

The role of face-to-face meetings in technology-supported self-organizing distributed teams¹

Kevin Crowston
James Howison
Chengetai Masango
U. Yeliz Eseryel

Syracuse University
School of Information Studies

348 Hinds Hall
Syracuse, NY 13244

+1 315 443-1676

{crowston, jhowison, cmasango, uyeserye}@syr.edu

Draft of 12 February 2007

Revised research paper submitted to *IEEE Transactions on Professional Communications*

¹ This research was partially supported by NSF Grants 03-41475, 04-14468 and 05-27457. Earlier versions of this paper have been presented at the Academy of Management Conference, OCIS Division. We thank two anonymous FLOSS developers and IEEE TPC reviewers for their comments on earlier drafts of this paper.

The role of face-to-face meetings in technology-supported self-organizing distributed teams

Abstract

We examine the role of face-to-face meetings in the context of technology-supported self-organizing distributed or virtual teams, specifically Free/Libre Open Source Software (FLOSS) development teams. Based on a qualitative inductive analysis of data from interviews and observations at FLOSS conferences, we identify a variety of settings in which developers meet face-to-face, activities performed in these settings and benefits obtained. Contrary to the conventional wisdom, FLOSS developers generally do not meet face-to-face until the project is well under way. An additional benefit of face-to-face meetings is time away from a regular job and speed of interaction for certain kinds of tasks. (101 words)

Index terms: face-to-face meetings, technology-supported self-organizing distributed teams, virtual teams, Free/Libre Open Source Software (FLOSS) development

The role of face-to-face meetings in technology-supported self-organizing distributed teams

Introduction

In this paper, we explore the role of face-to-face meetings as a form of professional communication in the life of technology-supported self-organizing distributed (or virtual) teams. Virtual teams play an increasingly important role in many organizations, posing a practical problem for managers who must plan or oversee their work, as well as for team members who must learn to contribute effectively in this context. Practitioner research suggests the need for face-to-face meetings when a team is formed, but few academic studies have focused on the role of face-to-face meetings during a distributed team's life. Our study is intended as a first step towards addressing these gaps in the research literature by documenting the role that face-to-face meetings play in constituting intact and functional virtual teams. To address this gap, we draw on prior research on mixed-mode virtual teams [e.g., 1, 2] and on the literature on meetings as a constituting mechanisms of professional communications [e.g., 3].

Theory: Hybrid-mode virtual teams

Virtual teams are increasingly common in many organizations and researchers have begun to grapple with their concerns. [4] define a virtual team as, “a group of people who interact through interdependent tasks guided by common purpose” and work “across space, time, and organizational boundaries with links strengthened by webs of communication technologies”. However, in the research community, there is a growing

realization that virtual is not all or nothing [5-8]. Rather, teams fall along a continuum from traditional teams meeting only face-to-face to fully distributed teams, with many exhibiting a mixed mode of interaction.

In this paper, we examine cases in which a distributed team (or subsets of a team) meets in different modes at different times, specifically, primarily distributed with occasional face-to-face meetings. Such teams are interesting to study because the use of multiple modes has become more common as the geographic span of peoples' networks has increased [9], making exclusive or even frequent face-to-face meetings difficult. Furthermore, there is evidence that the mix of modes is beneficial. For example, Ocker et al. [1] found that mixed mode teams outperformed both face-to-face and distributed teams. Indeed, [2] went as far as to say that, "collaborative technologies in virtual environments enable better face-to-face meetings", However, much of this evidence is based on laboratory experiments, and authors such as [10] argued for the primacy of face-to-face interaction in real world interactions because of the "thickness" of this mode. They noted for example that face-to-face is used for the most important forms of interaction, such as conveying important or bad news. [11] argued that this preference stems from the fact that the human beings have evolved over millennia to be good at face-to-face communications. These contradictory findings point to the need for further research.

Prior research on meetings

[12] defined an encounter characterized by face-to-face multi-party interaction for some organizational purpose as a meeting (p. 61–62), and we will use this label for the

face-to-face encounters that we studied. The term is used to refer to a variety of encounters in the academic and professional literature: meetings may be scheduled or unscheduled, formal or informal. [12] discusses 2 roles for meetings aside from their ostensible reasons (e.g., making a decision): sensemaking, and social and cultural validation. [9] expands on this point, stating: “Beyond the literal making of decisions, various other functions result from meetings: seeing how one is heard, executing standard procedures and duties, distributing rewards, status and blame, reinforcing friendship as well as distance, judging the commitment of the organization, having an enjoyable time with colleagues and so on” (p. 165).

Face-to-face meetings for virtual teams

The practitioner literature on distributed teams tends to emphasize a need for face-to-face meetings at the start of a project for project planning [13], project definition [14], team development [15, 16], relationship building [17] and to increase trust among team members [18]. [19] suggests that initial face-to-face meetings are effective for building relationships (specifically trust) necessary to launch a project. These relationships also help team members develop shared understandings and thus perform better as a team. These suggestions are supported as well by the research literature. [20] found that developing a shared understanding in virtual teams through shared experiences can influence the ability of teams to co-ordinate work and perform well. [11] stressed the importance of mental schema alignment arising from shared understandings in reducing the effort required to communicate via CMC (p. 337). Similarly, [21] found that group cohesiveness improved performance. [22] suggested that starting a project with a face-to-

face meeting “gives people a chance to establish relationships and develop a sense of belonging to the team”. [9] suggested that what is exchanged in the meetings are “social goods”, allowing participants to put a face to a name (p. 162).

A few researchers have examined the role of face-to-face meetings during the life of distributed teams, which is our focus. [23] argued for the significance of “intermittent corporal co-presence within social life” (p. 257) even for groups that primarily use electronic interaction, noting that virtual and physical co-presence can not be viewed as simple substitutes. Studies have identified two specific functions for such face-to-face meetings.

First, researchers have noted that individuals choose different media for different kinds of tasks [e.g., 24, 25] in part because certain kinds of work are more suited to face-to-face meetings. However, there are two reasons offered for this fit. On the one hand, [23] argued that co-presence provides the possibility for richer, denser interaction needed for some tasks. [22] found that most teams rely to some extent on face-to-face meetings to ease the process of collaboration and coordination. [26] suggested more specifically that conducting regular meetings in person is essential to global virtual team effectiveness to the extent that the task requires a high degree of interdependence and there are geographic, organizational, and/or cultural boundaries that must be spanned. For example, one team they studied met face-to-face every four months to manage the future development of the contract by sharing plans and information, and generating ideas about co-development. On the other hand, [10] suggested that face-to-face interactions are beneficial because of the increased speed of interactions. Supporting this suggestion, [27]

found that the mean response time to contributions in an asynchronous medium was 138 hours vs. no more than 1 hour face-to-face and that asynchronous contributions took longer to prepare as well. [10] noted that quick interactions are particularly beneficial in cases of uncertainty about how to achieve an outcome, requiring working things out along the way (p. 270). Similarly, [28] noted that one benefit of face-to-face meetings is the collocation of people, such as customers, tutors or more experienced fellow workers who can answer questions as they come up (p. 120). They further stated that the awareness afforded in collocation also allows people to engage in informal training sessions (p. 119).

Second, echoing the function of kick-off meetings, researchers have noted that face-to-face communications are important to create and sustain social relationships that enable distributed work. [9, 23] argues that moments of physical contact are crucial for social life because physical co-presence is an important way to build group social capital since it provides an ability to gauge commitment [10] and build trust. Since ties degrade over time [29], periodic travel is necessary to establish and maintain ties [30]. From their ethnographic study of 22 employees collaborating across organizational boundaries, [29] argue that face-to-face meetings are needed to “establish and nurture the human relationships underlying business relationships”. Subjects in their studies talked about the importance of shared bodily activities in facilitating social bonding and showing commitment such as touching, eating and drinking together, engaging in mutually meaningful experiences in a common physical space and ‘showing up’ in person. [26] elaborate on this point, stating that effective virtual teams’ interactions are sequenced in a repeating temporal pattern. This basic pattern is defined by regular face-to-face meetings

in which the intensity of interaction is extremely high, followed by a period of some weeks in which interactions are less intense. They also mention that these meetings enable participants to maintain strong relationships through social meals and breaks. [31] go farther, arguing that because uncertainty, ambiguity and risk are difficult to manage through the lean channel of ICT, there may be “a minimum ratio of face-to-face to electronically mediated exchange that is vital to maintain in order for network organizations to work effectively” (p. 290).

Summary

In summary, existing research on face-to-face interactions in distributed teams suggests that these interactions will be used first for socialization to build or sustain team relationships (specifically trust, shared understandings and group cohesion) and second for work activities for which face-to-face interaction is better suited. However, there is not much academic literature on face-to-face meetings in mixed-mode virtual teams. In some ways, these meetings are treated as the ground against which the novel practice of virtual work is studied, echoing [12]’s observation that meetings are taken for granted as an organizational practice (p. 54). Yet, as she notes, “meetings produce organization”, so studying groups when they meet provides further information about how the concrete realization of their organization (the particular team or the larger organization in which it is situated, if any). These results suggest the need for further study to better understand the role of face-to-face meetings for distributed teams.

