Free Software Development: Cooperation and Conflict in A Virtual Organizational Culture

Margaret S. Elliott
Institute for Software Research
University of California, Irvine
Irvine, CA 92697
949 824-7202
melliott@ics.uci.edu

Walt Scacchi
Institute for Software Research
University of California, Irvine
Irvine, CA 92697
949 824-4130
wscacchi@ics.uci.edu

August 2003
Previous version: May 2003

Revised version submitted to:
S. Koch (ed.), *Free/Open Source Software Development*, IDEA Publishing, 2004.

Acknowledgements: The research described in this report is supported by grants from the National Science Foundation #IIS-0083075, #ITR-0205679 and #ITR-0205724. No endorsement implied. Mark Ackerman at the University of Michigan Ann Arbor; Les Gasser at the University of Illinois, Urbana-Champaign; John Noll at the Santa Clara University; Chris Jensen, Mark Bergman, and Xiaobin Li at the UCI Institute for Software Research, and also Julia Watson at The Ohio State University are collaborators on the research project that produced this chapter.

Free Software Development: Cooperation and Conflict in A Virtual Organizational Culture

1 Introduction

Free/open source software development (F/OOSD) projects are growing at a rapid rate. The SourceForge Web site estimates 600,000+ users with 700 new ones joining every day and a total of 60,000+ projects with 60 new ones added each day. Thousands of F/OOSD projects have emerged within the past few years (DiBona, *et al*., 1999; Pavlicek, 2000) leading to the formation of globally dispersed virtual communities (Kollock and Smith, 1999). Examples of open software projects are found in the social worlds that surround computer game development; X-ray astronomy and deep space imaging; academic software design research; business software development; and Internet/Web infrastructure development (Elliott, 2003; Elliott and Scacchi, 2002; Elliott and Scacchi, 2003; Scacchi 2002a, 2002b). Working together in globally distributed virtual communities, F/OSS developers communicate and collaborate using a wide range of web-based tools including Internet Relay Chat (IRC) for instant messaging, CVS for concurrent version control (Fogel, 1999), electronic mailing lists, and more (Scacchi, 2002b).

Proponents of F/OSS claim advantages such as improved software validity, simplification of collaboration, and reduced software acquisition costs. While some researchers have examined F/OOSD using quantitative studies exploring issues like developer defect density, core team size, motivation for joining free/open source projects, and others (Koch and Schneider, 2000; Mockus *et al*., 2000, 2002), few researchers have explored the social phenomena surrounding F/OOSD (Berquist, M. and J. Ljungberg, 2001; Mackenzie *et al.,* 2002). While the importance of understanding the culture of FOSS developers has been discussed in popular literature (Pavlicek, 2000; Raymond, 2001), no researchers have articulated the work culture of F/OOSD in a virtual organization. In this chapter, we present the results of a virtual ethnography to study the work culture and F/OOSD work processes of a *free software* project, GNUenterprise (GNUe) (http://www.gnuenterprise.org). We identify the beliefs and values associated with the free software movement (Stallman, 1999a) which are manifested into the work culture of the GNUe community and we show the importance of computer-mediated communication (CMC) such as chat/instant messaging and summary digests in facilitating teamwork, resolving conflicts, and building community.

The free software movement promotes the production of free software that is open to anyone to copy, study, modify, and redistribute (Stallman, 1999b). The Free Software Foundation (FSF) was founded by Richard M. Stallman (known as RMS in the F/OSS community) in the 1970s to promote the ideal of freedom and the production of free software, based on the concept that source code is fundamental to the furthering of computer science, and that free source code is necessary for innovation to flourish in computer science (DiBona *et al*., 1999). It is important to distinguish between the terms free software (Stallman, 1999a) and open source (DiBona *et al*., 1999). Free software differs from open source in its philosophical orientation. RMS feels that the difference is in their values, their ways of looking at the world.

> "For the Open Source movement, the issue of whether software should be open source is a practical question, not an ethical one. As one person put it, `Open source is a development methodology; free software is a social movement.' For the Open Source movement, non-free software is a suboptimal solution. For the Free Software movement, non-free software is a social problem and free software is the solution.
> http://www.fsf.org/philosophy/free-software-for-freedom.html

A popular expression in the free software culture is "Think free speech, not free beer." The FSF promotes the use of the General Public License (GPL) for free software development as well as other similar licenses (http://www.gnu.org/licenses/license-list.html). While the majority of open source projects use the GPL, alternative licenses suggested by the Open Source Inititative (OSI) are also available (see http://www.opensource.org).

