Face-to-face interactions in self-organizing distributed teams [1]

Kevin Crowston, James Howison, Chengetai Masango and U. Yeliz Eseryel

Syracuse University School of Information Studies

crowston@syr.edu, jhowison@syr.edu, cmasango@syr.edu, uyeserye@mailbox.syr.edu

Please ask before citing or distributing

Draft of 23 January 2005

[1] This research was partially supported by NSF Grants 03-41475 and 04–14468. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. This paper has benefited from suggestions from Kathy Chudoba. The authors also thank the FLOSS developers who contributed to the research.

Face-to-face interactions in self-organizing distributed teams

Abstract

We explore the role of face-to-face meetings in the life of distributed teams using data from Free/Libre Open Source Software (FLOSS) development teams. Such distributed teams are part of many organizations' new vision of management in the 21st century. Practitioner research has suggested the need for face-to-face meetings when a team is formed, but few studies have considered the role of face-to-face meetings during a team's life. Based on a qualitative inductive analysis of data from interviews and observations at FLOSS conferences, we identify a variety of settings in which FLOSS developers meet face-to-face, activities performed in these settings and benefits obtained. Contrary to prior research, we find that FLOSS developers generally do not meet until the project is well under way. We also find that an additional benefit of face-to-face meetings is time away from a regular job. We conclude by noting limitations in our data collection due to a focus on core developers in large projects and with directions for further research.

Face-to-face interactions in self-organizing distributed teams

In this paper, we explore the role of face-to-face meetings in the life of distributed or virtual teams. Distributed teams are an important part of many organizations' vision of management in the 21st century. Practitioner research has suggested the need for face-to-face meetings when a team is formed, but few studies have considered the role of face-to-face meetings during a team's life. Our study is intended as a first step towards addressing this gap in the literature.

Theory: Hybrid-mode distributed teams

Distributed (or virtual) teams are becoming more commonly used in many organizations and researchers have begun to grapple with their concerns. Lipnack and Stamps (1997) 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". In the research community, there is a growing realization that distributed is not all or nothing (Niederman & Beise, 1999; O'Leary et al., 2002; Watson-Manheim, Chudoba, & Crowston, 2002; Gaved & Mulholland, 2005). Rather, teams fall along a continuum from traditional face-to-face to fully distributed, with many exhibiting a mixed mode of interaction.

One such mix is the partially distributed team, where one member is not co-located but rest are (e.g., Burke, Aytes, Chidambaram, & Johnson, 1999) or where the team is composed of co-located subteams (e.g., Mark & Abrams, 2005). In this paper, we examine cases in which the team (or subsets of the team) meets in different modes at different times. As Dubé and Paré note, "most teams in real-life settings rely, to some extent, on face-to-face meetings to ease the process of collaboration and coordination" (2004). Ocker et al. (1998) found that mixed mode teams out performed both face-to-face and distributed teams. Indeed, Qureshi goes as far as to say that, "collaborative technologies in virtual environments enable better face-to-face meetings" (2001).

The practitioner literature on distributed teams tends to emphasize a need for face-to-face contacts at the start of a project. For example, Anschuetz (1998) suggests that:

> Initial face-to-face meetings and social contacts are best for jumpstarting trust, but even a videoconference in which team members see one another and email 'introductions' that provide some personal background have proven effective.

This theme is echoed in the research literature. Furst, Blackburn and Rosen, (1999) note that:

> Popular discussions of virtual teams (Lipnack & Stamps, 1997) suggest the need for at least one early face-to-face meeting of virtual team members to begin the norm development process.

Dubé and Paré (2004) similarly state that:

> starting a new project with a face-to-face meeting is a highly useful investment. Such a meeting provides the necessary clarity of focus and direction and gives people a chance to establish relationships and develop a sense of belonging to the team.

A few researchers have examined the role of face-to-face meetings during the life of distributed teams. These studies have identified two roles for face-to-face meetings.