Data collection and analysis

Setting: Free/Libre Open Source Software development

To further explore the role of face-to-face meetings in the life of distributed teams, we use data from Free/Libre Open Source Software (FLOSS) development teams. FLOSS² is a broad term used to embrace software developed and released under an “open source” license allowing inspection, modification and redistribution of the software’s source code. There are thousands of FLOSS projects, spanning a wide range of applications. Due to their size, success and influence, the Linux operating system and the Apache Web Server are the most well known, but hundreds of others are in widespread use, including projects on Internet infrastructure (e.g., sendmail, bind), user applications (e.g., Mozilla, OpenOffice) and programming languages (e.g., Perl, Python, gcc). (The various projects mentioned in this paper are briefly described in Table I.) The research literature on software development and on distributed work emphasizes the difficulties of distributed software development, but the case of FLOSS development presents an intriguing counter-example, making it an interesting setting for our study.

Insert Table I about here

² FLOSS software is generally available without charge (gratis or “free as in beer” as [32] Free Software Foundation, “The Free Software Definition,” [Online document], 2006, [cited 20 January 2007], Available HTTP: <http://www.gnu.org/philosophy/free-sw.html> puts it). Some (though not all) FLOSS software is also “free software”, meaning that the user is guaranteed certain rights to the software and derivative works must be made available under the same license terms (“free as in speech”, thus “libre”). We have chosen to use the acronym FLOSS to accommodate this range of meanings.

FLOSS projects are an appropriate setting for our study of team communications because core developers working on a particular project form a work team [33]. Much of the literature on FLOSS has conceptualized developers as forming virtual communities [34-37], which is a useful perspective for understanding why developers choose to join or remain in a project. However, for the purpose of this study, we view the projects as entities that have a goal of developing a product, whose members are interdependent in terms of tasks and roles, and who have a user base to satisfy, in addition to having to attract and maintain members. These aspects of FLOSS projects suggest analyzing them as work teams. Guzzo and Dickson [33, pg. 308] defined a work team as “made up of individuals who see themselves and who are seen by others as a social entity, who are interdependent because of the tasks they perform as members of a group, who are embedded in one or more larger social system (e.g. community, or organization), and who perform tasks that affect others (such as customers or coworkers)”. FLOSS developers form a distributed team as they contribute from around the world and coordinate their activity primarily by means of computer-mediated communications (CMC) [38, 39]. FLOSS teams are an extreme example of [9]’s observation about growth in the geographic span of individuals’ networks, since the scope of these networks are often global.

FLOSS teams have several features that set them apart from many of the distributed teams that have been studied before, again making them interesting settings for our study. [40] notes that for virtual work to be feasible requires many assumptions about work that might not hold in general. However, these assumptions mostly do apply to FLOSS teams. First, because software products can be transferred electronically, the

work of software development can be executed entirely on-line, which is not the case for many other kinds of work [40, 41]. However on-line-only work might be typical of other “knowledge economy” industries. Second, as programmers, team members are accustomed to using CMC, suggesting that they have already developed ways of working adapted to this medium [11]. Third, FLOSS teams have a high isolation index [6] in that most team members work on their own rather than as part of a collocated subgroup. Fourth, FLOSS developers typically contribute to projects without being employed by a common organization (or for many, being paid at all). Finally, the teams are largely self-organizing, most often without formally appointed leaders or indications of rank or role. As a result, these teams depend on processes that span traditional boundaries of place and ownership [7].

These features differentiate FLOSS teams from many other virtual teams, but the issues and challenges that FLOSS teams face are not inconsistent with what many organizations now face in recruiting and motivating professionals and in developing distributed teams. As [42] put it, “increasingly employees are going to be volunteers, because a knowledge worker has mobility and can go pretty much every place, and knows it... Businesses will have to learn to treat knowledge workers as volunteers”. In short, FLOSS teams provide a vision of how work might be conducted in the future in a variety of settings [43].

FLOSS development projects include different classes of members with distinct roles and patterns of engagement. Research on organizationally-sponsored virtual teams has generally tacitly assumed that members are expected to contribute with more-or-less

equal intensity. Our study provides the opportunity to examine the implications of varying degrees of contribution more explicitly. Several authors have described FLOSS teams as having a hierarchical or onion-like structure [44, 45], as shown in Figure 1. At the centre are the core developers, who contribute most of the code and oversee the design and evolution of the project. The core is usually small and exhibits a high level of interaction, which would be difficult to maintain if the core team were large. Surrounding the core are the co-developers. These individuals contribute sporadically by reviewing or modifying code or by contributing bug fixes. The co-developer group can be much larger than the core, because the required level of interaction is much lower.

Insert Figure 1 about here

Surrounding the developers are the active users: a subset of users who use the latest releases and contribute bug reports or feature requests but not code. Still further from the core are the passive users. The border of the outermost circle is indistinct because the nature and variety of FLOSS distribution channels make it difficult or impossible to know the exact size of the user population. We do not consider either class of users to be team members because they do not necessarily share the team's goals and their tasks are not interdependent with the rest of the team, though clearly this is a matter of degree rather than a sharp boundary. As their involvement with a project changes, individuals may change their roles. A common pattern is for an active user to move into the core as their involvement increases. However, core developers must have a deep understanding of the software and the development processes, which poses a significant barrier to entry for individuals who want to join a team [46-48].

Data

The primary data for this paper comes from interviews and observation of face-to-face meetings of FLOSS developers at several conferences, specifically *ApacheCon* 2003 and 2004, *ApacheCon Europe* 2005 and 2006, *Comdex* 2003, *PloneCon* 2004, *OSCon* 2004 and *OSDC* 2004. These conferences are described more fully in Table II. We must clarify that our goal was not to compare teams that met face-to-face with those that do not. Rather, our goal was to understand the dynamics and the benefits of the face-to-face interactions in those virtual teams that do use this means of collaboration, hence our focus on these meetings for data collection. Members from many different FLOSS teams attended these conferences, so attendance enabled us to observe activities across multiple teams.

During the meetings, we observed the interactions of developers in order to establish what kinds of activities took place during face-to-face meetings. Our initial observations were open ended, as we tried to make sense of what was going on around us, but as our research questions crystallized, we focused our observation on the questions shown in Appendix A.

Insert Table II about here.

While observing the practices of developers at these conferences, we conducted formal and informal interviews with 27 developers, representing 22 FLOSS projects, including well-known projects such as Apache httpd, Perl and Mozilla, as well as lesser-known projects such as SpamAssassin and Plone. The projects represented were all by and large successful in that they had attracted and retained developers and released code

that is widely used [49]. We used convenience sampling at the conferences to select interview subjects. We approached potential participants during the conference sessions and breaks, introduced ourselves, described our research project and goals, and asked potential subjects if they would be willing to be interviewed. Although we do not have specific notes of refusals, we can say that nearly all individuals approached agreed to be interviewed. All interviewees reported that they were active members of FLOSS development teams.

Interviews were semi-structured, starting with a list of questions (shown in Appendix B), but then exploring in more depth the topics that were of interest to the respondent or for which she or he had particular insight. The initial round of interviews focused on issues of coordination and team building. Later rounds added specific questions about how face-to-face work was used in the team. Several of the interviews were recorded and transcribed; for the others, the interviewer took notes that were transcribed and analyzed.

Analysis

To analyze interview transcripts and observation notes, we applied a qualitative inductive analysis technique. The analysis was conducted using the Atlas-ti software package (<http://www.atlasti.com>). One author began by examining all interview transcripts and notes to identify text segments referring to face-to-face work in some way. These segments were then assigned to theoretically meaningful categories derived initially from the literature review summarized above (e.g., timing of meetings, outcome of meetings, types of work more appropriate for face-to-face settings). However, the

categories evolved through the course of the data analysis. As we coded each segment, we discussed whether the segment fit an existing code, required a new code or existing codes to be revised. We continued to revise the codes until each identified segment fit cleanly within some category. These codes were then grouped into higher-level categories and the relationships between these codes were elaborated. The process resulted in three sets of codes: opportunities for face-to-face encounters, activities during face-to-face encounters and results of these activities. The final code network developed is shown in Figure 2.

Insert Figure 2 about here

Findings

In this section, we discuss the main findings from our analysis: first the opportunities available for FLOSS developers to meet face-to-face, followed by the activities during face-to-face encounters and the perceived benefits of such encounters.

Opportunities for face-to-face meetings

Our first finding is that developers described face-to-face meetings as possible though rare. The developers we interviewed said that face-to-face meetings with other developers occur at most two or three times a year at conferences, if they are lucky. Furthermore, developers described not meeting fellow developers until well into a project, in sharp contrast to the generally accepted recommendation for a kick-off meeting for virtual teams. For example, one developer stated that he had worked on a project for two or three years before meeting any other member face-to-face.