The free software movement has spawned a number of free software projects all adhering to the belief in free software and belief in freedom of choice (http://www.gnu.org) as part of their virtual organizational culture. As with typical organizations (Martin, 1992, Schein, 1992), virtual organizations develop work cultures, which have an impact on how the work is completed. Each of these free software projects basically follow the suggested work practices outlined on the FSF Web site (see http://www.fsf.org) for initiating and maintaining a free software project. However, each project may also have cultural norms generic to their particular virtual organization. Subsequently, there is a need for better articulation of how these free software beliefs and values may influence F/OOSD. Managers and developers of F/OOSD projects would benefit from an understanding of how the culture of the free software movement influences work practices. In this chapter, we present empirical evidence from the GNUe case study of the influence that beliefs and values of the free software movement have on teamwork, tool choices, and conflict resolution in a free software development project. The results show a unique picture of one free software community and how they rely on CMC for software and documentation reviews, bug fixes, and conflict resolution. As with all qualitative research (Yin, 1992; Strauss and Corbin, 1990), we do not intend to portray a generalized view of all free software development projects. However, research has shown that many F/OOSD projects follow similar procedures (Scacchi 2002b). Future research will show how closely the GNUe work culture resembles that of other free software projects.

In the section 2 we present the GNUe project, followed by research methods in section 3. In section 4, we present background and in section 5 we discuss the GNUe virtual organizational culture with a conceptual diagram followed a description of the cases in section 6. Next we present a discussion of the data in section 7 followed by recommendations in section 8. We finish the chapter with section 9 on future research and section 10 as conclusions.

2 GNUe Project

GNUe is a meta-project of the GNU (http://www.gnu.org) Project. GNUe is organized to collect and develop free electronic business software in one location on the Web. The plans are for GNUe to consist of:

1. a set of tools that provide a development framework for enterprise information technology professionals to create or customize applications and share them across organizations;
2. a set of packages written using the set of tools to implement a full Enterprise Resource Planning system; and
3. a general community of support and resources for developers writing applications using GNUe tools. The GNUe Web site advertises it as a "Free Software project with a corps of volunteer developers around the world working on GNUe project."

GNUe is an international virtual organization for software development (Crowston and 2002; Noll and Scacchi, 1999) based in the U.S. and Europe. This organization is centered about the GNUe Web portal and global Internet infrastructure that enables remote access and collaboration. As of the writing of this paper, GNUe contributors consist of 6 core maintainers (co-maintainers who head the project); 18 active contributors; and 18 inactive contributors. The 6 core maintainers share various tasks including the monitoring of the daily IRC, accepting bug fixes to go into a release, testing software, documentation of software, etc. Another task for these core maintainers appears to be that of trying to resolve conflicts and answering questions regarding GNUe. For the duration of the IRC logs that we studied, several core maintainers were on the IRC almost the entire day. Companies from Austria, Argentina, Lithuania, and New Zealand support paid contributors, but most of the contributors are working as non-paid participants.

3 Research Methods

This ongoing ethnography of a virtual organization (Hine, 2000; Olsson, 2000) is being conducted using the grounded theory approach (Strauss and Corbin, 1990) with participant-observer techniques. The sources of data include books and articles on OSSD, instant messaging (Herbsleb and Grinter, 1999, Nardi *et al.*, 2000) transcripts captured through IRC logs, threaded email discussion messages, and other Web-based artifacts associated with GNUe such as Kernel Cousins(summary digests of the IRC and mailing lists – see http://kt.zork.net). This research also includes data from email and face-to-face interviews with GNUe contributors, and observations at Open Source conferences. The first author spent over 100 hours studying and perusing IRC archives and mailing list samples during open and axial coding phases of the grounded theory. During open coding the first case study presented here was selected as representative of the strong influence of cultural beliefs on GNUe software development practices. The selection of cases was aided by the indexing of each Kernel Cousin into sections labeled with a topic. For example, we read through all Kernel Cousins looking mainly at the indices only and found the following title “Using Non-Free Tools for Documentation” in (http://kt.zork.net/GNUe/gnue20011124_4.html). Hyperlinks from this cousin pointed us to a similar case where non-free tools were being used for documentation of code. The third case was found by coding the last file in the three day series for the case two debate. In the third case, a newcomer asks for help regarding the use of GNUe and we show how cooperation and community building are facilitated by the use of IRC.

The initial research questions that formed the core of the grounded theory are:

1) How do people working in virtual organizations organize themselves such that work is completed?
2) What social processes facilitate open source software development?
3) What techniques are used in open source software development that differ from typical software development?

We began this research with the characterization of open source software communities as communities of practice. A community of practice (COP) is a group of people who share similar goals, interests, beliefs, and value systems in a common domain of recurring activity or work (Wenger, 1998). An alternative way of viewing groups with shared goals in organizations is to characterize them as organizational subcultures (Trice and Beyer, 1993; Schein, 1992; Martin, 2002). As the grounded theory evolved, we discovered rich cultural beliefs and norms influencing "geek" behavior (Pavlicek, 2000). This led to us to the characterization of the COPs as virtual organizations having organizational cultures.

We view culture as both objectively and subjectively constrained (Martin, 2002). In a typical organization, this means studying physical manifestations of the culture such as dress norms, reported salaries, annual reports, and workplace furnishings and atmosphere. In addition, subjective meanings associated with these physical symbols are interpreted. In a virtual organization, these physical cultural symbols are missing, so we focus on unique types of accessible manifestations of the GNUe culture, such as Web site documentation and downloadable source code. We use the grounded theory approach to build a conceptual framework and develop a theory regarding the influence of organizational culture on software development in a free software project (Strauss and Corbin, 1990). Data collection includes the content analysis of Web site documents; IRC archives; mailing lists; kernel cousins; email interviews; and observations and personal interviews from open source conferences.