First, echoing the function of kick-off meetings, researchers have noted that face-to-face communications are important to sustain social relationships that enable distributed work. Nardi and Whittaker (2002) 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: (a) touching; (b) eating and drinking together;
> (c) engaging in mutually meaningful experiences in a common physical space;
> (d) 'showing up' in person.

These relationships help team members develop shared understandings and perform better as a team. Hinds and Weisband (2003) suggest that developing a shared understanding in virtual teams through shared experiences can influence the ability of teams to co-ordinate work and perform well. Similarly, Similarly, Huang, Carte and Chidambaram (2004) found that group cohesiveness improved performance.

Since these ties degrade over time (Nardi et al., 2002), periodic travel seems to be necessary to establish and maintain ties (Schwarz, Nardi, & Whittaker, 1999). Maznevski and Chudoba (2000) 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.

Second, researchers have noted certain kinds of work are more suited to face-to-face meetings. Maznevski and Chudoba (2000),

> speculate 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.

One team in their study meet by conference call regularly and face-to-face for two days every four months. Maznevski and Chudoba describe the purpose of these meetings as "to manage the future development of the contract by sharing plans and information, generating ideas about co-development" (in addition to "building strong relationships through social meals and breaks"). In discussing the advantages of face-to-face work, Olson et al. (2002) note that "when people had questions, often the person who could answer it (e.g., the customer, the tutor, a fellow worker who had more experience or expertise on a topic) was at hand" (p. 120) and that "the awareness afforded in collocation also allowed people to engage in informal training sessions" (p. 119).

In summary, existing research on face-to-face interactions in distributed teams suggests that these interactions will be used first for socialization to build the team and second for work activities that are more appropriate for face-to-face interaction.

Data

To 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. In this section, we first describe FLOSS development in general and then our data collection efforts in more detail.

Research setting

FLOSS[2] 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).

In analyzing the work of FLOSS development, much of the literature on FLOSS has conceptualized developers as forming communities, which is a useful perspective for understanding why developers choose to join or remain in a project. However, for our research, we have chosen to analyze developers as comprising a work team. Guzzo and Dickson (1996) 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 team, 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 projects are entities that have a goal of developing and maintaining a product, whose members are interdependent in terms of tasks and

[2] FLOSS software is generally available without charge ("free as in beer"). Some (though not all) OSS software is also "free software", meaning that 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 rather than the more common OSS to accommodate this range of meanings.

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.

More specifically, FLOSS projects are examples of distributed teams, albeit ones with a few unique features. Developers contribute from around the world, meet face-to-face infrequently and coordinate their activity primarily by means of computer-mediated communications (CMC) (Raymond, 1998; Wayner, 2000). Indeed, it is commonly held that teams do not meet face-to-face at all, but as we show below, many developers do. The teams have a high isolation index (O'Leary & Cummings, 2002) in that most team members work on their own and in most cases for different organizations (or no organization at all). As a result, these teams depend on processes that span traditional boundaries of place and ownership (Watson-Manheim, Chudoba & Crowston, 2002), providing a vision of how work might be conducted in the future. 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.

FLOSS teams have several features that set them apart from many of the distributed teams that have been studied before. First, the work of software development can be executed entirely on-line, which is not the case for many other kinds of work (e.g., as discussed by Furst, Blackburn, & Rosen, 1999). However on-line-only work might be typical of other "knowledge economy" industries. As well, many (though by no means all) programmers contribute to projects as volunteers, without working for a common organization or being paid. Finally, the teams are largely self-organizing, often without formally appointed leaders or indications of rank or role. These features make FLOSS teams extreme examples, but they are not inconsistent with what many organizations are facing in recruiting and motivating professionals and in developing distributed teams.