Furthermore, it is not expected that every member of a team would be present for such meetings. At the meetings we observed for one project, only 4 of 6 key developers were present, for another 5 of 7, and one developer found few of his teammates present. In other words, meetings between developers at conferences often seemed opportunistic rather than specifically arranged or planned in advance.

Nevertheless, our interviewees described a variety of settings in which developers might meet face-to-face, including conferences, project meetings and “sprints”. We introduce each here briefly then expand on the nature of the interaction below.

Conferences are large events that usually bring together multiple FLOSS teams, though some are specific to a single project, such as in the case of Plone and PloneCon. Although each conference is in different in its schedule and program, conferences typically include panels, keynote speakers, project presentations, social events, etc., but are not specifically for the purpose of bringing together project members. Interviewees attended the conferences first for professional reasons, such as a presentation or professional development and only secondarily to be able to meet other developers. *Project meetings* are the formal or informal gatherings used by some FLOSS teams in order to coordinate project activities. The goals and the format of the project meetings also vary from team to team. *Sprints* are events organized to bring together a small group of people to accomplish a specific task in a given, usually short, period of time; hence the term sprint. Sprints are usually organized to develop code in a quick and agile manner, although there might be other uses of sprints, such as project initiation sprints. In the remainder of this section, we describe these opportunities for face-to-face interaction in more detail.

Conferences. Unsurprisingly given our approach to data collection, all respondents reported meeting other developers at conferences. In addition to the conferences we attended ourselves, respondents mentioned many other conferences including the *Ottawa Linux Symposium*, the *DebianConf*, *CodeCon*, *FOSDEM* (Europe), *EclipseCon*, the *MySQL AB Conference*, *PyCon*, the *GeekCruise* and “*Yet Another Perl Conference*”. Developers also reported chance meeting at specialized conferences such as an anti-Spam conference.

Many of the conferences described grew out of user group meetings. Computer user groups are associations of users interested in particular hardware or software packages that typically have regular meetings that allow users to meet and learn from each other. Many (indeed most) of the attendees at the conferences we attended were users of the software rather than developers. However, *ApacheCon* at least had a clear slant towards developers. Even so, we observed a distinct change in tone between the initial two days when only the developers were present and the final days when the rest of the attendees arrived. For example, the physical arrangement of attendees shifted from small group interactions to rows of audience members facing a stage (contrast Figure 3 and Figure 4).

Insert Figure 3 and Figure 4 about here

Project meetings. A second kind of meeting is a project meeting. In many cases, project meetings are held as planned adjuncts to conferences (i.e., rather than relying on chance interactions at the conferences). For example, the Perl developers were reported to be holding a meeting after *OSCon*, and Linux developers are reported to hold an

invitation only *Linux Kernel summit* prior to the *Ottawa Linux Symposium*. In a few cases, these are special stand-alone meetings. For example, one Apache developer described the original meeting to discuss the possibility of forming a non-profit foundation (the Apache Software Foundation, or ASF).

Sprints. A few of the face-to-face meetings reported were specifically organized to allow for development work. An extreme example of a task-based meeting is the “sprint”, short (2–3 day) programming events attended by perhaps a dozen developers who are invited to contribute to a particular set of tasks within a project. We were told that Sprint hosts and attendees are often funded by clients for whom the participants are working and using the project in that work. For a recent sprint, we were told that the host received \$10,000 in funding to cover costs and provided accommodation for the group in an Austrian castle.

Other meetings. Finally, in addition to these group meetings, developers may occasionally meet in small groups on an *ad hoc* basis. Several interviewees mentioned knowing other developers from work or school. One Apache developer mentioned that when traveling, he looks up the locations of other ASF members from a CVS based system known as “ICBM” (a play on the targeting system for nuclear missiles), and tries to arrange to meet. The group has a mailing list for announcing face-to-face parties. We interviewed the founder of Codehaus, an open source community and repository that grew out of an amicable difference of emphasis with the Apache community. This group

places a strong emphasis on the value of face-to-face interaction; indeed item 4 of their 10 point “manifesto”³ is:

4. The Codehaus encourages committers⁴ to be respectful friends, meet up with each other as often as possible. Face-to-face is superior to email.

Codehaus leaders encourage their core members to notify each other of their travel itineraries and make the effort to meet. Consistent with this value, Codehaus core members present at *ApacheCon* met for a “beer bash”.

Activities during and benefits of face-to-face meetings

Our second set of findings concerns the nature of the interactions during face-to-face meetings. We observed a variety of activities during face-to-face meetings, with associated benefits. Activities observed included socializing, socialization and team building, training, identify verification and work. In this section, we discuss these activities and their benefits together.

Socializing. A main benefit of face-to-face meetings is the opportunity to socialize with team members as opposed to planned work. Many interviewees reported spending time on socializing before (or instead of) getting down to work. The conferences we attended provided opportunities for socializing, some more than others. For example, *ApacheCon* included get-togethers on different afternoons, some with a company-

³ <http://codehaus.org/Manifesto>

⁴ Committers are individuals with the right to add code to a project’s source code repository, i.e., core members of the project.

sponsored open bar, during which developers could be observed talking together in small groups. Developers generally went to dinner together, with other members of their development teams or new acquaintances; during the dinners we observed, the conversation was primarily social as opposed to work-related. [9] similarly notes that work and play are frequently intermixed in meetings. An interviewee noted that it was worth taking leave time to come to a meeting because “it was a lot of fun” to get a bunch of developers together in the same room.

As expected, time spent socializing was seen as important for building and maintaining personal relationships. As [29] put it, you “demonstrate an enormous amount of unconscious commitment when you actually take the time and the trouble to put yourself in the same place as the person you want to build a relationship with”. For example, one developer attending *ApacheCon* for the first time commented that he definitely felt like he was getting connected to other developers and contributors and noted, “It’s not a technical connection”. Indeed, to link face-to-face with distributed team interactions, we observed some individuals hand-editing their conference name badges to include the user IDs, or ‘handles’, by which they are known on-line, including hand drawn cartoons and other personalizations. At one social event, we observed two attendees talking to each other for about ten minutes before one of them mentioned his handle to the other. The other immediately changed his neutral stance and gave his handle. Both exclaimed, shook hands, reintroduced themselves again and restarted their conversation in a more upbeat fashion with a totally different energy, having connected their online personas with their physical ones.

Many interviewees described the advantage of having met other developers on previous occasions. One developer noted that having met the other developers meant that he was now more comfortable sending them an email. Other echoed this sentiment:

When you haven't meet people, you build an image. After you've met them, you can have better email interactions.

Like anything in open source, so much easier to work with them via email after having met them. Much harder to get annoyed with them.

Much nicer to work together after getting to know each other more.

Once you have met each other face to face, the next time you see them online and they request something you are more likely to go out of your way to do it, rather than passing it over.

In summary then, face-to-face socializing seemed to be important for developers in developing social ties that facilitated on-line interaction, but similarly, a history of on-line interaction can be extended into the face-to-face realm; the two modes reinforce each other [8].

Socialization and team building. Face-to-face meeting may also provide important opportunities for socialization of members into teams, though our evidence on this subject is limited because most of our interviewees were already established members of teams. For example, we observed one team taking votes around the table as to how to pronounce their project name, an example of team norm setting, though they noted: “We

see each other once a year and it doesn't matter how we pronounce it." "As long as it is spelt the same".

We also saw some evidence that conferences provided opportunities for recognition of individual contributions, which is believed to be an important motivation for participation and thus an important part of bringing members into teams. One interviewee described his pleasure at going to a conference for the first time and being recognized and complemented by one of the core developers for his contributions on the mailing list. Several speakers we observed used their time in front of the audience to single out and recognize developers who had contributed to the particular project or feature being presented.

One group seems to have been particularly innovative in ensuring that their conferences support developer interactions. In this group, conference presenters, who are mostly core developers, have their way to the conference paid for by the conference organizers making it easier for them to attend. A few developers we talked to said that they decided to attend the conference when they received an invitation to present. This conference also included a formal "fellowship" program that supported attendance for selected developers. One such "fellow" told us that he viewed the support as important recognition of his current work as well as a "hook" for future work. In other words, the conference served as a way of converting user interest in the project as reflected in conference registrations into resources to support developer team building. [50] suggest that corporate participation can similarly benefit FLOSS projects by supporting developer travel.

Identity verification. An unexpected finding was that an important role of face-to-face meetings for some FLOSS teams was to provide an opportunity for developers to verify each other's identity. Recall that FLOSS teams are composed of individuals who typically do not work for the same organizations (unlike organizationally-sponsored teams). As a result, team members may feel the need to verify the identity of other participants, as noted by [50]. Many of the conferences conducted a formalized 'key-signing' for this purpose, during which participants matched digital identities, represented by PGP encryption keys, against real world identities represented by physical presence and government issued photo identification, typically passports and driver's licenses. An example of this verification process is shown in Figure 5. Formal key-signing sessions have a prescribed format, called for by security concerns, and approach a ritual in their formality. The online-offline boundary-spanning artifact formed by the key signature can be drawn on as evidence of an individual's bone fides even by those not present (provided they trust the process and the individual signing the key). [50] have studied the role of similar key-signings in the Debian community.