During the open coding, we interpreted books and documents as well as Web site descriptions of the OSSD process. We discovered strong cultural overtones in the readings and began searching for a site to apply an analysis of how motivations and cultural beliefs influenced the social process of OSSD. We selected GNUe as a research site because it exemplified the essence of free software development providing a rich picture of a virtual work community with a rapidly growing piece of downloadable free software. The GNUe Web site offered access to downloadable IRC archives and mailing lists as well as lengthy documentation - all facilitating a virtual ethnography. We took each IRC and kernel cousin related to the three cases and applied codes derived from the data (Strauss and Corbin, 1990). We used a text editor to add the codes to the IRC text logs using [Begin and End] blocks around concepts we identified such as "belief in free software". In this way, we discovered the relationships shown in Figure 1. During the axial coding phase of several IRC chat logs, mailing lists and other documentation, we discovered relationships between beliefs and values of the work culture and manifestations of the culture. In the next section we discuss the organizational culture perspective and studies relating to conflict resolution in cyberspace.

4 Background

In this section, we discuss the organizational culture perspective that is used to characterize the work culture of the virtual organization, GNUe. Next we discuss literature related to conflict resolution in virtual communities.

4.1 Organizational Culture Perspective

Popular literature has described open source developers as members of a “geek” culture (Pavlicek, 2000) notorious for nerdy, technically savvy, yet socially inept people, and as participants in a “gift” culture (Berquist and Ljungberg, 100; Raymond, 2001) where social status is measured by what you give away. However, no empirical research has been conducted to study FOSS developers as virtual organizational cultures (Martin, 2002; Schein, 1992) with beliefs and values that influence decisions and technical tool choices. Researchers have theorized the application of a cultural perspective to understand IT implementation and use (Avison and Myers, 1995), but few have applied this to the workplace itself (Dube´ and Robey, 1999; Elliott, 2000).

Much like societal cultures have beliefs and values manifested in norms that form behavioral expectations, organizations have cultures that form and give members guidelines for “the way to do things around here.” An organizational culture perspective (Martin, 2002; Schein, 1992; Trice and Beyer, 1993) provides a method of studying an organization’s social processes often missed in a quantitative study of organizational variables. Organizational culture is a set of socially established structures of meaning that are accepted by its members (Ott, 1989).

The substances of such cultures are formed from ideologies, the implicit sets of taken-for-granted beliefs, values, and norms. Members express the substance of their cultures through the use of cultural forms in organizations -- acceptable ways of expressing and affirming their beliefs, values and norms. When beliefs, values, and norms coalesce over time into stable forms that comprise an ideology, they provide causal models for explaining and justifying existing social systems. In a virtual organization, cultural beliefs and values are manifest in norms regarding communication and work issues (if a work-related community like OSSD) and in the form of electronic artifacts – IRC archives, mailing list archives, and summary digests of these archives as Kernel Cousins. Most organizational culture researchers view work culture as a consensus-making system (Ott, 1989; Trice and Beyer, 1993; Schein, 1992). In the GNUe study, we apply an integration perspective (Martin, 2002) to the GNUe community to show how beliefs and values of the free software movement tie the virtual organization together in the interests of completing the GNUe free software project (See Elliott and Scacchi, 2003 for a detailed report of the GNUe study). We present the GNUe virtual organization as a subculture of the FSF inculcating the beliefs and values of the free software movement into their everyday work.

4.2 Conflict Resolution in Virtual Communities

Researchers have attempted to understand conflict resolution in virtual communities (Kollock and Smith, 1996; Smith, 1999) in the areas of online communities and in the game world. Many others have studied conflict resolution in common work situations such as computer-supported cooperative work (CSCW) (Easterbrook, 1993). For our purposes, we are interested in virtual communities and how they resolve conflicts so this discussion does not include studies on conflict management tools.

Smith (1999) studied conflict management in MicroMUSE, a game world dedicated to the simulation and learning about a space station orbiting the earth. There were two basic classes of participants: users and administrators. Disputes arose in each group and between the two groups regarding issues like harassment, sexual harassment, assault, spying, theft, and spamming. These problems emerged due to the different meanings attributed to MicroMUSE by its players and administrators and due to the diverse values, goals, interests, and norms of the group. Smith concluded that virtual organizations have the same kinds of problems and opportunities brought by diversity as real organizations do, and that conflict is more likely, and more difficult to manage than in real communities. Factors contributing to this difficulty are: wide cultural diversity; disparate interests, needs and expectations; nature of electronic participation (anonymity, multiple avenues of entry, poor reliability of connections and so forth); text-based communications; and power asymmetry among users. On the contrary, in our GNUe study, we found that text-based communications via the archival text (IRC and Kernel Cousins) enabled the conflict resolution.

