



Contents lists available at ScienceDirect

Information & Management

journal homepage: www.elsevier.com/locate/im



Roles and politeness behavior in community-based free/libre open source software development

Kangning Wei^{a,*}, Kevin Crowston^b, U.Yeliz Eseryel^c, Robert Heckman^b

^a School of Management, Shandong University, Jinan, Shandong, 250100, China

^b School of Information Studies, Syracuse University, Syracuse, NY, 13244, USA

^c Faculty of Economics and Business, Department of Innovation Management & Strategy, University of Groningen, 9747, AE, Groningen, The Netherlands

ARTICLE INFO

Article history:

Received 7 January 2016

Received in revised form 31 October 2016

Accepted 27 November 2016

Available online xxx

Keywords:

Open source software development

Core-periphery structure

Politeness behavior

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.

© 2016 Elsevier B.V. All rights reserved.

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 noncoercive 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 usually 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, and active users and other registered users

including newcomers. Rullani and Haeffiger [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 core 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 they have been a primary focus of FLOSS research [12]. Peripheral members, on the other hand, contribute at a lower level, for example, 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 project success. 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]; therefore, maintaining a strong periphery is important to the long-term success of the projects. Amrit and van Hillegersberg [7] examined the core-periphery movement in open source projects and

* Corresponding author.

E-mail addresses: kwei@sdu.edu.cn, kwei.su@gmail.com (K. Wei), crowston@syr.edu (K. Crowston), u.y.eseryel@rug.nl (U.Y. Eseryel), rheckman@syr.edu (R. Heckman).

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]. Thus far, 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 occurs through information technologies such as asynchronous (e.g., e-mail 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, because they are the main communication channels that enable social-relational interaction between core and peripheral members for their development efforts, which is the focus of this paper.

Prior studies of relations in FLOSS have mostly examined the patterns of interactions among participants by using network-based analysis [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 e-mail 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 several 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. On the basis of 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 strategies²

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 tasks 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 considers 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 (FTAs) [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]. As 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, whereas positive face includes wanting respect, membership in a valued community, and a reputation for competence and fairness [29].

Face—the value that one claims for one's self—can, however, only be validated by others and so is dependent on others. It thus becomes everyone's interest to maintain the group by 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]. Moreover, 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 FTAs, and can either be directed toward the speaker or the hearer, and can threaten both types of

² This research builds on and extends our earlier work published in Information & Management (Wei et al. [24]). The previous and current paper share theoretical and empirical bases. As a result, there is duplication or near duplication of text from Wei et al. [24] for the presentation of politeness theory and politeness strategies (Section 2.1) and the data coding and data analysis process (Sections 3.3 and 3.4). This duplication is with permission from I&M editors to ensure the readability of this article..

Download English Version:

<https://daneshyari.com/en/article/4972603>

Download Persian Version:

<https://daneshyari.com/article/4972603>

[Daneshyari.com](https://daneshyari.com)