Finally, FLOSS development projects include different classes of members with distinct roles and patterns of engagement, which may be different from organizational sponsored teams in which members are expected to contribute with more-or-less equal intensity. Several authors have described teams as having a hierarchical or onion-like structure (Cox, 1998; Moon & Sproull, 2000), 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. 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 outer circle is indistinct because the nature and variety of FLOSS distribution channels makes it difficult or impossible to know the exact size of the user population. As their involvement with a project changes, individuals may move from role to role. However, core developers must have a deep understanding of the software and the development processes, which poses a significant barrier to entry (Fielding, 1997; Gacek & Arief, 2004; Hecker, 1999).

Despite the heavy reliance on CMC-supported work, our research has shown that many members of FLOSS teams both have regular opportunities to meet and do indeed meet. The overall goal of our research project is to understand the work practices of effective FLOSS teams. In this paper, we seek to understand the role that face-to-face meetings play in the practices of FLOSS teams through out their life. In the following section, we review the literature on distributed teams, before turning to our data collection and analysis and then to a discussion of our results.

Data collection

Our primary data comes from interviews and observation of face-to-face meetings of FLOSS developers at several conferences, specifically *ApacheCon* 2003 and 2004, *Comdex* 2003, *PloneCon* 2004, *OSCon* 2004 and *OSDC* 2004. In the remainder of this section, we briefly describe these conferences and our data collection before turning to the analysis and a presentation and discussion of our findings.

ApacheCon is 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. The conference runs Sunday to Wednesday, with Sunday devoted to tutorials and Monday through Wednesday 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. *Comdex* 2003 included an Open

Source pavilion where 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. *PloneCon* is an annual meeting of the developers of the Plone Content Management System and is thus specific to a single project. *OSCon* is the O'Reilly Open Source Conference. The conference grew out of the Perl Conference and is still attended by many Perl users, but now includes tracks on other languages and systems. The conference has the same general format as the *ApacheCon*. *OSDC* is the Open Source Developers' Conference, an Australian conference similar to *OSCon* in its evolution from meetings of different user groups.

In the course of these conferences, we conducted formal and informal interviews with 27 developers, representing 22 FLOSS projects (including Apache httpd, SpamAssassin, Perl, Plone and Mozilla). Interviews were semi-structured, starting with a list of questions, 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. In addition, we observed the interaction of developers throughout the conferences and took notes about these.

Analysis

To analyze interview transcripts and notes, we applied a qualitative inductive analysis technique. The analysis was supported using Atlas-ti. 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. However, the categories evolved through the course of the data analysis. As we coded each segment, we decided whether the segment fit an existing code, required a new code or required revision of the existing codes. We continued to revise the codes until each segment fit cleanly within some category. These codes were then grouped into higher-level categories and the relationships between these codes elaborated. The process resulted in three main sets of codes: opportunities for face-to-face encounters, activities during face-to-face encounters and results of these activities.

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 perceived benefits of such encounters.

Opportunities for face-to-face meetings

Our first finding is that developers generally described face-to-face meetings as rare, consistent with typical descriptions of FLOSS development. For example, we were told (typewrite font indicates quotations from interviews):

> Except the conferences, no well, I mean we work for the same company, so now I am more likely to see him, but except you know, usually its conference. Although, we get couple a year maybe two or three.

> Face-to-face meeting are rare and so may be once or twice a year if you are lucky.

Others described not meeting fellow developers until well into a project, in sharp contrast to the generally accepted recommendation for a kick-off meeting:

> [I] worked on it for three years, maybe two years, for a couple of years and before I met anybody else.

Even at the conferences we attended, not all members of teams were present: 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.

Nevertheless, our interviewees described a variety of settings in which developers might meet face-to-face, including conferences, project meetings and "sprints". The most salient of these settings were conferences (the context of our interviews). In addition to the three we attended ourselves, respondents mentioned many others, including *ApacheCon* Europe, the *Ottawa Linux Symposium*, the *DebianConf*, *CodeCon*, *FOSDEM* (Europe), *EclipseCon*, the *MySQL AB Conference*, *PyCon*, the *GeekCruise* and "*Yet Another Perl Conference*". Developers might also meet at specialized conferences such as anti-Spam conferences. Maznevski and Chudoba (2000) note the importance of a regular temporal pattern for face-to-face meetings, something that would be provided by an annual meeting (just as it is for academics).