Kollock and Smith (1996) explored the implications of cooperation and conflict in Usenet groups emphasizing the importance of recognizing the free-rider problem. In a group situation where one person can benefit from the product or resource offered by others, each person is motivated not to contribute to the joint effort, instead free-riding on others' work. The authors do a detailed analysis of this free-rider problem and give suggestions for how to avoid it in Usenet groups. For example, they suggest bandwidth be used judiciously, posting useful information and refraining from posting inappropriate information as a way to better manage bandwidth. Success on a Usenet group also depends on its members following cultural rules of decorum. We explore the topic of following cultural rules in the next section by presenting the conceptual framework of the GNUe study.

5 Conceptual Diagram of GNUe Virtual Organizational Culture

The substance of a culture is its ideology – shared, interrelated sets of emotionally charged beliefs, values and norms that bind people together and help them to make sense of their worlds (Trice and Beyer, 1993). While closely related to behavior, beliefs, values, and norms are unique concepts as defined below (Trice and Beyer, 1993):

- **Beliefs** – Express cause and effect relations (i.e. behaviors lead to outcomes).
- **Values** – Express preferences for certain behaviors or for certain outcomes.
- **Norms** – Express which behaviors are expected by others and are culturally acceptable

As members of the FSF, free software developers share an ideology based on the belief in free software and the belief in freedom of choice. These beliefs are espoused in the literature on free software (Williams, 2002). The values of cooperative work and community are inferred from this research. Figure 1 shows a conceptual diagram of the GNUe case study. The causal conditions consist of the beliefs (free software and freedom of choice) and the values (cooperative work and community). The phenomenon is the free software development process – its formal and informal work practices. The interaction/action occurs on the IRC and mailing lists. It consists of 1) the conflict over the use of a non-free tool to create a graphic diagram of the emerging GNUe system design, 2) the conflict over the use of a non-free tool to create GNUe documentation. The consequences are: 1) building community; 2) resolution of conflicts with a reinforcement of the beliefs; and 3) teamwork is strengthened. The beliefs, values, and norms are described below; the consequences are presented in the Discussion section.

5.1.1 Belief in Free Software

The belief in free software appears to be a core motivator of free software developers. GNUe developers extol the virtues of free software on its Web site and in daily activity on the IRC logs. The FSF Web site has many references to the ideological importance of developing and maintaining free software (See http://www.fsf.org). This belief is manifested in electronic artifacts such as the Web pages, source code, GPL license, software design diagrams, and accompanying articles on their Web site and elsewhere. The data analysis of the GNUe cases showed that this belief varies from moderate to strong in strength. For example, those who have a strong belief in free software refuse to use any form of non-free software (such as a commercial text editor) for development purposes. The variation in strength of this variable becomes the focal point of case two.