Insert Figure 5 about here

Training. We observed some informal examples of training (in addition to the formal tutorials that are characteristic of user group meetings), as suggested by [28]. For example, some of the conversation between developers appeared to be tips and hints for the use of new software tools for development and for general laptop administration. Developers from different projects, in similar genres (such as multiple Wiki implementations) were observed discussing how a particular feature had been

implemented in each project, thus providing an opportunity for cross-fertilization between projects that would not happen on the project developer mailing list.

Work. We were particularly interested in whether developers took advantage of their face-to-face meetings to work on development tasks, consistent with the prediction in the literature that some kinds of activities are best suited for face-to-face work. We observed numerous instances of developers working together face-to-face, in contrast to the accepted definition of FLOSS as developed by entirely distributed teams. For example, *ApacheCon* begins with an official “hack-a-thon“, during which developers take over a conference room for a weekend to work individually or in small teams (see Figure 3; note the face-to-face orientation of the groups, in contrast to the later sessions in which participants faced forward towards a presenter). The conference organizers provide wireless Internet access and power strips in the conference rooms for the laptops brought by almost all participants. In many cases, the hack-a-thon is informal, but at one of the conference we attended, it was a scheduled event sponsored by IBM. However, the hack-a-thon does not have a formal program; indeed, the only formal announcements we observed during the days of the hack-a-thon were to announce deliveries of pizza and donuts or that a group were about to leave for dinner.

Since participants came and left, it is not feasible to determine the total number of distinct participants attending a hack-a-thon. We did, however, count the number of participants in the room at regular intervals. At the *ApacheCon* hack-a-thon, typical of the pattern, there was a steady growth on the first morning to a peak of just over 50 participants that was sustained during our observation hours through until the hack-a-thon

merged into the Apache Foundation members meeting on the afternoon of the second day.

During the *ApacheCon* hack-a-thon, we saw many examples of software design work, such as joint work at a whiteboard (see Figure 6). One developer described the process of fixing a bug during a previous meeting and commented that there was lots of useful whiteboard work, which was not required, but which facilitated fixing the bug. He added that half the value of a whiteboard is being co-present and able to point, similar to the observations of [28]. On the other hand, some of the groups we observed working face-to-face continued to communicate via CMC such as internet relay chat (IRC) in order to include developers who had been unable to make the trip, again reflecting the mixing of face-to-face and distributed modes of work. This practice led to a number of curious seemingly ‘one-sided’ communications that we observed, such as an individual standing, seemingly unprompted, raising their hand while visually searching the room, announcing “Yes, that’s me, let’s talk about it”, evidently in response to a query on IRC. Because not all project team members are able to attend a hack-a-thon, there is a strong norm in this group that the face-to-face discussion be oriented towards raising possibilities and working out details, with the final decision deferred to a discussion on the mailing list to which all members can contribute.

Insert Figure 6 about here

We also observed many examples of individuals and small groups actually coding. Indeed, one project team stated that they had completed a new release of their system during the *ApacheCon* conference. As far as we know, the organized hack-a-thon

was unique to *ApacheCon*; we did not observe a similarly organized meeting at the other conferences we attended. However, developers at other conferences reportedly do take advantage of the time together to do some work. For example, FOSDEM in Europe is reported to make “developer rooms” formally available for projects to book for developer meetings. We heard one developer at *ApacheCon* state: “I am hoping to get locked in a room with the proxy guy to finally implement this thing”, while another interviewee reported “We sat down at PyCon and wrote it [new email parser] from scratch”.

In interviews, certain kinds of work were described as being more suitable for a face-to-face setting, and indeed, a couple of developers noted work had been explicitly put off until the face-to-face meeting (though this seemed to be the exception rather than the norm for other developers). One team that took particular advantage of the *ApacheCon* hack-a-thon was the Apache infrastructure project, the group that maintains the various servers used by Apache projects. An infrastructure team member commented “ApacheCon is great for infrastructure because we can discuss things and get things done”. The value of the face-to-face meeting was attributed to “higher bandwidth, lower latency”, which fit the time sensitivity of moving machines (for example, a team mailing list was down during the time it took to move it from one machine to another). Contrariwise, speaking about software developers, the team member commented that “it [non face-to-face work] slows them down, which is good” implying that slower work allowed for greater input from geographically distributed participants as well as more reflection on the problems and solutions. Indeed, other authors have argued that the distributed nature of FLOSS development may actually lead to more robust and maintainable code. The argument is that because distributed developers cannot consult

each other easily, they make fewer assumptions about how their code will be used and thus write more robust code that is highly modularized [51].

As noted above, some face-to-face meetings are specifically organized to allow for development work, making these projects more like traditional organizational-sponsored teams. We interviewed a number of participants from the Plone project who provided details about their sprints. Many Plone developers are consultants who install Plone for their clients. A sprint might be organized by a developer to develop new functionality required by a client. The new functionality is contributed back to the project, extending its capabilities.

Face-to-face was also seen as more appropriate for new ideas and strategic thinking. One developer noted, “We’re talking about [a jabber strategic issue] because we’re face to face this week”. Similarly, the ASF board and members meeting (including election of new members) are held face-to-face during *ApacheCon*, though with many members participating via CMC. On the other hand, contrary to the findings of [26], we did not observe much face-to-face time spent on project management, for example, long term planning, sketching of time-lines or delegation or assignment of tasks. The lack of such formal coordination mechanisms seems to be consistent with their overall low usage in FLOSS development [52].

Time to work. Finally, a few interviewees noted an advantage of coming to a face-to-face meeting that has been not mentioned in earlier work on distributed teams, namely the ability to spend focused time on the project. A distinctive feature of FLOSS teams is that many developers are volunteers who are not paid directly for their work or

employees for whom FLOSS work is a (frustratingly) small part of their total duties. Time away from demands of their “real jobs” to work on the project is therefore highly valued. For example, one interview commented that “it was great to ‘book some time’ for Apache business”. One interviewee told us that his FLOSS development occurs in ‘spare time’ —mornings before his “real job” and evenings after his family time—making it difficult for him to find the long blocks of continuous time offered by face-to-face meetings at conferences. This aspect of the conferences is consistent with the surprisingly large number of developers we observed working by themselves in a room full of others.

Discussion

In summary, our interviews and observations suggest that face-to-face meetings, while not frequent, do occur regularly, contrary to the general assumption that FLOSS teams are purely virtual, and serve important functions within FLOSS development teams. Face-to-face socialization helps to build and maintain social ties that developers report facilitate interactions. As well, we speculate that face-to-face interaction helps develop mental schema alignment, though this hypothesis remains to be tested. These ties are important for core developers, even though the bulk of their interaction happens via CMC. [50] similarly note that Debian developers seem to trust people they have met more, as reflected in the elections as project leaders of individuals central in the social network, as revealed by the links between signed PGP keys that result from the key signings mentioned above.

Interestingly though, we observed relatively few scheduled meetings of developers. Rather, the scheduled conference events produced a “ripple” of unscheduled

meeting that seemed as or more important than the formal event itself [3]. [23] similarly noted that travel facilitates unplanned meetings and so people travel to the site of likely co-encounters, even though most of encounters may be unplanned. The importance of these unplanned encounters can be seen in the increasing number of networking events: conferences with no preset program entirely oriented towards unplanned interactions (e.g., FOO camp and its derivatives such as BAR camp and OS camp). [53] suggests that these kinds of interactions represent a “network sociality” that consists of “fleeting and transient, yet iterative social relations”. In his view, when the primary form of interaction is via CMC, relationships are more ephemeral: intense, but short-lived.

However, despite the apparent importance of face-to-face meetings for building social relations, the developers we interviewed belonged to well-established projects that had already experienced substantial success, rather than representing newly-formed teams meeting to launch a project as suggested in literature on non-voluntary teams. It seems that the expense and effort required for face-to-face meetings only makes sense for developers once a project is well underway and their commitment to the project has developed. As a result, the teams seem to function without the planned face-to-face kick-off meeting called for by the conventional wisdom for managing virtual teams. A question for future research then is how such group commitment is developed via CMC.

Second, certain kinds of team activities are easier to accomplish with face-to-face interactions, enough so that in a few exceptional cases they may actually be deferred until a planned meeting. In contrast with prior literature though, the tasks that were reserved for face-to-face activities seemed to be ones that required quick interaction, rather than

those having high levels of interdependence that span boundaries [26]. These findings are compatible with [28]'s finding that a benefits of face-to-face meetings as the collocation of people who can answer questions as they come up. The preference in FLOSS teams seems to be to organize work so as to reduce interdependence, e.g., by writing more modular code and using technical systems to manage possible conflicts. Because the work can be accomplished entirely electronically, for the most part it is. We believe that the emphasis on socialization similarly reflects the fact that socialization is harder to virtualize than programming.

