strategies, positive strategies (to encourage positive face) and negative strategies (to encourage negative face), are taken to redress or mitigate any threats to others' face engendered by an FTA [25].

Positive politeness strategies attend to the hearer's positive face desire by treating the hearer as someone who is liked or esteemed [32]. Examples of positive politeness tactics include use of colloquialisms or slang, vocatives and inclusive pronouns, and expressions of agreement, understanding, cooperation and sympathy [25, 32]. These strategies demonstrate intimacy, proximity, friendly attitude and claim common ground between the speaker and the hearer [33].

Conversely, negative politeness strategies try to maintain relationships by attending to the hearer's direct for negative face [32, 33]. Negative politeness is realized through strategies such as hedges, indirect inquiries, subjunctives, honorifics, apologies, formal verbiage, passive voice and expressed rationale for FTAs [25, 28]. These strategies demonstrate the speaker's wish not to be seen as imposing upon the hearer [32].

Research has found that in CMC contexts, both positive and negative politeness strategies are used, although they might be used at different frequencies. For example, by studying a real-time, online discourse of K-12 students, Park [17] found that students seldom used negative politeness strategies. Rather, they use more positive strategies such as seeking of common ground and agreement and in-group language use. By analyzing two community-based FLOSS projects, Wei, et al. [24] found that members of the studied FLOSS projects used both positive and negative politeness tactics, with an emphasis on the former.

2.2 Core vs. Periphery in FLOSS Development

Prior research on core-periphery structure in FLOSS development has examined how core and peripheral members behave differently in development-related activities. The general findings suggest participation inequality between core and periphery in these activities. For example, Luthiger Stoll [34] found that core members make different time commitment than peripheral members. In their study, core participants spent an average of 12.15 hours per week, with project leaders averaging 14.13 hours, and bug-fixers and otherwise active users only around 5 hours per week. Similarly, using social network analysis, Toral et al. [23] found that a few core members posted the majority of messages and acted as middlemen or brokers among other peripheral members. Core members and peripheral members also had different power over the projects that lead to different team behaviors, as participants nearer to the core have greater control and discretionary decision-making authority compared to those further from the core [35, 36].

Another important feature of the core-periphery structure is that the structure is not static over time [8], as a developer may change roles over time. A peripheral member may move into the core, or a core member may reduce engagement and become a peripheral member or even leave the project altogether [37]. A number of studies have examined this movement, especially how a peripheral member, specifically a newcomer, becomes a core member through socialization [16]. For example, in a study of socialization in the Freenet project, von Krogh, et al. [14] discovered that peripheral members who eventually became core members differed from those that stayed in a peripheral role in both the level and the type of activities they conducted. In a study of Apache Software Foundation projects, Gharehyazie, et al. [38] found that the likelihood of peripheral members becoming committers could be predicted by the amount of two-way communication in which they participated (i.e., the number of messages they responded to and the number of messages they received in response to their own messages). Again, most of the analysis focuses on software development activities, such as the number of emails sent to the developer list and the type of questions asked in the list.

Considering the inequality in participation between core and peripheral members in code-related behaviors, we expect participants in different roles also engage in different politeness behaviors when communicating with each other online. Core members have a stronger attachment to their online communities and research has found that core members are likely to take a social role in online communities and offer camaraderie to their members [18]. These behaviors correspond to the positive politeness behaviors noted above. Therefore, we hypothesize that:

H1: When communicating with other members online in community-based FLOSS development, core members are more likely to show positive politeness behaviors than peripheral members.

Peripheral members contribute to FLOSS projects in ways that may not be as obvious as the core members' contributions, e.g., by reporting bugs, offering comments and suggestions and providing small or ad hoc solutions relevant to the project. These communications are primarily directed to the core members, since core members are the ones who can implement these suggestions. As the peripheral members are not on an intimate footing with core members, we suggest that these interactions are more likely to be characterized by negative politeness strategies.

Sometimes, peripheral members may just ask questions or seek help for the problems they encounter when using the software. In these cases, we also expect peripheral members use more negative politeness strategies because improper use of politeness strategies can cause problems. For example, Chejnová [39] found that making requests using positive politeness strategies could cause pragmatic failures, that is, a failure to understand the intent of the request. Therefore, we hypothesize that:

H2: When communicating with other members online in community-based FLOSS development, peripheral members are more likely to show negative politeness behaviors than core members.

3. Research Method

To address the research question and test our two hypotheses, we compared politeness behaviors between core and peripheral members through a content analysis of email archives from two FLOSS teams. The following section describes our project selection strategy, data collection method, coding scheme development and coding process and data analysis methods.

3.1 Project Selection

We have selected the projects to be similar using a replication logic, "which is analogous to doing multiple experiments… The replications may attempt to duplicate the exact conditions of the original experiment…Only with such replications would the original finding be considered robust and worthy of continued investigation or interpretation", as suggested by Yin [40] (p.47). To control unwanted variance brought by systematic factors such as different types of software developed and different ages of the projects, we decided to focus on projects that were developing similar software at similar levels of development. Specifically, we selected two projects that developed the same kind of software (i.e., multi-platform Instant Messaging (IM) clients) from SourceForge.net for this study: Gaim[1] and Fire. The two projects were thus similar in terms of their project goals, nature of tasks, and potential users. Fire stopped development in early 2007, which limited us to collect messages posted to the mailing lists before early 2007 for both projects.