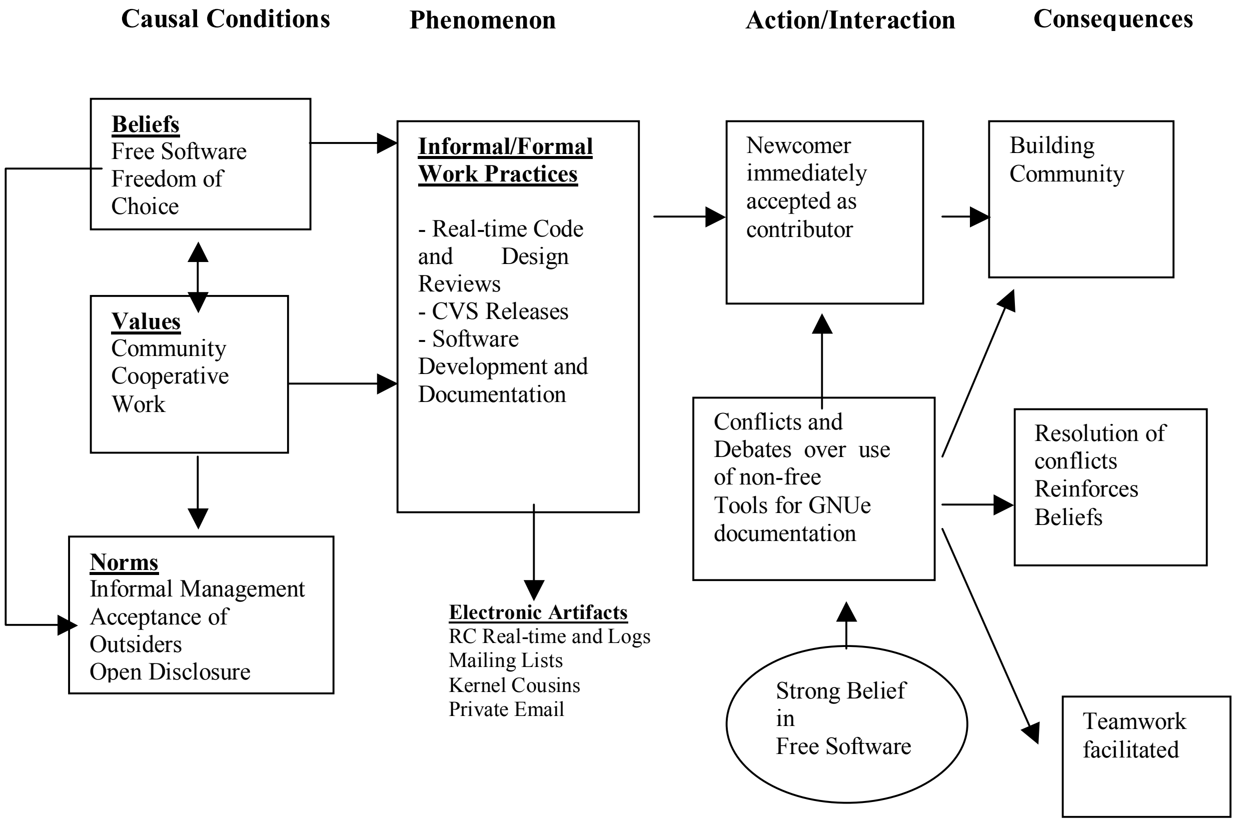

Figure 1. **Conceptual Diagram of Variables**

5.1.2 Belief in Freedom of Choice

Open source software developers are attracted to the occupation of OSSD for its freedom of choice in work assignments. Both paid and unpaid GNUe participants to some degree can select the work they prefer. This belief is manifested in the informal methods used to assign or select work in an open source project. During an interview with one of the core contributors of GNUe, Derek, at a LinuxWorld conference in August 2002, we asked how assignments were made and monitored. Derek answered with:

> "The number one rule in free software is 'never do timelines or roadmaps'."

The belief in freedom of choice also refers to the ability to select the tool of choice to develop free software. Some OSS developers believe that a mix of free versus non-free software tools is acceptable when developing free software, while others adhere to the belief in free software only.

5.1.3 Value in Community

The beliefs in free software and freedom of choice foster a value in community building as part of routine work. This value is evident in the IRC archives when newcomers join GNUe offering suggestions, or pointing out bugs, and GNUe contributors quickly accept them as part of the community. For example, when frequent contributors (insiders) have a problem with procedures or code related to free versus non-free software, the maintainers rally around the insider trying to convince him that a temporary use of non-free software is OK.

5.1.4 Value in Cooperative Work

The GNUe community's beliefs in free software and freedom of choice combined with the value in community foster a value in cooperative work. As with previous researchers (Easterbrook, 1993; Kollock and Smith, 1996; Smith, 1999), our results indicate that conflict arises during the course of cooperative work. GNUe contributors work cooperatively to resolve conflicts through the use of IRC and mailing lists.

5.1.5 Open Disclosure

Open disclosure refers to the open content of the GNUe Web site including the software source code, documentation, and archived records of IRC, kernel cousins, and mailing list interchanges. The GNUe contributors join others online via IRC on a daily basis and record the conversations for future reference. All documentation and source code are easily downloaded from the GNUe Web site and user criticism is welcomed by frequent GNUe maintainers.

5.1.6 Informal Management

The entire GNUe virtual organization is informal. There is no lead organization or prime contractor that has brought together the alliance of individuals and sponsoring firms as a network virtual organization. It is more of an emergent organizational form where participants have in a sense discovered each other, and have brought together their individual competencies and contributions in a way whereby they can be integrated or made to interoperate (Crowston and Scozzi, 2002). The participants come from different small companies or act as individuals that collectively move the GNUe software and the GNUe community forward. Thus, the participants self-organize in a manner more like a meritocracy (Fielding, 1999). There is a flow to the work determined by participants' availability.

5.1.7 Immediate Acceptance of Outsider Critiques

In the GNUe organization, outsiders who have not visited the GNUe IRC before, can easily join the discussion and give criticisms of the code or procedures. Sometimes this criticism revolves around the use of free versus non-free tools and other times it is related to attempts to fix bugs in the code. In either case, the GNUe maintainers who discuss these critiques respect and respond to outsiders reviews with serious consideration even without knowing the reviewer's credentials.

6 GNUe Case Study

The GNUe case study consists of the analysis of three cases of software development communication over the IRC. They involve 1) the debate over the use of a non-free tool for creation of a graphic; 2) the debate over the use of a non-free tool for GNUe documentation creation and maintenance; and 3) the initiation of a newcomer who fixes bugs in realtime. Each case will be described briefly in this section. For a more detailed description, see (Elliott and Scacchi, 2003).

6.1 Case One – Use of Non-Free Graphic Tool for Documentation

In this section we present the first case study that reveals a trajectory of a conflict and debate over the use of a non-free tool to create a graphic on the GNUe Web site (See http://www.gnuenterprise.org/irc-logs/gnue-public.log.25Nov2001). This exchange takes place on November 25, 2001 on the IRC channel and ends the next morning. This example illustrates the ease with which a newcomer comes onboard and criticizes the methods used to produce a graphical representation of a screenshot on the GNUe Web site. CyrilB, an outsider to GNUe, finds a graphic that was created using Adobe Photoshop, a non-free graphical tool. He begins the interchange with a challenge to anyone onboard stating that "it is quite shocking" to see the use of non-free software on a free software project. He exhibits a **strong belief in free software,** which causes a debate lasting a couple of days. Table 1 displays the total number of contributors and the number of days of the conflict. Eight of the nine regular GNUe contributors were software developers and one was working on documentation. The infrequent contributors drifted on and off throughout the day – sometimes lurking and other times involved in the discussion.