Finally, spending time at a meeting is important for many attendees since it allows them to focus their attention on a project. This final observation suggests that the relevant theories for studying FLOSS teams may include literature on volunteers in voluntary associations or organizations [54, 55]. Of course, there are many kinds of voluntary associations and several typologies have been proposed [56]. [57] differentiated groups on three dimensions: status, accessibility and purpose (expressive, instrumental or instrumental-expressive). [58] divided voluntary associations into service-oriented, issue-oriented, self-expressive, economic self-interest and philanthropic volunteerism, though he notes that each may contain aspects of the others. [59] offered a similar typology of service providers, research and advocacy, common enthusiasm, self-help and intermediary bodies. In these typologies, FLOSS projects seem to be examples of self-expressive or common enthusiasm groups, shading into service-oriented for the larger projects. [58] noted that members of self-expressive associations usually do not consider themselves "volunteers" but rather members of an organization (p. 113), as seems to be the case with most of the developers we interviewed.

However, [58] noted that the literature on voluntary associations is dispersed across many fields (p. 117) and without much focused attention, resulting in a fragmented state of knowledge. As well, most voluntary associations literature seems oriented towards the problems of professional managers in these organizations who need to recruit, motivate and manage volunteers, or like much of early literature on FLOSS, focused on the motivations of volunteers [60, 61]. There seems to be little research on communication practices within voluntary associations, which is problematic because as [62] suggested, most voluntary associations are comparatively smaller, simpler and operate in a different environmental context than for-profit companies, so many theories developed in for-profit settings seem hard to apply. Some exceptions include [3]'s study of meetings in a voluntary organization, [63]'s examination of management of volunteers and the relation between paid and volunteer members in church congregations and [64]'s study of the governance structure and decision making processes of NOW (the National Organization for Women).

Nevertheless, findings from this literature show many points of similarities with FLOSS research, supporting its possible applicability. For example, [65] found that a small group of individuals are responsible for the majority of contributions to voluntary associations, as has been frequently observed in FLOSS projects [52] (and counter the tacit assumption that all members should be roughly equal contributors). Also, the observation that voluntary organizations tend to become more formal over time [66] seems consistent the experiences of larger FLOSS projects, e.g., as evidenced by the creation of formal organizations such as the Apache Software Foundation. A particularly relevant perspective may be [67]'s study of grassroots associations, i.e., local

organizations with no paid staff. [67] notes that participants in grassroots associations gain satisfaction primarily from participation, based on solidarity and purposive incentives, which matches the observed motivations of many FLOSS developers [68, 69]. Finally, voluntary associations are similar to FLOSS teams in that it is difficult for outsiders to assess their effectiveness; certainly simple profit calculations cannot be used [62].

Research in the area of voluntary associations might therefore provide useful guidance for FLOSS research, e.g., [70]'s study of the size of voluntary groups and [71]'s study of the dynamics of their composition could guide research on the composition of FLOSS teams. Contrariwise, research on FLOSS teams might help illuminate topics of interest to professional communications researchers studying voluntary associations, such as the shifting relationship between work and employment [72].

Conclusion

Researchers studying FLOSS development often contend that FLOSS teams do not meet face-to-face at all as a justification for basing their research only on online artifacts. However, as we show above, many developers in fact do meet face-to-face, and these meetings may play an important but overlooked role in FLOSS team practices. Many members of at least the larger and more established FLOSS teams both have regular opportunities to meet and use these opportunities. For example, in their study of Debian developers, [50] document a pattern of meetings, driven in part by a need to verify potential developers' identities and commitment to the project. The norm in some groups that final decisions be made on-line to allow full participation suggests that the on-line record for these groups should be a reliable record of the groups' decisions.

However, these records may be incomplete for phenomena such as group maintenance or socialization that have a significant face-to-face component.

We conclude with a few observations about the generality of our findings and directions for possible follow on research. A first limitation of our study is that most of our interviewees were established core members of their teams. We have less data about the role of face-to-face meetings for other kinds of FLOSS community members. However, while active and passive users are the majority of those attending the user conferences we observed, we did not observe them to interact as extensively as the core developers. These users seem to attend sessions as a way of learning more about the project, rather than in an attempt to become more involved or even to influence the direction of the project. Some attendees were sent by their companies just to get a feel of what was going on with the project (e.g., an attendee from Microsoft who was there “just to observe”). These observations are consistent with our definition of the project team as excluding users, even active users. More systematic data is needed though to fully understand these interactions.

Second, as we noted, our data is drawn primarily from large projects. We believe that many of these projects have regularly scheduled opportunities for face-to-face meetings of developers, though we have not been able to observe some of them because they are not open to the public, e.g., the Perl developers meeting or the Linux Kernel summit. However, we do not know how typical such meetings are. As a result, we cannot claim that the patterns we observed are typical for FLOSS. Indeed, FLOSS projects follow a power law distribution in size, meaning that there are a few very big projects and

a lot of small ones. Such a distribution makes the notion of a “typical” project meaningless.

We speculate instead that while large projects have meetings, small projects have only a few members who may already know each other from prior interactions (making them perhaps more like typical organizationally-sponsored distributed teams). The most interesting case for further study then should be medium-sized projects, ones that have reached a size where face-to-face interactions would be beneficial or even necessary but where the resources to support such meetings have not yet developed. In terms of the taxonomy of voluntary organizations proposed by Handy [59], these projects are also transitioning from mutual interest organizations focused on the needs of members to service organizations addressing the needs of a larger population of users, with the inherent tensions this transition implies. There may be a natural evolution as projects grow until they reach a point where developers need to meet to be able to manage the further development of the project. We have heard anecdotally of projects of this size where members try to arrange meetings at various conferences that they attend, but we have not yet been able to study them systematically. It is unclear whether this is a step towards project success or a sign of serious continuing coordination difficulties within a project that are not caused by growth alone. If the later were the case, a one-time face-to-face meeting is unlikely to resolve such issues.

Similarly, developers on projects may go through a similar evolution in their participation. An active user may be quite effective without having ever met any other developers. However, it may be that developers typically meet at least some other

developers face-to-face in the process of becoming accepted as core members of the project. Indeed [50] report that this practice is the policy for at least one large project, Debian. At present, we do not have the data to answer this question because our interviews were all conducted with individuals who were present at conferences. A more systematic survey of core developers and co-developers is needed.

In conclusion, it seems clear that research on the development practices of FLOSS teams needs to take into account face-to-face interactions, at least for large projects and possibly for others as well. Understanding these interactions is important for understanding both the social development of the team and the development of the system. In particular, the phenomenon of “sprints” seems to be a rich area for further study. Second, we need more systematic data about who attends face-to-face meetings and who does not. Such data would help understand the evolution of individual roles within teams and the role of face-to-face meetings in the life of distributed teams.

References

- [1] R. J. Ocker, J. Fjermestad, S. R. Hiltz, and K. Johnson, "Effects of four modes of group communication on the outcomes of software requirements determination," *Journal of Management Information Systems*, vol. 15, pp. 99–118, 1998.
- [2] S. Qureshi and I. Zigurs, "Paradoxes and perogatives in global virtual collaboration," *Communications of the ACM*, vol. 44, pp. 85–88, 2001.
- [3] H. Schwartzman, *The meeting: Gatherings in organizations and communities*. New York, NY: Plenum, 1989.
- [4] J. Lipnack and J. Stamps, *Virtual teams: Reaching across space, time and organizations with technology*. New York, NY: John Wiley and Sons, Inc, 1997.
- [5] F. Niederman and C. Beise, "Defining the 'virtualness' of groups, teams, and meetings," in *Proceedings of the 1999 ACM SIGCPR Conference*. New Orleans, LA, 1999, pp. 14–18.
- [6] M. O’Leary and J. Cummings, "The Spatial, Temporal, and Configurational Characteristics of Geographic Dispersion in Teams," presented at Academy of Management Conference, Denver, CO, 2002.
- [7] M. B. Watson-Manheim, K. M. Chudoba, and K. Crowston, "Discontinuities and continuities: A new way to understand virtual work," *Information, Technology and People*, vol. 15, pp. 191–209, 2002.
- [8] M. Gaved and P. Mulholland, "Grassroots Initiated Networked Communities: A Study of Hybrid Physical/Virtual Communities," in *Proceedings of the 38th Hawaii International Conference on System Sciences (HICSS 2005)*. Big Island, HI, 2005.
- [9] J. Urry, "Social networks, travel and talk," *British Journal Of Sociology*, vol. 54, pp. 155-175, 2003.
- [10] D. Boden and H. L. Molotch, "The compulsion of proximity," in *NowHere: Space, Time and Modernity*, R. Friedland and D. Boden, Eds. Berkeley: University of California, 1994, pp. 257–286.
- [11] N. Kock, "The psychobiological model: Towards a new theory of computer-mediated communication based on Darwinian evolution," *Organization Science*, vol. 15, pp. 327–348, 2004.
- [12] H. Schwartzman, "The meeting as a neglected social form in organizational studies," in *Research in organizational behavior*, vol. 9, B. Staw and L. Cummings, Eds. Greenwich, CT: JAI, 1986, pp. 233–258.