[1] Because of a naming dispute with AOL, Gaim was renamed Pidgin in April 2007.

3.2 Data Collection

The data of this study constituted publicly available email messages from the public mail lists of the two FLOSS projects. It is possible that participants might use other communication channels such as IRC or personal emails to interact with each other, even though many projects discourage reliance on channels that are not available to all project members. However, our analysis examines differences in politeness behaviour between core and peripheral members, and we expect these differences to hold true regardless of the communication media used. Our observation of IRC channels used by Gaim confirmed our assumption. We quickly evaluated IRC channels as a potential medium for investigation by recording interaction in the IRC channel for Pidgin (new name for Gaim) for several days (Fire, having ceased development, is no longer available). We found that many members stayed logged in to the IRC channel but the activity was low compared to the activity on the listservs. Therefore, even if email were not a complete record of the project communications, it provides a suitable sample of communications with which to test the hypotheses.

The messages were sampled for Fire over 43 months between June 2002 and December 2005 before the project ceased, and for Gaim over 45 months between June 2002 and February 2006. The extended sampling frame we used enabled a thorough analysis of politeness strategies used in the daily communication. Although software development methods and tools certainly evolve over time, with this study we investigated a fundamental aspect of social interaction, a phenomenon that we do not expect to change rapidly over time. Furthermore, many FLOSS projects still use the same communication technologies, namely email lists or discussion fora. Therefore, analysis of this data should generalize to community-based FLOSS projects, which are developed and maintained by teams of independent volunteer developers where teams are decentralized, self-organized, and without formal hierarchy or coercive leadership structures.

On the one hand, availability of rich public data enables the investigation of FLOSS, but on the other hand, the availability of massive number of messages creates a methodological challenge for the researchers. Automatic analysis of thousands of email messages on linguistic symbols of politeness behaviors is still at its early development [41]. Manual analysis of thousands of email messages is infeasible due to the labor-intensiveness of the content analysis method. Thus we had to balance the need for a large sample to obtain sufficient power for the statistical tests, with the need for a smaller sample size that would make the content analysis feasible. Therefore, we developed a hierarchical sampling strategy to uniformly cover the history of the projects. This approach included dividing each email list's messages into 360 sequential message sets with equal number of messages, regardless of the duration of each message set. We completed our sampling by selecting a random message from each set, and replacing this message with another one if the sampled message did not constitute a real email by a FLOSS participant rather than a spam or an automated email forwarded to the email list.

Using this process, 360 messages were selected from the Gaim mail list and 336 messages from the Fire mail list. The sending time of each message and the sender's name were collected for identifying the message senders' roles. The descriptive statistics for the sample messages are given 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. To test the hypotheses, two types of data were needed.

The first type of data was the role of the sender of each message (i.e., core or peripheral members). Prior research has suggested several ways to identify core members in FLOSS teams. For examples, Crowston, et al. [35] introduced three methods to identify core and peripheral members in FLOSS teams: relying on project-reported formal roles, analysis of distribution of contributions based on Bradford's Law of Scatter, and core-and-periphery analysis of project social network. Their analysis showed that all 3 measures suggest that the core of community-based FLOSS projects is a small fraction of the total number of participants. Oliva, et al. [42] characterized key developers using the number of core commits (i.e., those commits that contributed to the technical core rather than just made peripheral changes) each developer made to the codebase. Although the dichotomous distinction between core and peripheral members is based on code-related activities, the types of activities are not specified in our research. Therefore, using data from only codebase or one type of interaction (e.g., bug tracker system, email lists, discussion fora and so on) might paint an incomplete picture of core members' activities. Since most projects grant developer status based on a track record of various contributions [35], in this study, we identified core members based on their self-reported roles from the project developer lists, which include project leaders/administrators and core developers. We categorized all other contributors, namely the co-developers, active users and passive registered users, as the periphery. Our approach constitutes a common way to define core members.

Acknowledging that the roles of the members may change over time, we extracted the developer lists at different time points during the period of June 2002 to February 2006 from archive.org, which has archived SourceForge project homepages from as early as August 2000. From this source we were able to get developer lists for Fire at 7 different points (Nov. 2, 2002; Jan. 15, 2003; Jun. 24, 2004; Sept. 19, 2004; Oct. 30, 2004; Mar. 04, 2005 and Apr. 05, 2005) and those for Gaim at 8 different points (Nov. 06, 2002; Mar. 01, 2003; Jun. 23, 2003; Mar. 01, 2004; May 17, 2004; Jun. 03, 2004; Apr. 18, 2005 and Dec. 13 2005). We assigned the role of core to the senders whose names were on the developer list at the time closest to the time the message was sent. We note that some members who were listed as core members by the project were not actively sending messages within the time frame of our data collection (i.e., they had essentially left the project). However, since there are no messages from these members, this kind of change (a core member leaving the project) does not result in incorrectly labelled messages.

Table 2 summarizes the numbers of core and peripheral members in Fire and Gaim separately and the numbers of messages they sent.