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. As at many conferences, attendees received a conference bag; however, the *ApacheCon* bag included “geek toys” (aka “swag”), such as T-shirts, pop rocks, an alien head, and pop-bottle-bottom glasses (see Figure 2). The conference reception included appearances by Star Trek characters (see Figure 3). 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 Figures 4 and 5).

A second kind of meeting is a project meeting. In many cases, project meetings are held as adjuncts to 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 meetings. For example, one Apache developer described the original meeting to discuss the possibility of forming a non-profit foundation (the ASF).

ApacheCon began with an official hack-a-thon, sponsored by IBM, during which developers took over a conference room for the weekend to work in small teams (see Figure 4). The conference organizers provided wireless Internet access in the conference rooms for the laptops brought by almost all participants. The hack-a-thon did not have a formal schedule or program; indeed, the only formal announcements we observed during the two days of hack-a-thon were to announce deliveries of pizza and donuts. Since participants came and left in an informal fashion it was not possible to count the total number of participants attending. We did, however, make a regular count of participants. 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 members meeting on the afternoon of the second day.

A few face-to-face meetings are 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. We interviewed a number of participants from the Plone project who provided details about their sprints (though we have not yet been able to attend one). Many Plone developers are consultants who install Plone for their clients. A sprint is organized by a developer to develop new functionality required by a client. The new functionality is contributed back to the project, extending its capabilities. Sprint attendees are reported to get clients to fund their trips; for a recent sprint, we were told that the host also got $10,000 funding to cover costs and provided accommodation for the group in an Austrian castle. Sprints were said to have their origins in the Zope community (Plone is based on Zope), in particular in meetings between in-house developers from the Zope Corporation and external Zope developers.

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 committers to be respectful friends,
> meet up with each other as often as possible. Face-to-face is superior
> to email.

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

Activities during and benefits of face-to-face meetings

We observed a variety of activities during face-to-face meetings, with associated benefits. In this section, we discuss these activities and benefits together.

Socializing

A main benefit of face-to-face meetings is the opportunity to socialize with team members. Many interviewees reported spending time on socializing before 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,

[3] http://codehaus.org/Manifesto

some with a company-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; during the dinners we observed, the conversation was primarily social as opposed to work-related.

As expected, time spent socializing was important for building and maintaining personal relationships. As Nardi and Whittacker 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 by which they are known on-line. At one social event, we observed two attendees talking to each other for about ten minutes before one of them mentioned his ID to the other. The other immediately changed his neutral stance and gave his ID. They both exclaimed, shook hands, reintroduced themselves again and restarted their conversation in a more upbeat fashion with a totally different energy, once they had connected their online personas with their physical ones.

Many interviewees described the advantage of having met other developers on previous occasions. The developer mentioned above 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 meet 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 (Gaved & Mulholland, 2005).

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 the 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 from the attendance fees paid largely by users, 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, for the first time, 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 into resources to support developer team building.

Work

In addition to time spent socializing, 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. 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 Olson et al. (2002). 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 individual developers who had been unable to make the trip, again reflecting the mixing of face-to-face and distributed modes of work.

We also observed many examples of individuals and small groups actually coding. Indeed, one project team completed a new release of their system during the *ApacheCon* conference. 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.

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 developers noted work had been explicitly put off until the face-to-face meeting. 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 projects. An infrastructure team member commented that:

> *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, for developers, the team member commented that:

> It [non face-to-face work] slows them down, which is good

Indeed, other authors argued that the distributed nature of FLOSS development may actually lead to more robust and maintainable code. Because developers cannot consult each other easily, it may be that they make fewer assumptions about how their code will be used and thus write more robust code that is highly modularized (Lee & Cole, 2003).

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*.