Total Contributors	**Regular Contributors**	**Infrequent Contributors**	**Number of Days**
17	9	8	1

Table 1 – Contributors and Duration of Conflict in Case One

The **strong belief in free software** of the outsider leads to conflict among those insiders who have a moderate view of the use of free software for GNUe software development. A daylong debate ensues among the Neilt, creator of the graphic, CyrilB, and other GNUe contributors regarding the use of a non-free software tool to create a graphic for a GNUe screenshot for Web site documentation.

CyrilB uses his **strong view of belief in free software** to promote the spirit of the free software movement by exclaiming that images on the gnuenterpise.org Web site seem to be made with non-free Adobe software. His reaction provokes strong reactions from GNUe contributors:

> "I hope I'm wrong: it is quite shocking…We should avoid using non-free software at all cost, am I wrong? **(Strong BIFS-1)**"

Reinhard responds with a **moderate view of belief in free software:**

> "Our main goal is to produce good free software. We accept contributions without regarding what tools were used to do the work especially we accept documentation in nearly any form we can get because we are desparate for documentation." **(Moderate View BIFS-1).**

Once CyrilB has pointed out the use of the non-free graphic, Neilt, who originally created the GNUe diagram using Adobe Photoshop, joins the IRC, reviews the previous discussion on the archived IRC, and returns to discuss the issue with Reinhard and CyrilB. A lively argument ensues between Neilt and others with onlookers contributing suggestions for the use of free tools to develop the Adobe graphic.

Meanwhile Maniac, who has been "listening" to this debate, jumps in and gives technical details about a PNG image. Then Reinhard and Neilt agree that CyrilB had a valid point since a PNG has no vector information stored and so it would be difficult to use free software to edit the graphic. These exchanges illustrate how participants use the IRC medium to support and enable the cooperative work needed to resolve this issue. It also conveys the community spirit and cooperative work ethic that is a value in the GNUe work culture. They both agree to wait until CyrilB comes back to give more suggestions for an alternative.

Outside critiques of software and procedures used during development are common to the GNUe project. One of the norms of the work culture is **immediate acceptance of outsider**

contributions. Eventually, Neilt, the creator of the non-free graphic questioned CyrilB's qualifications and was satisfied when he learned that CyrilB was a member of the European Free Software Foundation. However, he was willing to fix the graphic prior to the revelation of CyrilB's credentials.

Consequences of the debate are a reinforcement of the **belief in free software, value in community,** and **value in cooperative work;** and a recreation of a Web site graphic with free software to replace the original created with a non-free software tool.

6.2 Case Two – Use of Non-Free Software for GNUe Documentation

The second case study explores project insider review of the procedures and practices for developing GNUe documentation (See http://www.gnuenterprise.org/irc-logs/gnue-public.log.15Nov2001 for the full three day logs). Once again the debate revolves around polarized views of the use of non-free tools to develop GNUe documentation. In this case, Chillywilly, a frequent contributor, balks at the need to implement a non-free tool on his computer in order to edit the documentation associated with a current release. Even though his colleagues attempt to dissuade him from his concerns by suggesting that he can use any editor – free or non-free- to read the documentation in HTML or other formats, Chillywilly refuses to back down from his stance based on a **strong belief in free software**. This debate lasts three days. Table 2 displays the number of contributors and their classification for participation in case two. This case exemplifies the fierce adherence to the belief in free software held by some purists in the free software movement and how it directs the work of the day. While the three day debate reinforces beliefs and values of the culture, at the same time, it ties up valuable time which could have been spent writing code or documentation, yet it contributes to community building.

Total Contributors	Regular Contributors	Infrequent Contributors	Number of Days
24	9	15	3

Table 2 – Contributors and Duration of Conflict over Documentation

In order to understand this example, some background information is needed. The GNUe core maintainers selected a free tool to use for all documentation called *docbook* (http://www.docbook.org). DocBook is based on an SGML document type definition which provides a system for writing structured documents using SGML or XML. However, several GNUe developers as of November 15, 2001 were having trouble with its installation. Consequently, they resorted to using lyx tool to create documentation (http://www.lyx.org)...

The problem with lyx is that even though it was developed as a free software tool, its graphical user interface (GUI) requires the installation of a non-free graphics package (called *libxforms*). Chillywilly gets upset with the fact that he has to install non-free software in order to read and edit GNUe documentation. A lengthy discussion ensues with debates over which tool to use for GNUe documentation. This debate lasts for three days taking up much of the IRC time until Chillywilly finally gives up the argument. The strength in the **belief in free software** drives this discussion. The debate and its resolution also illustrate the tremendous effort by developers to collaborate and work cooperatively through the use of the IRC channel. Although the discussion is heated at moments, a sense of fun also pervades. Chillywilly begins on the November 14, 2001 IRC with an observation that a fellow collaborator, jamest, has made documents with lyx:

> Action: chillywilly trout whips jamest for making lyx docs
> Action: jcater troutslaps chillywilly for troutslapping jamest for making easy to do docs
> <chillywilly> lyx requires non-free software
> <Maniac> lyx rules
> <chillywilly> should that be acceptable for a GNU project?