Table 2. Descriptive Statistics of Core and Peripheral Members

Projects	**Core**		**Periphery**	
	Number of core members	Number of messages sent by core	Number of peripheral members	Number of messages sent by periphery
Fire	7 (5%)	178 (53%)	129 (95%)	158 (47%)
Gaim	15 (10%)	153 (43%)	137 (90%)	207 (57%)
Fire & Gaim	22 (8%)	331 (48%)	266 (92%)	365 (52%)

The second type of data constituted the politeness strategies used in each message to test our hypotheses. Two independent analysts content analyzed the messages in both projects for "politeness strategies" using a subset of a coding scheme that was developed by Wei et al. [24].

In their study, Wei et al. [24] provided a comprehensive description of group maintenance behaviour in community-based FLOSS development teams in terms of emotional expression, positive politeness and negative politeness. Since this paper focuses on politeness strategies, we adopted the codes for positive and negative politeness, with one change. Specifically, we dropped jargon/metaphor from the coding scheme. Wei, et al. [24] concluded that many words that they coded as jargon were simply technical terms that team members use for normal communications, rather than some project-specific language. Their study identified the need for future research on distinguishing generally-used programming terms from project-specific language. For this study, we simply excluded jargon/metaphor from further analysis. Table 3 shows the revised coding scheme adapted from Wei, et al. [24].

Table 3. Coding Scheme of Politeness Behaviors

Category	Indicator	Definition	Example
Positive Politeness	Slurring/ Colloquialisms	Spelling out phonological slurring, using colloquialisms or slang; beyond group specific; used to show familiarity.	"Saturdayish" "BTW"
	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	Stating an FTA as a general rule to minimize impact or as to not single out	"In general we want to avoid forking the MSN library

Category	Indicator	Definition	Example
	FTA	an individual; Explaining the reasons behind an action that might threat someone’s face.	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.*

In coding the messages, we annotated the selected texts with codes for selected categories [43]. We used the “thematic unit” as the unit of coding. A thematic unit is “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” [44]. Thus our annotated texts vary in size from a word, a phrase, a part of a sentence, a sentence, or even a few sentences that capture the meaning of the code.

The initial analysis of the textual data was done by two research assistants, referred to as coders. Two or more coders are needed for each piece of text to be able to compare their separate judgements and identify possible mistakes or errors in analysis [45]. Although two coders is only the minimum requirement to determine reliability, for practical reasons, this number of coders is commonly used [e.g. 46, 47]. For example, to study citizen-driven information processing through Twitter services, Oh, et al. [47] 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 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.

The reliability of the coding schema was established by having the two coders independently code a subset of the data. We determined reliability using simple percent agreement rate [48]. Agreement is often assessed using Cohen's Kappa, which corrects for chance agreement, but this correction was not needed in this case, because the possibility of chance agreement was negligible for our coding, due to infrequent observation of the codes per thematic unit (that is, only a few words or sentences in a message constitute examples of positive or negative politeness behaviors).

To improve coding reliability, the coders independently coded pilot data, then enriched the coding scheme with notes and examples from their discussion of the coded data, thus developing an enhanced coding scheme. Using this process, inter-rater reliability reached first 80% in the second half of pilot-coding process and then 85% in the last 1/5 of the pilot-coding process. Inter-rater reliability of higher than 80% is generally considered acceptable for the research [48]. After the coding process was deemed reliable, each coder independently coded half of the remaining messages.

3.4 Statistical Analysis

After content analysis, we conducted quantitative analysis of the coded text, also known as quantitative content analysis. Quantitative content analysis allows researchers to determine specific frequencies of relevant categories and examine the relationships involving these categories using statistical methods [49]. For the quantitative analysis, the unit of analysis was the message. For each politeness strategy, we identified its rate of occurrence, i.e., the frequency of use per message, since multiple instances of a particular code per message (i.e., multiple examples of a particular politeness strategy) was possible. To investigate the possible effects of differing length of messages, we ran the same tests using behavior density, namely the count of a particular indicator (politeness strategy in this case) in a message divided by the number of words in the message [50]. However, we did not find a meaningful difference in the results of the analysis between frequency and density measures. Therefore, for simplicity of presentation, results are presented using frequency of indicator occurrence.

We conducted a series of Mann-Whitney U tests comparisons between data for messages sent by core and peripheral members to identify between-group similarities and differences in politeness behaviors. We used a non-parametric test because the counts of codes were not normally distributed. Since we were making multiple comparisons (12 comparisons, one for each indicator), one or more comparisons could 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 0.004 to declare statistical significance. This approach is conservative and does reduce the power of the statistical tests.

4. Findings

In this section, we report the comparison results between core and peripheral members in politeness behaviors. Table 5 displays the percentages of the occurrence frequencies of the politeness strategies used by core and peripheral members respectively. From the table we can see that, although both positive and negative politeness strategies were used frequently by core and peripheral members, they were used with different frequencies. Both core and peripheral members used more positive politeness strategies (85.8% and 76.4%) in their communication than negative politeness strategies (63.7% and 65.8%). Further, the use frequencies of individual positive politeness indicators were different for core and peripheral members. The most frequently used positive politeness indicators for core members are vocatives (51.4%), inclusive pronouns (44.4%), and slurring/colloquialisms (21.1%), while for peripheral members, phatics (39.7%), slurring/colloquialisms (27.9%), appreciation (24.4%) and vocatives (20.5%) are the most frequently used ones. Among negative politeness strategies, hedges/subjunctives were the most frequently used indicator for both core (57.1%) and peripheral members (58.9%).