On the other hand, contrary to the findings of Maznevski & Chudoba (2000), 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. More systematic observation would be necessary to identify more precisely what kinds of tasks are undertaken in face-to-face meetings.

Training

We observed some informal examples of training (in addition to the training, such as formal paid tutorials, that is characteristic of user group meetings), as expected by Olson et al. (2002). 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 could discuss how a particular feature had been implemented in each project, thus providing an opportunity for cross-fertilization between projects that otherwise might not happen.

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 whose 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", while another 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. 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.

Discussion

In summary, our interviews and observations suggest that face-to-face meetings, while not frequent, do occur regularly and serve important functions within FLOSS development teams. Face-to-face socialization helps to build and maintain social ties that developers report facilitate interactions. Certain kinds of team activities are easier to accomplish with face-to-face interactions, enough so that in a few cases they may actually be deferred until a planned meeting. Finally, spending time at a meeting allows many attendees to focus their attention on a project. The final observation suggests that the relevant theories for studying FLOSS teams may include literature on volunteer organizations in addition to the corporate organizations. On the other hand, we also noted cases where the face-to-face interaction was supported by or linked to CMC-supported interactions, suggesting a true hybrid mode of interaction.

There are limitations of our data on the role of face-to-face meetings. In particular, a major 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 (i.e., an attendee from Microsoft who was just there to observe). More systematic data is needed though to fully understand these interactions.

As well, our observation of developer interactions is primarily based on Apache, a large and particularly well-organized project. We believe that several other projects have similar interactions that we were not been able to observe 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. Again, more systematic data collection is needed.

Conclusion

We conclude with a few observations about the generality of our findings and directions for possible follow on research. As we noted, our data is drawn primarily from large projects and we believe that many of these projects have regularly scheduled opportunities for face-to-face meetings of developers. However, we cannot claim that this pattern is 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 organizational distributed teams). The most interesting case for further study then may 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. 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.

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. 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 would be needed.

Our results so far indicate the need for further research. First, 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

Anschuetz, L. 1998. Managing Geographically Distributed Teams. Proceedings of the IEEE Professional Communication Society.

Burke, K., Aytes, K., Chidambaram, L., & Johnson, J. J. 1999. A study of partially distributed work groups: The impact of media, location, and time on perceptions and performance. Small Group Research, 30(4): 453–490.

Cox, A.; Cathedrals, Bazaars and the Town Council; http://slashdot.org/features/98/10/13/1423253.shtml; 22 March, 2004.

Dubé, L. & Paré, G. 2004. The Multifaceted Nature of Virtual Teams: Idea Group.

Fielding, R. T.; The Apache Group: A case study of Internet collaboration and virtual communities; http://www.ics.uci.edu/fielding/talks/ssapache/overview.htm.

Furst, S., Blackburn, R., & Rosen, B. 1999. Virtual team effectiveness: A proposed research agenda. Information Systems Journal, 9: 249–269.

Gacek, C. & Arief, B. 2004. The many meanings of Open Source. IEEE Software, 21(1): 34–40.

Gaved, M. & Mulholland, P. 2005. Grassroots Initiated Networked Communities: A Study of Hybrid Physical/Virtual Communities, In Proceedings of the 38th Hawaii International Conference on System Sciences, IEEE Press.

Guzzo, R. A. & Dickson, M. W. 1996. Teams in organizations: Recent research on performance effectiveness. Annual Review of Psychology, 47: 307–338.

Hecker, F.; Mozilla at one: A look back and ahead; http://www.mozilla.org/mozilla-at-one.html.

Hinds, P. & Weisband, S. 2003. Knowledge sharing and shared understanding in virtual teams. In C. B. Gibson & S. G. Cohen (Eds.), Virtual teams that work: Creating conditions for virtual teams effectiveness: 21–36. San Francisco: Jossey-Bass.