<jcater> chillywilly: basically, given the time frame we are in, it's either LyX documentation with this release, or no documentation for a while (until we can get some other stinking system in place)
<jcater> pick one :)
<chillywilly> use docbook then

…

<Maniac> lyx's graphics library is non-gpl **(i.e. non-free software)**
<chillywilly> I'm not writing your docs for you
<Maniac> this is an issue the developers are aware of but do not, at this time, have the time to rectify
<chillywilly> Maniac: because they are **** KDE nazis
<chillywilly> that's who the original lyz authors are matthias, et. al.
<Maniac> well, my understanding is, they are working toward UI independance, to make it able to use differnt toolkits ie. kde, gnome, xyz as time/coding permit

Maniac questions chillywilly's incessant reminders about using non-free software as though this myopic view of free software development is unnecessary. Chillywilly continues his debate showing his **strong view of free software**.

Reinhard agrees with chillywilly as do others, but in order to complete the documentation, they agree to use an interim solution. Chillywilly is so adamantly opposed to the use of non-free software that he references Richard Stallman as part of his reasoning – "**I will NOT install lyx and make vrms unhappy**". This passage shows how RMS is considered the "guru" of the free software movement. Eventually chillywilly sends an email to the mailing list:

> "OK, I saw on the commit list that you guys made some LyX documents. I think it is extremely ***that a GNU project would require me to install non-free software in order to read and modify the documentation. I mean if I cannot make vrms happy on my debian system them what good am I as a Free Software developer? Is docbook really this much of a pain? I can build html versions of stuff on my box if this is what we have to do. This just irks me beyond anything. I really shouldn't have to be harping on this issue for a GNU project, but some ppl like to take convenience over freedom and this should not be tolerated… Is it really that unreasonable to request that we not use something that requires ppl to install non-free software? Please let me know. (Chillywilly, mailing list)"

A lengthy discussion of technical issues unrelated to the documentation problem ensues. Meanwhile Jcater has sent a reply to Chillywilly's message to the mailing list.:

> "I would like to personally apologize to the discussion list for the childish email you recently received. It stemmed from a conversation in IRC that quickly got out of hand. It was never our intention to alienate users by using a non-standard documentation format such as LyX. Writing documentation is a tedious chore few programmers enjoy. The developers of the GNUe client tools are no exception…The upcoming release was originally planned for this past weekend. James and I decided to postpone the release… LyX was chosen because it is usable and, more importantly, installable. After many failed attempts at installing the requirements for docbook, James and I made the decision that LyX-based documentation with the upcoming 0.1.0 releases was better than no documentation at all…
> PPS, By the way, Daniel, using/writing Free software is NOT about making RMS happy or unhappy. He's a great guy and all, but not the center of the free universe, nor the motivating factor in many (most?) of our lives. For me, my motivation to be here is a free future for my son (Jcater, mailing list)."

The belief in freedom is a motivating factor for Jcater as stated above, even freedom for his son.

6.3 Case Three – Newcomer Asking for Help with GNUe Installation

In this example, mcb30 joins the IRC as a newcomer who wants to install and use GNUe business applications for his small business in England (http://www.gnuenterprise.org/irc-logs/gnue-public.log.16Nov2001). In addition, he offers his services as a contributor and immediately starts fixing bugs in realtime. This case is a good example of the community building spirit of GNUe since mcb30 is immediately accepted by frequent contributors especially because he posts significant bug fixes very rapidly.

```
<mcb30> Is anyone here awake and listening?
<reinhard> yes
<mcb30> Excellent.  I'm trying to get a CVS copy of GNUe up and running for the first(ish) time - do you mind if I ask for a few hints?
<reinhard> shoot away :)
<reinhard> btw what exactly are you trying to run?
<reinhard> as "GNUe" as a whole doesen't exist (yet)
<reinhard> GNUe is a meta-project (a group of related projects)
<mcb30> OK - what I want to do is get *something* running so I can get a feel for what there is, what state of development it's in etc. - I'd like to contribute but I need to know what already exists first!
<reinhard> ok cool
<reinhard> let me give you a quick overview
<mcb30> I have finally (about 5 minutes ago) managed to get "setup.py devel" to work
```

```
properly - there are 2 bugs in it
<mcb30> ok
```

Mcb30 goes offline and continues to fix bugs. He then comes back and suggests that he has a patch file to help

```
<mcb30> I've got a patch file - who should I send it to?  jcater?
<reinhard> jcater or jamest
<mcb30> ok, will do, thanks
<reinhard> mcb30: btw sorry if i tell you things you already know :)
<mcb30> don't worry - I'd rather be told twice than not at all! :-)
<reinhard> people appearing here in IRC sometimes have _very_ different levels of
information :)

<reinhard> look at examples/python/addrbook.py
<mcb30> excellent, thanks!
<mcb30> will have a play around
<reinhard> mcb30: i will have to thank you
<reinhard> mcb30: we are happy if you are going to help us
<reinhard> gotta leave now
```

Later mcb30 comes back to the IRC and posts code that he wrote to fix a problem and several frequent contributors thank him and say that they wish they could hire him for pay. As with the first case, contributors immediately accept them into the "club" and, as the chat unfolds, they ask him for his credentials, motivation, and location (mcb30 is an educational consultant for IT in the English school system).