Table 5. Descriptive Statistics of Indicators Occurrence Frequency per Message for Core and Peripheral Members and Comparisons between Them[2]

Indicator	**Mean**		**Range**		**Percentage***	
	Core	**Periphery**	**Core**	**Periphery**	**Core**	**Periphery**
Positive Politeness	**2.90**	**2.03**	**0-19**	**0-19**	**85.8%**	**76.4%**
Slurring/colloquialisms	0.27	0.36	0–4	0–6	21.1%	27.9%
Vocatives	1.18	0.73	0–10	0–15	51.4%	20.5%
Inclusive Pronouns	1.08	0.58	0–17	0–7	44.4%	10.1%
Phatics	0.09	0.27	0–2	0–3	8.8%	39.7%
Complimenting	0.02	0.03	0–2	0–3	2.1%	11.2%
Agreement	0.05	0.05	0–2	0–2	5.1%	1.9%
Participation	0.05	0.06	0–1	0–1	5.4%	5.2%
Appreciation	0.14	0.11	0–2	0–2	12.4%	24.4%
Negative Politeness	**1.81**	**1.76**	**0-40**	**0-20**	**63.7%**	**65.8%**
Disclaimers	0.11	0.18	0–3	0–3	8.5%	13.4%
Rational for FTA	0.07	0.09	0–4	0–2	5.4%	9.9%
Hedges/subjunctives	1.55	1.77	0–37	0–18	57.1%	58.9%
Formal Verbiage	0.08	0.06	0–2	0–4	7.3%	6.6%

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

A series of Mann-Whitney tests were conducted to statistically analyze differences in the use of politeness strategies between core and peripheral members. The statistical results are shown in the first column of Table 6. From the results we can see that, in general, core members used

[2] The distributions of the occurrence frequencies of each politeness strategy in messages were heavily skewed. A majority of its occurrence frequencies in messages were zero, which made the medians of all the 12 politeness strategies 0. Therefore, we did not provide medians of each politeness strategy here.

more positive politeness strategies than peripheral members did (p=0.000). However, the results show no significant difference in the use of negative politeness strategies between core and peripheral members. Since the two projects have different development status (i.e., Gaim continues development while Fire has ceased development), we conducted further analysis between core and peripheral members within each projects. Table 7 displays the means and ranges of occurrence frequencies of each politeness indicator for core and peripheral members within Fire and Gaim. The statistical results are shown in the second and third columns of Table 6. For negative politeness strategies, both projects showed no significant differences between core and peripheral members, which was consistent with the overall trend. But for positive politeness strategies, only in Fire did core members use more positive politeness strategies than peripheral members did. The data show no difference in the use of positive politeness strategies between core and peripheral members in Gaim.

Table 6. Mann-Whitney U Tests Results Showing Comparisons between Core and Peripheral Members across Gaim and Fire and within Each Project

Indicator	**First Column**			**Second Column**			**Third Column**		
	C vs. P across Gaim and Fire			**C vs. P in Fire**			**C vs. P in Gaim**		
	Mean Rank		**p value**	**Mean Rank**		**p value**	**Mean Rank**		**p value**
	C	**P**		**F-C**	**F-P**		**G-C**	**G-P**	
Positive Politeness	***376.76***	***322.87***	***0.000****	***185.56***	***149.28***	***0.001****	**188.91**	**174.28**	**0.178**
Colloquialisms/slang	334.86	360.87	0.024	164.46	173.05	0.283	170.37	187.99	0.035
Vocatives	*406.68*	*295.74*	*0.000**	*204.01*	*128.50*	*0.000**	*201.08*	*165.29*	*0.000**
Inclusive Pronouns	*412.06*	*290.84*	*0.000**	*204.58*	*127.85*	*0.000**	*205.45*	*162.06*	*0.000**
Phatics	*290.45*	*401.14*	*0.000**	*140.17*	*200.42*	*0.000**	*149.30*	*203.56*	*0.000**
Complimenting	*331.88*	*363.57*	*0.000**	*153.23*	*185.70*	*0.000**	179.05	181.57	0.425
Agreement	354.36	343.19	0.021	170.16	166.63	0.207	185.42	176.86	0.031
Participation	348.92	348.12	0.891	171.33	165.32	0.136	177.56	182.67	0.245
Appreciation	*326.98*	*368.02*	*0.000**	*155.30*	*183.37*	*0.001**	*168.95*	*189.04*	*0.000**
Negative Politeness	**344.63**	**352.01**	**0.616**	**166.25**	**171.03**	**0.635**	**181.63**	**179.63**	**0.849**
Disclaimers	339.42	356.74	0.037	162.54	175.21	0.011	179.02	181.59	0.700
Rational for FTA	340.56	355.70	0.032	165.06	172.38	0.220	176.16	183.71	0.155
Hedges /subjunctives	348.13	348.84	0.961	167.14	170.03	0.770	184.79	177.33	0.485
Formal Verbiage	349.80	347.32	0.712	171.07	165.61	0.305	177.85	179.63	0.258

Note: C indicates core; P indicates periphery;

F-C indicates core members in Fire; F-P indicates peripheral members in Fire;
G-C indicates core members in Gaim; and G-P indicates peripheral members in Gaim.
Scores in italics are statistically significantly different.