Huang, R., Carte, T. A., & Chidambaram, L. 2004. Cohesion and Performance in Virtual Teams: An Empirical Investigation, Proceedings of the Tenth Americas Conference on Information Systems. New York, NY.

Lee, G. K. & Cole, R. E. 2003. From a firm-based to a community-based model of knowledge creation: The case of Linux kernel development. Organization Science, 14(6): 633–649.

Lipnack, J. & Stamps, J. 1997. Virtual teams; Reaching across space, time and organizations with technology. New York, NY: John Wiley and Sons, Inc.

Mark, G. & Abrams, S. 2005. Differential Interaction and Attribution in Collocated and Distributed Large-scale Collaboration, Proceedings of the 38th Hawaii International Conference on System Sciences: IEEE Press.

Maznevski, M. L. & Chudoba, K. M. 2000. Bridging space over time: Global virtual team dynamics and effectiveness. Organization Science, 11(5): 473–492.

Moon, J. Y. & Sproull, L. 2000. Essence of distributed work: The case of Linux kernel. First Monday, 5(11).

Nardi, B. A. & Whittaker, S. 2002. The place of face to face communication in distributed work. In P. Hinds & S. Kiesler (Eds.), Distributed Work: New Research on Working across Distance Using Technology: 83–110. Cambridge, MA: MIT Press.

Niederman, F. & Beise, C. 1999. Defining the 'virtualness' of groups, teams, and meetings, Proceedings of the 1999 ACM SIGCPR Conference: 14–18. New Orleans, LA.

Ocker, R. J., Fjermestad, J., Hiltz, S. R., & Johnson, K. 1998. Effects of four modes of group communication on the outcomes of software requirements determination. Journal of Management Information Systems, 15(1): 99–118.

O’Leary, M. & Cummings, J. 2002. The Spatial, Temporal, and Configurational Characteristics of Geographic Dispersion in Teams. Paper presented at the Academy of Management Conference, Denver, CO.

Olson, J. S., Teasley, S., Covi, L., & Olson, G. 2002. The (currently) unique advantages of collocated work. In P. Hinds & S. Kiesler (Eds.), Distributed Work: 113–135. Cambridge, MA: MIT Press.

Qureshi, S. & Zigurs, I. 2001. Paradoxes and perogatives in global virtual collaboration. Communications of the ACM, 44(12): 85–88.

Raymond, E. S. 1998. The cathedral and the bazaar. First Monday, 3(3).

Schwarz, H., Nardi, B. A., & Whittaker, S. 1999. The hidden work in virtual work. Paper presented at the International Conference on Critical Management Studies, Manchester, UK.

Watson-Manheim, M. B., Chudoba, K. M., & Crowston, K. 2002. Discontinuities and continuities: A new way to understand virtual work. Information, Technology and People, 15(3): 191–209.

Wayner, P. 2000. Free For All. New York: HarperCollins.