- [13] A. Powell, G. Piccoli, and B. Ives, "Virtual teams: A review of current literature and directions for future research," *Database for Advances in Information Systems*, vol. 35, pp. 6-36, 2004.
- [14] V. Ramesh and A. Dennis, "The object-oriented team: Lessons for virtual teams from global software development," in *Proceedings of the Thirty-Fifth Annual Hawaii International Conference on System Sciences*. Hawaii, 2002.
- [15] C. S. Saunders, "Virtual teams: Piecing together the puzzle," in *Framing the Domain of IT Management: Projecting the Future Through the Past*, R. W. Zmud, Ed. Cincinnati, OH: Pinnaflex, 2000.
- [16] B. C. Y. Tan, K. K. Wei, W. W. Huang, and G. N. Ng, "A dialogue technique to enhance electronic communication in virtual teams," *Professional Communication, IEEE Transactions on*, vol. 43, pp. 153-165, 2000.
- [17] D. Robey, H. M. Khoo, and C. Powers, "Situated-learning in cross-functional virtual teams," *IEEE Transactions on Professional Communication*, pp. 51-66, 2000.
- [18] J. Suchan and G. Hayzak, "The communication characteristics of virtual teams: A case study," *IEEE Transactions on Professional Communications*, vol. 44, pp. 174-186, 2001.
- [19] L. Anschuetz, "Managing Geographically Distributed Teams," presented at IEEE Professional Communication Society Conference, Québec City, Canada, 1998.
- [20] P. Hinds and S. Weisband, "Knowledge sharing and shared understanding in virtual teams," in *Virtual teams that work: Creating conditions for virtual teams effectiveness*, C. B. Gibson and S. G. Cohen, Eds. San Francisco: Jossey-Bass, 2003, pp. 21-36.
- [21] R. Huang, T. A. Carte, and L. Chidambaram, "Cohesion and Performance in Virtual Teams: An Empirical Investigation," in *Proceedings of the Tenth Americas Conference on Information Systems*. New York, NY, 2004.
- [22] L. Dubé and G. Paré, "The multifaceted nature of virtual teams," in *Virtual Teams: Projects, Protocols and Processes*, D. J. Pauleen, Ed. Hershey, PA: Idea Group, 2004, pp. 1-39.
- [23] J. Urry, "Mobility and proximity," *Sociology-the Journal of the British Sociological Association*, vol. 36, pp. 255-274, 2002.
- [24] J. L. McKenney, M. H. Zack, and V. S. Doherty, "Complementary communication media: A comparison of electronic mail and face-to-face communication in a programming team," in *Networks and Organizations:*

- Structure, Form, and Action*, N. Nohria and R. G. Eccles, Eds. Boston, MA: Harvard Business School, 1992, pp. 262–287.
- [25] R. L. Daft, R. H. Lengel, and L. K. Trevino, "Message equivocality, media selection and manager performance: Implications for information systems," *MIS Quarterly*, pp. 355–366, 1987.
- [26] M. L. Maznevski and K. M. Chudoba, "Bridging space over time: Global virtual team dynamics and effectiveness," *Organization Science*, vol. 11, pp. 473–492, 2000.
- [27] N. Kock, "Can communication medium limitations foster better group outcomes? An action research study," *Information & Management*, vol. 34, pp. 295–305, 1998.
- [28] J. S. Olson, S. Teasley, L. Covi, and G. Olson, "The (currently) unique advantages of collocated work," in *Distributed Work*, P. Hinds and S. Kiesler, Eds. Cambridge, MA: MIT Press, 2002, pp. 113–135.
- [29] B. A. Nardi and S. Whittaker, "The place of face to face communication in distributed work," in *Distributed Work: New Research on Working across Distance Using Technology*, P. Hinds and S. Kiesler, Eds. Cambridge, MA: MIT Press, 2002, pp. 83–110.
- [30] H. Schwarz, B. A. Nardi, and S. Whittaker, "The hidden work in virtual work," presented at International Conference on Critical Management Studies, Manchester, UK, 1999.
- [31] N. Nohria and R. G. Eccles, "Face-to-face: Making network organizations work," in *Networks and Organizations: Structure, Form, and Action*, N. Nohria and R. G. Eccles, Eds. Boston, MA: Harvard Business School, 1992, pp. 288–308.
- [32] Free Software Foundation, "The Free Software Definition," [Online document], 2006, [cited 20 January 2007], Available HTTP: <http://www.gnu.org/philosophy/free-sw.html>
- [33] R. A. Guzzo and M. W. Dickson, "Teams in organizations: Recent research on performance effectiveness," *Annual Review of Psychology*, vol. 47, pp. 307–338, 1996.
- [34] B. Wellman, J. Salaff, D. Dimitrova, L. Garton, M. Gulia, and C. Haythornthwaite, "Computer networks as social networks: Collaborative work, telework, and virtual community," *Annual Review of Sociology*, vol. 22, pp. 213–238, 1996.

- [35] M. Bergquist and J. Ljungberg, "The power of gifts: organizing social relationships in open source communities," *Information Systems Journal*, vol. 11, pp. 305–320, 2001.
- [36] S. Sharma, V. Sugumaran, and B. Rajagopalan, "A framework for creating hybrid-open source software communities," *Information Systems Journal*, vol. 12, pp. 7–26, 2002.
- [37] B. Fitzgerald and J. Feller, "A further investigation of open source software: community, co-ordination, code quality and security issues," *Information Systems Journal*, vol. 12, pp. 3–6, 2002.
- [38] E. S. Raymond, "The cathedral and the bazaar," *First Monday*, vol. 3, 1998.
- [39] P. Wayner, *Free For All*. New York: HarperCollins, 2000.
- [40] G. Symon, "Information and communication technologies and the network organization: A critical analysis," *Journal of Occupational and Organizational Psychology*, vol. 73, pp. 389–414, 2000.
- [41] S. Furst, R. Blackburn, and B. Rosen, "Virtual team effectiveness: A proposed research agenda," *Information Systems Journal*, vol. 9, pp. 249–269, 1999.
- [42] J. Collins and P. Drucker, "A Conversation between Jim Collins and Peter Drucker," in *Drucker Foundation News*, vol. 7, 1999, pp. 4–5.
- [43] T. W. Malone, *The Future of Work: How the New Order of Business Will Shape Your Organization, Your Management Style and Your Life*. Boston: Harvard Business School, 2004.
- [44] A. Cox, "Cathedrals, Bazaars and the Town Council," [Online document], 1998, [cited 22 March 2004], Available HTTP: <http://slashdot.org/features/98/10/13/1423253.shtml>
- [45] J. Y. Moon and L. Sproull, "Essence of distributed work: The case of Linux kernel," *First Monday*, vol. 5, 2000.
- [46] R. T. Fielding, "The Apache Group: A case study of Internet collaboration and virtual communities," [Online document], 1997, [cited 20 January 2007], Available HTTP: <http://roy.gbiv.com/talks/ssapache/title.htm>
- [47] C. Gacek and B. Arief, "The many meanings of Open Source," *IEEE Software*, vol. 21, pp. 34–40, 2004.
- [48] F. Hecker, "Mozilla at one: A look back and ahead," [Online document], 1999, [cited 20 January 2007], Available HTTP: <http://www.mozilla.org/mozilla-at-one.html>

- [49] K. Crowston, J. Howison, and H. Annabi, "Information systems success in Free and Open Source Software development: Theory and measures," *Software Process—Improvement and Practice*, vol. 11, pp. 123–148, 2006.
- [50] S. O'Mahony and F. Ferraro, "Managing the Boundary of an 'Open' Project," in *Market Emergence and Transformation*, W. Powell and J. Padgett, Eds. Cambridge, MA: MIT Press, In Press.
- [51] G. K. Lee and R. E. Cole, "From a firm-based to a community-based model of knowledge creation: The case of Linux kernel development," *Organization Science*, vol. 14, pp. 633–649, 2003.
- [52] 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*, vol. 11, pp. 309–346, 2002.
- [53] A. Wittel, "Toward a Network Sociality," *Theory, Culture & Society*, vol. 18, pp. 51–76, 2001.
- [54] S. K. Shah, "Motivation, governance, and the viability of hybrid forms in open source software development," *Management Science*, vol. 52, pp. 1000–1014, 2006.
- [55] B. S. Butler, "When Is a Group not a Group: An Empirical Examination of Metaphors for Online Social Structure," presented at OCIS division, Academy of Management, New Orleans, LA, 2004.
- [56] C. E. Smith and A. Freedman, *Voluntary Associations: Perspectives on the Literature*. Cambridge, MA: Harvard, 1972.
- [57] C. W. Gordon and N. Babchuk, "A typology of voluntary associations," *American Sociological Review*, vol. 24, pp. 22–29, 1959.
- [58] D. H. Smith, "Research and communication needs in voluntary action," in *Volunteerism: An Emerging Profession*, J. G. Cull and R. E. Hardy, Eds. Springfield, IL: Charles C. Thomas, 1974, pp. 111–186.
- [59] C. Handy, *Understanding Voluntary Organizations: How to Make Them Function Effectively*. New York: Penguin, 1988.
- [60] J. Wilson, "Volunteering," *Annual Review of Sociology*, vol. 26, pp. 215–240, 2000.
- [61] R. C. Dailey, "Understanding organizational commitment for volunteers: Empirical and managerial implications," *Journal of Voluntary Action Research*, vol. 15, pp. 19–31, 1986.