Table 7. Descriptive Statistics of Indicators Occurrence Frequency per Message for Core and Peripheral Members in Fire and Gaim

Indicator	Fire				Gaim			
	Mean		Range		Mean		Range	
	Core	Periphery	Core	Periphery	Core	Periphery	Core	Periphery
Positive Politeness	**3.38**	**1.99**	**0-19**	**0-8**	**2.35**	**2.05**	**0-19**	**0-19**
Colloquialisms/slang	0.27	0.43	0-3	0-6	0.27	0.43	0–4	0–5
Vocatives	1.42	0.18	0-10	0-4	0.90	0.59	0–7	0–15
Inclusive Pronouns	1.21	0.08	0-9	0-2	0.94	0.31	0–17	0–7
Phatics	0.13	0.61	0-2	0-3	0.05	0.43	0–1	0–2
Complimenting	0.02	0.25	0-1	0-3	0.03	0.03	0–2	0–1
Agreement	0.03	0.01	0-1	0-1	0.08	0.03	0–2	0–2
Participation	0.07	0.03	0-1	0-1	0.04	0.07	0–1	0–1
Appreciation	0.23	0.41	0-2	0-2	0.04	0.15	0–2	0–2
Negative Politeness	**1.48**	**1.42**	**0-30**	**0-15**	**2.18**	**2.02**	**0-40**	**0-20**
Disclaimers	0.06	0.15	0-2	0-2	0.16	0.18	0–3	0–3
Rational for FTA	0.07	0.10	0-4	0-2	0.07	0.10	0–2	0–1
Hedges/subjunctives	1.22	1.09	0-24	0-13	1.92	1.66	0–37	0–18
Formal Verbiage	0.13	0.08	0-2	0-1	0.03	0.08	0–1	0–4

5. Discussion

The results suggest that both similarities and differences exist in the use of politeness strategies between core and peripheral members in their online communication.

First, in general, both core and peripheral members used more positive politeness strategies than negatively politeness strategies. The results suggest that although team members were careful to respect others' autonomy, both roles seemed more likely to use positive politeness strategies to create cohesive social relations. Similar to other types of virtual teams, FLOSS development faces a variety of discontinuities that characterize virtual collaboration environment [51], so it is reasonable to expect that to reduce the distances created by these discontinuities, FLOSS team members, no matter their roles, are more likely to use positive politeness strategies.

However, although both core and peripheral members used more positive politeness strategies, they were used differently. By examining the individual indicators that showed significant differences between core and peripheral members in use of positive politeness strategies (the first column in Table 6), we found two different trends: core members used more vocatives and inclusive pronouns, but fewer phatics, complimenting and appreciation than peripheral members did. Analysis within the two projects also reveals a similar pattern, except for one indicator for Gaim (complimenting showed no difference between core and peripheral members). The results seem to generally suggest that it is not appropriate to simply group all the positive politeness strategies together and hypothesize that core members will use more of them than peripheral members do.

To explain this observation, we examined the definitions of the individual indicators in positive politeness category in more detail. We suggest that indicators convey two different intents regarding relationship building. The first three indicators—slurring/colloquialisms, vocatives and inclusive pronouns—are all employed to convey in-group membership and to show familiarity and intimacy among members. The other five indicators—phatics, complimenting, agreement, participation, and appreciation—instead seem to imply that the speakers want to give something that the other party desired, such as greetings, compliments, agreements, encouragements or appreciation.

Based on this observation, we suggest dividing the indicators in positive politeness category into two groups, which we label intimacy and respect separately. Intimacy is used to show familiarity and in-group membership between team members, while respect is used to give something that others desire. Further Mann-Whitney tests (Table 8) reveal consistent trends between core and peripheral members in using intimacy and respect indictors across and within groups. The results indicate that core members were more likely to use strategies that indicate intimacy such as inclusive pronouns and vocatives, while peripheral members were more likely to use strategies that show respect to others such as phatics and appreciation.

Table 8. Mann-Whitney U Tests Results

Indicator	**First Column**			**Second Column**			**Third Column**		
	C vs. P across Gaim and Fire			**C vs. P in Fire**			**C vs. P in Gaim**		
	Mean Rank		**p value**	**Mean Rank**		**p value**	**Mean Rank**		**p value**
	C	**P**		**F-C**	**F-P**		**G-C**	**G-P**	
Intimacy	423.39	280.59	0.000*	212.86	118.52	0.000*	207.90	160.25	0.000*
Respect	295.72	396.36	0.000*	140.34	200.22	0.000*	150.82	202.44	0.000*

One possible explanation for the observed differences is that core members usually function as leaders of the team. They know the project and other developers much better than the peripheral members do. It is their responsibility (rather than the peripheral members) to keep the project running. As a result, they are on a more intimate footing with other members.