Figures

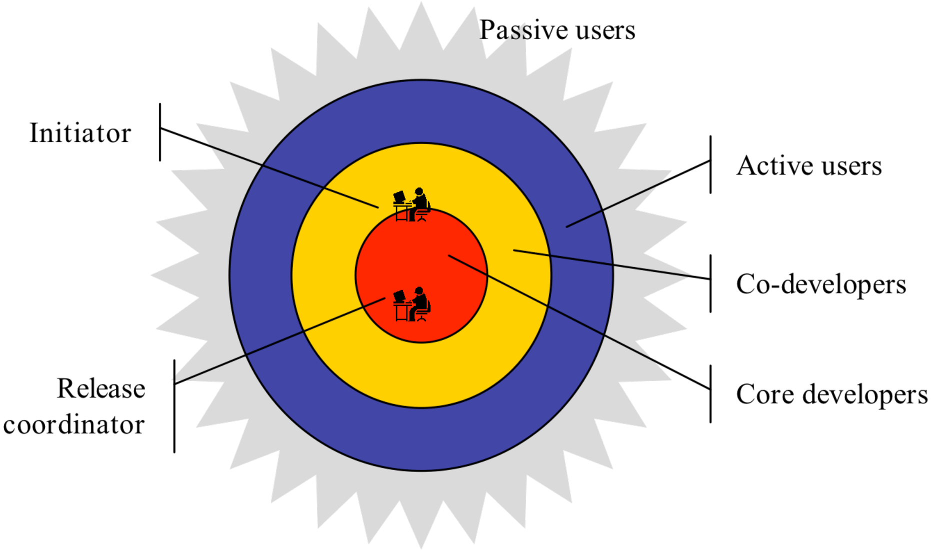

Figure 1. Hypothesized FLOSS team structure.

ApacheCon U.S. 2004
Sun
Novell
IBM
NERD SPECS
POP ROCKS
Nerds
STAR TREK THE EXPERIENCE
BORG
The Apache Software Foundation
http://www.apache.org
APACHECON 2004
ApacheCon U.S. 2004
"THE ONLY SURE THING IN LAS VEGAS"
The Apache Software Foundation Welcomes you to
ApacheCon U.S. 2004
Las Vegas, Nevada • November 13-17, 2004
http://www.apachecon.com/

Figure 2. *ApacheCon* conference "swag": Alien head, pop bottle eyeglasses.

Figure 3. *ApacheCon* reception included appearances by Star Trek characters.

Figure 4. *ApacheCon* hack-a-thon.

Figure 5. *ApacheCon* paper presentation.

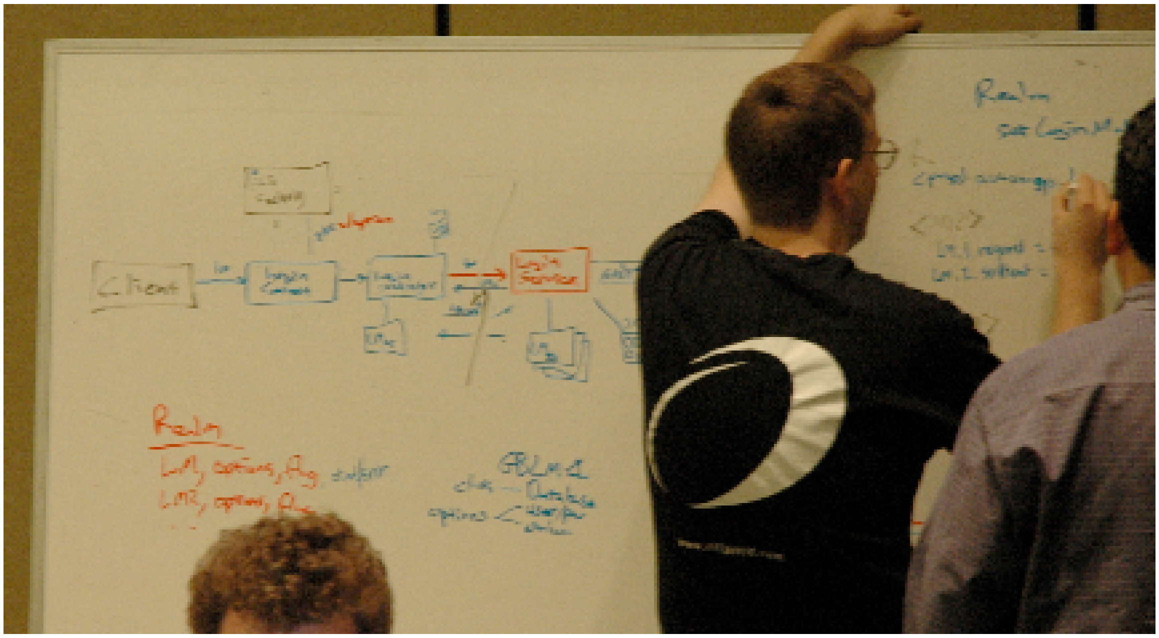

Figure 6. Design work at a whiteboard.