- [62] D. Knoke and D. Prensky, "What relevance do organization theories have for voluntary associations?," *Social Science Quarterly*, vol. 65, pp. 3–20, 1984.
- [63] M. Harris, "Doing it their way: Organizational challenges for voluntary associations," *Nonprofit and Voluntary Sector Quarterly*, vol. 27, pp. 144–158, 1998.
- [64] M. Barakso, "Civic engagement and voluntary associations: Reconsidering the role of the governance structures of advocacy groups," *Polity*, vol. 37, 2005.
- [65] P. B. Reed and L. K. Selbee, "The civic core in Canada: Disproportionality in charitable giving, volunteering, and civic participation," *Nonprofit and voluntary sector quarterly*, vol. 30, pp. 761–780, 2001.
- [66] F. S. Chapin and J. E. Tsouderos, "The formalization process in voluntary associations," *Social Forces*, vol. 34, pp. 342–344, 1956.
- [67] D. H. Smith, "Grassroots associations are important: Some theory and a review of the impact literature," *Nonprofit and voluntary sector quarterly*, vol. 26, pp. 269–306, 1997.
- [68] K. R. Lakhani and B. Wolf, "Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects," [Online document], 2003, [cited 1 March 2005], Available HTTP: <http://opensource.mit.edu/papers/lakhaniwolf.pdf>
- [69] A. Hars and S. Ou, "Why Is Open Source Software Viable? - A Study of Intrinsic Motivation, Personal Needs, and Future Returns," presented at The 2000 Americas Conference on Information Systems (AMCIS 2000), 2000.
- [70] J. M. McPherson, "The size of voluntary organizations," *Social Forces*, vol. 61, pp. 1044–1064, 1983.
- [71] J. M. McPherson and T. Rotolo, "Testing a dynamic model of social composition: Diversity and change in voluntary groups," *American Sociological Review*, vol. 61, pp. 179–202, 1996.
- [72] R. F. Taylor, "Extending conceptual boundaries: Work, voluntary work and employment," *Work, Employment and Society*, vol. 18, pp. 29–49, 2004.

Figures

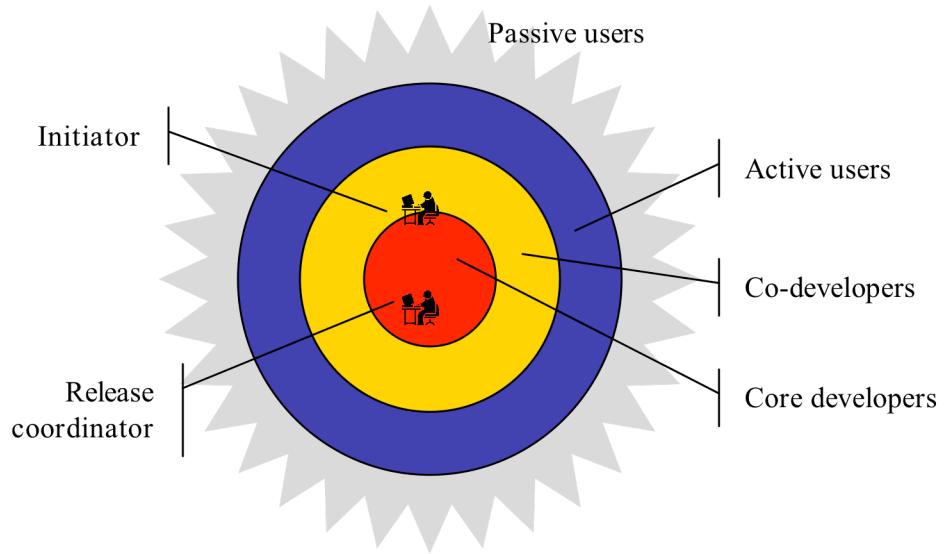


Figure 1. Hypothesized FLOSS team structure.

Table I. FLOSS projects mentioned in the paper.

Project	Description
Apache	A Web server (i.e., a program that runs on a server to provide HTML documents in response to requests). The Apache project now includes numerous related projects, with the Web server referred to as httpd.
bind	A domain name server (i.e., a program that runs on a server to respond to requests to translate computer names to IP addresses).
CVS	Concurrent versioning system, a source code control system (i.e., a system that facilitates integration of changes made to source files by various developers into a common repository). CVS can be used to manage any text file, not just source code.
Debian	A Linux distribution (i.e., the kernel plus a selection of utilities and applications tested and packaged for consistency and easy installation).
Eclipse	An integrated program development environment (IDE). Originally developed by IBM for Java programming and later released as open source. Eclipse support plugins that customize behaviour to support different programming tasks.
gcc	A C programming language compiler (i.e., a program that translates program source code into a form that can be executed).
IRC	Internet Relay Chat, a synchronous text based communications system.
Linux	The kernel of a Unix-like operating system. Some refer to the full operating system as GNU/Linux as many of the supporting utilities are products of the GNU (“GNU is not Unix”) project.
Mozilla	A Web browser derived from Netscape. The Mozilla project includes several spinoffs, such as the Firefox browser.
MySQL	A relational database program. Developed by MySQL AB.
OpenOffice	A suite of office applications intended to replace Microsoft Office. Originally developed by Sun.
Perl	An interpreted programming language with particular strengths in text manipulation and system scripting.
Plone	A content management system built on Zope.
Python	Another interpreted programming language.
sendmail	A mail transfer agent (i.e., a program that runs on a server to receive and send email, but which does not provide an end-user interface).
SpamAssassin	An email spam filter. Now an ASF sponsored project.
Zope	An object-oriented Web application platform.

Table II. Conferences attended for data collection.

Conference	Description
<p><i>ApacheCon.</i> The Annual meeting of the Apache Software Foundation (ASF). The ASF oversees the development of the Apache Web server and about two dozen other projects, most but not all associated with the Web server. ApacheCon is held annually in North America, Europe and most recently in Asia.</p>	<p>The conference runs Sunday or Monday to Wednesday or Thursday, with the first day or two devoted to tutorials and the remainder to short presentations. The conference also includes nightly “birds of a feather” sessions and a keynote address each day, given by a leading developer or other personality. Core developers from various ASF projects present most of the talks, usually overviews of a project or introductions of new features. Attendees are primarily users of ASF software, generally technical employees from various companies.</p>
<p><i>Comdex.</i> COMDEX’s Linux and Open Source Conference and educational program. Included an Open Source pavilion.</p>	<p>Developers from a number of projects were sponsored by O’Reilly publishing to attend, make presentations and answer questions about their products. The projects presenting were selected by online poll.</p>
<p><i>PloneCon.</i> Annual meeting of the developers of the Plone Content Management System.</p>	<p>Specific to the Plone project.</p>
<p><i>OSCon.</i> The O’Reilly Open Source Conference.</p>	<p>Grew out of the Perl Conference and still attended by many Perl users, but now includes tracks on other languages and systems. The conference has the same general format as the <i>ApacheCon</i>.</p>
<p><i>OSDC.</i> Open Source Developers’ Conference, an Australian conference.</p>	<p>Similar to <i>OSCon</i> in its evolution from meetings of different user groups</p>