Contrariwise, peripheral members post occasionally and most of the posts only make marginal contributions to software development (such as reporting a bug, making suggestions, asking questions and seeking help) compared to the core members who actually develop the software (such as writing code). As a result, it is normal for them to use positive politeness strategies such as phatics, complimenting and appreciation in their messages to show their respect and acknowledgement for the work of the core developers after they get the answer (e.g., *"thank you. You are great."*) or before they get any answer (e.g., *"I appreciate your time. You guys did a great job, but I think I found a bug."*).

5.1 Theoretical Implications

Our work has two theoretical implications. First, this research extends our understanding of social/emotional behaviors in community-based FLOSS development. Most prior research has focused on task-related behaviors such as collaboration [e.g., 52] and participation [e.g., 22]. Emotional behaviour plays an important role in everyday communication in FLOSS development, but few empirical studies have explored this topic [24]. Wei et al. [24] treated politeness strategies as an important component in group maintenance behavior and found that FLOSS team members do use a variety of politeness strategies in maintaining social relations with others. This research goes a step further and investigates how people with different roles enact different usage of politeness strategies. The results suggest that both core and peripheral members are actively engaged in maintaining social interactions using politeness strategies, although they use different strategies based on their roles. To our best knowledge, this is the first research that focuses on the relations between core-periphery structure and social-relational behaviors. Therefore, this research enriches the growing literature on FLOSS practice by investigating subtle matters in FLOSS project activities from contextual and behavioral perspectives.

Second, this work adds to our understanding of politeness theory, especially of positive politeness strategies, which are important components in politeness theory. Positive politeness strategies have been seen as less polite than negative politeness strategies and have the function of keeping people closer. Prior research usually treats all the positive politeness strategies as a whole and does not consider the different usage from different roles a person might assume. This research explores two important roles in FLOSS development and found that core and peripheral members do exhibit different patterns in using positive politeness strategies. Core members seemed to use more strategies that convey in-group membership (which we termed as intimacy), while peripheral members were more likely to use strategies that show respect and understanding to others (which we termed as respect). These two categories of positive politeness strategies have different degrees of politeness: intimacy seems to be less polite than respect. Therefore, this research underlies the importance of roles in studying individual's linguistic behavior using politeness theory.

5.2 Practical Implications

This research has implications for practitioners in community-based FLOSS development, which is built on relationships among members. Peripheral members might only perform a small amount of task-related activities, but they play an important role in relationship building during everyday communication in FLOSS development. In order to sustain a FLOSS community, it is important for project managers to be aware that members with different levels of knowledge of the project (as reflected in their roles of core/periphery) communicate differently. Specifically, the results suggest that core and peripheral members use different sets of positive politeness strategies when communicating with each other. Further, developers and project leaders should be aware of the nature of face threatening acts and consider how these are perceived by and the kind of impact they have on others. Differences between core and peripheral members in using politeness strategies described in this research might provide guidelines for community-based FLOSS participants for self-reflection about the effects of different kinds of intra-team communications. Much of this advice on how to approach email interaction is now commonly accepted based on research on email flame wars and how to avoid them, but our findings suggests how these results can be adapted to the context of virtual teams comprising individuals who work together regularly in a distributed mode via technology.

5.3 Limitations and Future Research

In this research, we examined the effect of member status—core vs. peripheral member—on individual use of politeness strategies in team communications. As with any study, there are a number of limitations that suggest opportunities for further research.

A first limitation is that our study focused on two factors and so did not include many other factors that might influence the use of politeness behaviours, such as project type, project culture, project governance structure, individual proficiency in language use and the communication channels used for interaction (e.g., IRC). Further, some of these factors, such as individual proficiency in language use, might affect both member status (e.g., the probability of becoming a core member) and the use of different politeness behaviors. Unfortunately, we do not have any data with which to test the impact of these factors; furthermore, our study design does not provide variance on project-related factors and communication medium. As a result, the specific effects of these factors remain topics for future research.

However, even if these factors do turn out to have direct impacts on the use of politeness behaviours, we have no reason to believe that they moderate the relationship between member status and the use of politeness in community-based FLOSS teams, which would change the results of our study. For example, although variance in individual-related factors (e.g., language proficiency) might affect the probability for an individual to become a core member, there should not be a difference in the impact of these factors on use of politeness behaviors which are different for core or peripheral members. That is, it should not explain the observed differences between core and periphery in use of politeness behaviour since we expect that such differences are found among both core and peripheral members.

A second limitation of the research is the possible impact of the small number of the core members in the data. In community-based FLOSS teams, the core usually includes a small number of people. Specifically, in the data we examined in this study, there were only 7 core members in Fire and 15 in Gaim, while the numbers of peripheral members were 129 and 137 respectively. Speakers might have specific personal traits or relational histories that predispose them to a set of particular strategies when speaking to others [53]. Because of the small numbers of the core members, variation in the behaviours of the core members might affect the observed use of politeness strategies. Although we averaged over all core members within each project to reduce the variance to some extent, subsequent research could examine more projects to create a large sample of core developers. Researchers could use other research methods such as interviews and surveys to further investigate the impact of personal traits on individual's use of politeness strategies.

Another limitation of this research is the difficulty of generalizing from a two-project study. This research only examined two projects from a category that develops same software: IM clients. Studying two projects from a same category controlled some unwanted variance (e.g., attractiveness of the project type to developers) and so provided a good setting in which to study the research question in detail. We argue that, although we investigated a special type of FLOSS team (i.e., community-based) with a particular medium (i.e., email lists), the phenomena we identified is a result of the nature of the relationships amongst core members and between core and peripheral members. Our hypothesis is that the different patterns of core and periphery in using politeness strategies are related to differences among members in knowledge of other members and of the project, as well as differences in authority over the project. We therefore expect that the relationship we observed will hold in many different types of FLOSS teams (e.g., those where many volunteers exist with different levels of commitment to the project). Nevertheless, we cannot rule out the possibility that the research findings could differ in other community-based FLOSS projects. Furthermore, though we do not expect our results to change, we have not tested our hypotheses in more hierarchical FLOSS projects, or FLOSS projects with different cultural types (those with more rational and hierarchical cultures). Future research should apply the framework of this study to a larger sample of projects that includes different kinds of projects. To do so, computerized analysis [e.g., 41, 54, 55] will be necessary to replace manual coding conducted in this research to reduce the cost of the necessary linguistic analysis to a manageable level [24].

A further unexplored factor is technology, which we held constant. It is possible that the specifics of different communication technologies may influence language use, which might affect the politeness strategies used. For example, politeness behaviours could be different on chats or IRC due to the immediacy brought by synchronous technologies. Tool support for FLOSS development also changes. For example, lots of discussions that might have happened in email lists or discussion fora are now happening in systems like Github. These changes in the technology support might affect communications and the use of politeness behaviours. At present, we do not have a theoretical perspective to offer specific hypotheses about such effects. Instead, the effect of communication medium on politeness behaviours is a further topic to investigate in future research. A complication is that our coding system was developed for discussion fora and might need to be revised for communications on a medium such as chat that favours terse messages.

Lastly, positive and negative politeness tactics may be used for a variety of reasons. In this paper, we have considered these as applied more-or-less automatically by developers as they respond to the communication situation, but it could be that developers use them more strategically. For example, in FLOSS projects core developers use different gatekeeping tactics to block unwanted contributions from the periphery. These tactics might be expressed via various positive and negative politeness tactics. Future research can build on our study as a basis to investigate how politeness is used for such political or operational reasons as well as to promote group well-being.

References

[1] E. von Hippel, *Democratizing Innovation*. 2005, Cambridge, Massachusetts: The MIT Press.

[2] A. Bonaccorsi and C. Rossi, *Why F/OSS can succeed.* Research Policy, 2003. **32**: p. 1243–1258.

[3] N. Economides and E. Katsamakas, *Two-Sided Competition of Proprietary vs. Open Source Technology Platforms and the Implications for the Software Industry.* Management Science, 2006. **52**(7): p. 1057-1071.

[4] W. Oh and S. Jeon, *Membership Herding and Network Stability in the Open Source Community: The Ising Perspective* management Science, 2007. **53**(7): p. 1086-1101.

[5] 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.

[6] F. Barcellini, F. Détienne, and J.-M. Burkhardt, *A situated approach of roles and participation in Open Source Software Communities.* Human–Computer Interaction, 2014. **29**(3): p. 205-255.

[7] C. Amrit and J. van Hillegersberg, *Exploring the impact of socio-technical core-periphery structures in open source software development.* journal of information technology, 2010. **25**(2): p. 216-229.

[8] C. Jensen and W. Scacchi. *Role migration and advancement processes in ossd projects: A comparative case study*. in *In proceedings of the 29th International Conference on Software Engineering (ICSE'07),*. 2007. IEEE.

[9] K. Crowston and J. Howison, *Assessing the health of open source communities.* IEEE Computer, 2006. **39**(5): p. 89-91.

[10] F. Rullani and S. Haefliger, *The Periphery on Stage: The Intra-Organizational Dynamics in Online Communities of Creation.* Research Policy, 2013. **42**(4): p. 941-953.

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

[12] G.A. Oliva, F.W. Santana, K.C.M.d. Oliveira, C.R.B.d. Souza, and M.A. Gerosa. *Characterizing Key Developers: A Case Study with Apache Ant*. in *CRIWG 2012, LNCS 7493, pp. 97–112*. 2012.

[13] L. Dahlander and S. O'Mahony, *Progressing to the center: Coordinating project work.* Organization Science, 2011. **22**(4): p. 961-979.

[14] 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.

[15] L. Dahlander and L. Frederiksen, *The Core and Cosmopolitans: A Relational View of Innovation in User Communities.* Organization Science, 2012. **23**(4): p. 988-1007.

[16] I. Steinmacher, M.A.G. Silva, M.A. Gerosa, and D.F. Redmiles, *A systematic literature review on the barriers faced by newcomers to open source software projects.* Information and Software Technology, 2015. **59**(67-85).

[17] 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.

[18] C.-C. Hsiao and J.-S. Chiou, *The impact of online community position on online game continuance intention: Do game knowledge and community size matter?* Information & Management, 2012. **49**(6): p. 292-300.

[19] Y. Fang and D. Neufeld, *Understanding Sustained Participation in Open Source Software Projects.* Journal of Management Information Systems, 2009. **25**(4): p. 9-50.