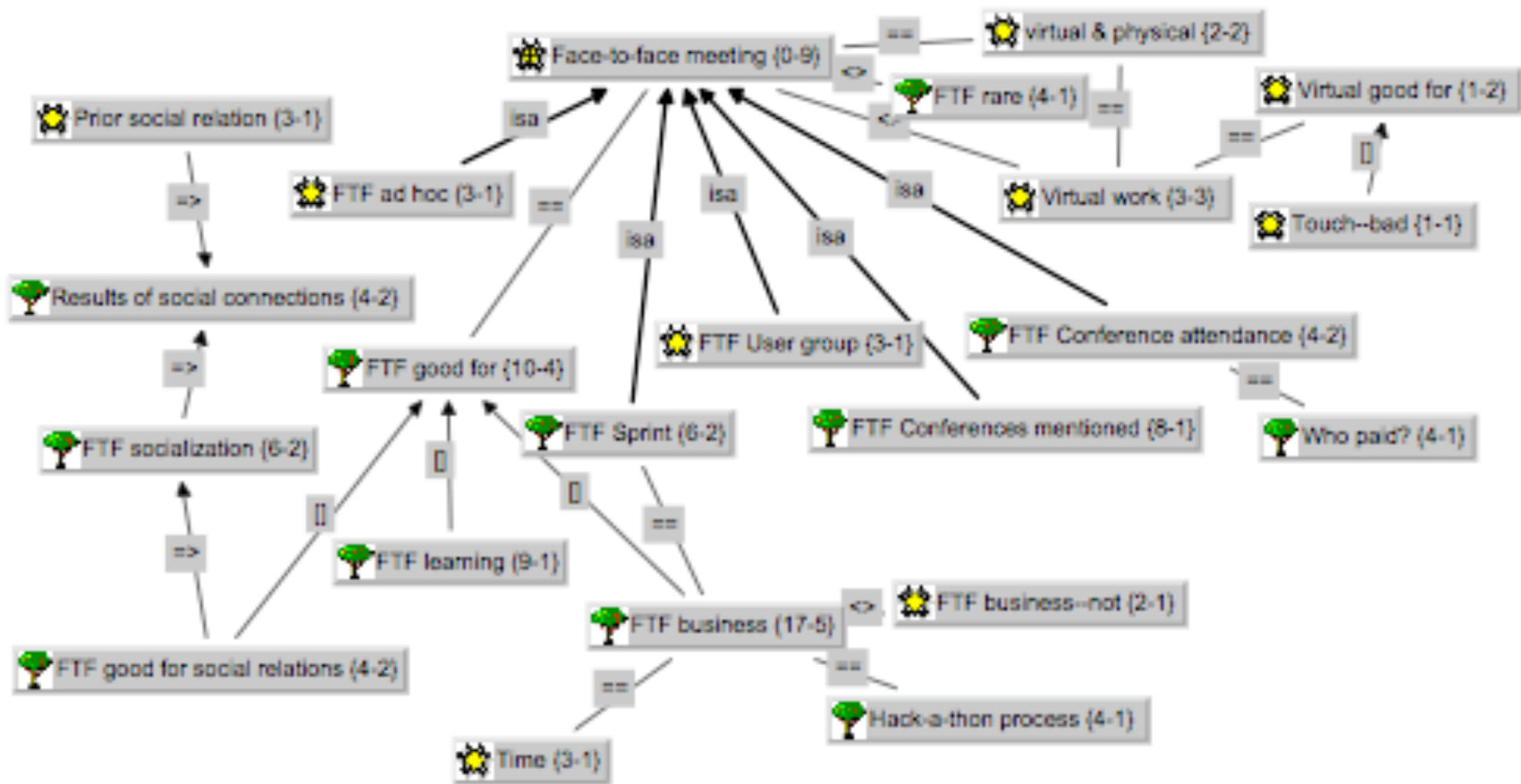


Figure 2. Network of codes induced from data analysis.



Figure 3. *ApacheCon* hack-a-thon.



Figure 4. *ApacheCon* paper presentation.



Figure 5. Key signing at a conference.



Figure 6. Design work at a whiteboard.

Appendix A: Observation questions

- What exactly do the group members do while at the conference? Do they develop code, talk about project plans and goals, socialize? Is the conference a time to write complicated portions of code, talk about goals and directions, or just have fun and develop cohesion?
- The amount of work done (if any) and how things accomplished at the conference are communicated to the rest of the group.
- Differences in roles inferred from interactions.
- Norms that can be inferred from the interactions.
- Are there any new members introduced at the conference.

Appendix B: Interview Questions

1. General/demographic
 - a) What projects are you involved in?
 - i) How big is it, how long has it been going, URL.
 - b) What is your role in these projects?
 - c) What is the level of your involvement in these projects?
 - d) How many people do you work with in your project/sub-project?
 - e) Are you paid by your employer to work on this project?
2. Conference
 - a) Why did you decide to attend the conference?
 - i) What do you hope to get out of being here?
 - b) Have you learned anything new (about project, your team members or the ASF)?
 - c) Who's paying for the trip?
 - d) How do you spend your time here?
3. How do you assess whether a project is a success?
 - a) If you were to evaluate or identify successful FLOSS projects versus less successful projects, what would be your criteria of success?
 - b) What factors do you think are important for a project to be successful?
 - i) In your opinion, what's the relative importance of team composition (number, skill, experience of developers) vs. practices vs. external factors?
 - ii) Are tools or access to other resources important?
 - c) Have you ever been involved in a failed project?
 - i) If so, what do you think were the factors that lead to its failure?
4. Collective mind/shared mental models
 - a) What are your project goals as you see them?

- i) Is there a general agreement in the group about the goals of the project and what other members are working on?
 - b) When you think about your team, do you have a good sense of what needs to be done and what others are doing?
 - i) Is that understanding generally shared by other developers, or do they have different understandings?
 - ii) How are misconceptions or misinterpretations addressed when they occur?
 - c) How does the group maintain a common understanding of the code structure, design, and coding style?
 - d) Have you ever had difficulties in interpreting some of the goals or requests (by other team members, developers external to the team etc.)?
 - e) What resources are available for members when they have questions or need help completing a particular task?
5. Socialization
- a) How do you recruit/select new members?
 - b) How are new members brought in to the team and introduced to the ways things are done within the project?
 - i) Do you have any protocols established to get new team members/volunteers integrated into the team?
 - ii) How do new members get up to speed in terms of the code structure?
 - c) What resources are available for new members when they have questions or need help completing a particular task?
 - d) Does the team do anything special to be sure new members share common understandings?
 - i) Are there social events that members attend (f-t-f or otherwise)?
 - ii) Are there commonly retold war stories about the project?
6. Coordination
- a) What are the main activities within the project?
 - b) How interdependent are these activities on each other?
 - i) What are the main coordination problems you face during the development?
 - ii) How do team members coordinate their work?
 - iii) Can you remember a time when your work was interrupted because you were waiting for something from another project member?
 - (1) Can you recall how that situation was resolved?
 - (2) Was that situation something that you could live with or did you want to stop that sort of thing happening again?
 - iv) Can you think of a time in which you were waiting for something from another project whose code yours uses?
 - c) What guides what you work on?
 - i) How are tasks identified, prioritized, and assigned in your project? (does one person decide; the whole group; segment of the group; each developer individually?)
 - ii) How are the identified tasks and the assignments communicated to the group?
 - iii) Do you use any software tools to help this process?

- d) Problem solving
 - i) Who identifies problems? Are they internal or external to the team?
 - ii) How many people identify plans/new features/new functionality to be added? Are they internal or external to the team? Who decide if to accept those contributions?
 - e) Process
 - i) Would you describe the process of adding a new feature?
 - ii) Would you describe the process of fixing a bug?
 - iii) Has your project developed any unique processes or practices that you think are particularly helpful?
 - f) Written rules and procedures?
 - i) Does the project have written procedures/rules/guidelines that determine task management or members' involvement and contributions?
 - ii) Describe is the process by which these written procedures/rules/guidelines are developed?
7. Roles
- a) What roles exist in the group?
 - i) What are the responsibilities of such roles?
 - b) How were these roles assigned or did they emerge?
 - c) How are these role modified?
8. Conflict management
- a) How does your team deal with conflicts?
 - b) How does the team resolve problems?
 - c) If, during problem solving, different alternatives emerge, who makes the decision?
9. Corporate involvement
- a) Is there corporate involvement in your project?
 - i) What type of involvement?
 - b) Do you know which other developers are corporately funded?
 - i) Does knowing who is corporately funded change your attitude to them or their code?
 - ii) If you do know corporately funded developers, are you interested in their institutional situation (e.g., who's their boss, why (strategically) the company is involved)?
10. Communication/ organizational structure
- a) By what means do you communicate with your team members (telephone, email, IRC)?
 - b) What percentage would you give each of these?
 - c) Do you meet other members of your project Face-to-Face? Which ones? How often?

- d) Do you find that you use different online mediums for specific communication tasks (i.e. do you find you use IRC for brain storming, email for scheduling tasks etc.)?
- e) Are records kept of all communications?
- f) Can a new team member or non-member access these records?

11. Time spent on FLOSS Development

- a) How many hours a week do you work on FLOSS?
- b) How did you get started with FLOSS?
- c) (If they work for a company) Does your company release all software that they incorporate FLOSS back into the FLOSS community or do they make some of it propriety?
- d) Are you involved in any FLOSS projects simply for the fun of it without a work angle (I ask because I get the impression that most of the people actually do so because they make a living out of it)?
- e) What percentage of your income is derived from FLOSS related work?
- f) What is your educational background?

12. Future of FLOSS

- a) How do you see the future of FLOSS development?
- b) What is your feeling on software patents?
- c) What do you think are the dangers and benefits of the increasing involvement of big companies in FLOSS?
- d) Have you felt any changes since a corporate sponsor got involved?
- e) Do you feel that the involvement of novices in FLOSS leads to a beneficial bazaar effect or more to a town meeting effect?

13. Research questions

- a) What kind of research or research results you think would be beneficial to your project or the FLOSS community in general?
- b) What do you think of our research questions?