[20] J.R. Park, *Interpersonal and Affective Communication in Synchronous Online Discourse.* Library Quarterly, 2007. **77**(2): p. 133-155.

[21] J. Howison, K. Inoue, and K. Crowston. *Social dynamics of free and open source team communications*. in *Proceedings of the IFIP 2nd International Conference on Open Source Software*. 2006. Lake Como, Italy: Springer.

[22] G. Kuk, *Strategic Interaction and Knowledge Sharing in the KDE Developer Mailing List.* Management Science, 2006. **52**(7): p. 1031-1042.

[23] S. Toral, M. Martínez-Torres, and F. Barrero, *Analysis of virtual communities supporting OSS projects using social network analysis.* Information and Software Technology, 2010. **52**(3): p. 296-303.

[24] K. Wei, K. Crowston, N.L. Li, and R. Heckman, *Understanding group maintenance behavior in Free/Libre Open-Source Software projects: The case of Fire and Gaim.* Information & Management, 2014. **51**(3): p. 297-309.

[25] 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.

[26] P. Brown and S. Levinson, *Politeness: Some Universals in Language Usage*. 1987, Cambridge, United Kingdom: Cambridge University Press.

[27] A.J. Meier, *Passages of Politeness.* Journal of Pragmatics, 1995. **24**(4): p. 381-392.

[28] 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.

[29] 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.

[30] T. Holtgraves, *Social psychology, cognitive psychology and linguistic politeness.* Journal of Politeness Research. Language, Behaviour, Culture, 2005. **1**(1): p. 73–93.

[31] 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.

[32] P. Hobbs, *The medium is the message: politeness strategies in men's and women's voice mail messages.* Journal of Pragmatics, 2003. **35**: p. 243-262.

[33] J.-R. Park, *Linguistic Politeness and Face-Work in Computer-Mediated Communication, Part 1: ATheoretical Framework.* Journal of the American Society for Information Science and Technology, 2008. **59**(13): p. 2051-2059.

[34] B. Luthiger Stoll. *Fun and Software Development*. in *Proceedings of the First International Conference on Open Source Systems*. 2005. Genova, Italy.

[35] K. Crowston, K. Wei, Q. Li, and J. Howison. *Core and periphery in Free/Libre and Open Source software team communications*. in *Hawai'i International Conference on System System (HICSS-39)*. 2006.

[36] W. Scacchi, *Free/Open Source Software Development: Recent Research Results and Methods.* Advances in Computers, 2007. **69**: p. 243-295.

[37] G. Robles and J.M. Gonzalez-Barahona. *Contibutor Turnover in Libre Software Projects*. in *in IFIP International Federation for Information Processing, Volume 203, Open Source Systems*. 2006.

[38] M. Gharehyazie, D. Posnett, B. Vasilescu, and V. Filkov, *Developer initiation and social interactions in OSS: A case study of the Apache Software Foundation.* Empir Software Eng, 2015. **20**(1318-1353).

[39] P. Chejnová, *Expressing politeness in the institutional e-mail communications of university students in the Czech Republic.* Journal of Pragmatics, 2014. **60**: p. 175-192.

[40] R.K. Yin, *Case study research: Design and methods*. 3rd ed. Applied Social Research Methods Series, ed. L. Bickman and D.J. Rog. Vol. 5. 2003, Thousand Oaks, CA: Sage Publications.

[41] 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.

[42] G.A. Oliva, F.W. Santana, K.C. de Oliveira, C.R. de Souza, and M.A. Gerosa, *Characterizing key developers: a case study with apache ant*, in *In International Conference on Collaboration and Technology*. 2012, Springer Berlin Heidelberg. p. 97-112.

[43] M.B. Miles and A.M. Huberman, *Qualitative Data Analysis: An Expanded Sourcebook*. 2nd ed. 1994, Thousand Oaks: Sage Publications.

[44] R. Budd, R.K. Thorpe, and L. Donohue, *Content analysis of communication*. 1967, New York: Macmilliam.

[45] K. Krippendorff, *Content analysis: An introduction to its methodology*. 2012: Sage.

[46] V.J. Duriau, R.K. Reger, 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.

[47] 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.

[48] K.A. Neuendorf, *The Content Analysis Guidebook*. 2002, Thousand Oaks, CA: Sage Publications.

[49] D. Riffe, S. Lacy, and F.G. Fico, *Analyzing media messages: Using quantitative content analysis in research*. 2005: Psychology Press.

[50] 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.

[51] M.B. Watson ‐ Manheim, K.M. Chudoba, and K. Crowston, *Perceived discontinuities and constructed continuities in virtual work.* Information Systems Journal, 2012. **22**(1): p. 29-52.

[52] J. Howison and K. Crowston, *Collaboration through open superposition: A theory of the open source way.* Mis Quarterly, 2014. **38**(1): p. 29-50.

[53] M. Sifianou, *Disagreements, face and politeness.* Journal of Pragmatics, 2012. **44**(12): p. 1554-1564.

[54] F. Thung, D. Lo, and L. Jiang. *Automatic Defect Categorization*. in *2012 19th Working Conference on Reverse Engineering (WCRE)*. 2012. IEEE.

[55] 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.