7 Discussion

The three examples from the GNUe case study will be discussed in this section in relation to the three main themes found in the data: realtime teamwork, building community, and conflict resoluton. Each example comes from a detailed conding and content analysis of the IRCs.

7.1 Building Community

Kollock (1996) suggests that there are design principles for building a successful online community such as identity persistence. He draws upon the work of Godwin (1994) showing that allowing users to resolve their own disputes without outside interference and providing institutional memory are two principles for making vritual communities work. Applying these principles to the GNUe project shows that disputes are resolved simultaneously via IRC, and recorded in IRC archives as a form of institutional memory. In the GNUe virtual community, the community is continuously changing (when newcomers join even if for a brief time) yet the core maintainers are dedicated for long periods of time. Here is a quote from Derek, a core maintainer, who believes that the IRC helps them sustain their community:

> "Many free software folks think IRC is a waste of time as there is 'goofing off', but honestly I can say its what builds a community. I think a community is necessary to survive. For example GNUe has been around for more than 3 years. I can not tell you how many projects have come and gone that were supposed be competition or such. I put our longevity solely to the fact that we have a community." (Derek, email interview (2002))

7.2 Conflict Resolution

In the two conflict resolution GNUe cases presented here, both issues resulted in a solution by debate on the IRC and mailing lists. In the first case, the contributor who created the graphic with ADOBE photoshop agreed to change it in the future using a free tool. In the second case, chillywilly stopped badgering his co-workers about the use of a non-free graphics package to complete documentation. His colleagues essentially told him to get back to work and use a text editor if he is so worried about the use of *lyx* until they all can use the free software *docbook*. In both cases, the conflicts were resolved in a reasonable amount of time via the IRC exchanges. At the same time, the beliefs in free software are reinforced by people defending their positions and this, in turn, helps to perpetuate the community.

7.3 Facilitating Teamwork

In each case there was evidence that as the day proceeded on the IRC, people were going offline to experiment with free software that would help to resolve the conflict (i.e. a free graphics package and a free text editor). Many infrequent contributors or newcomers who were lurking and watching the problem unfold on the IRC, also gave technical advice for a tool to use to solve the problem. The realtime aspect of the work clearly facilitates the teamwork since people could simultaneously work together solving a technical problem. In the third case, a newcomer who was having trouble with the GNUe installation was directed by a maintainer to the original author of the code. Surprisingly, the original author joins the IRC that day and discusses the bugs with mcb30.

8 Practical Implications

We have shown that the persistent recording of daily work using instant messaging (IRC) and Kernel Cousins can serve as a community building avenue. Managers of open source might benefit from incorporating these CMC mediums into their computing infrastructures. It assists employees in conflict management and also binds the groups together by reinforcing the organizational culture. As illustrated in the non-conflict GNUe example, the IRC serves as an expertise Q&A repository. The author of the software quickly emerged and mcb30 was able to gain detailed knowledge of how the system works. In addition, the IRC enables realtime software design and debugging. As F/OOSD projects proliferate, managers should consider the benefits of using an IRC to facilitate software development and to help build a community..

9 Future Research

We plan to continue with the analysis of GNUe data and compare the results with other free software communities. Likewise, we expect to find similar beliefs and values in some open source projects and plan to explore this phenomena. In this way, we can assertain whether GNUe is in fact a unique culture (Martin, 2002) or whether other free software projects have similar software development processes. A review of other GNU projects shows evidence of proselitization of beliefs in free software (http://www.gnu.org/projects/projects.html). In addition, in the LINUX community, there is an ongoing dispute about using Bitkeeper (non-free) versus CVS (free) as a case management system. Other future research of interest is to determine if having strong beliefs and values regarding free software contribute to a successful, productive F/OOSD community.

10 Conclusions

Previous CSCW research has not addressed how the collection of IRC messaging, IRC transcript logs, and email lists, and periodic digests (Kernel Cousins) can be collectively mobilized and routinely used to create a virtual organization that embodies, transmits, and reaffirms the cultural beliefs, values, and norms such as those found in free software projects like GNUe. Strong organizational cultural beliefs in an F/OOSD virtual community combined with persistent recordation of chat logs tie a group together and helps to build a community and perpetuate the project. The beliefs in freedom, free software, and freedom of choice create a special bond for the people working on free software projects. These beliefs foster the values of cooperative work and community-building. Schein's (1990) theory of organizational culture includes revelation of underlying assumptions of cultural members that are on a mostly unconscious level. In the GNUe world, the underlying assumptions of cooperative work and community-building become ingrained in the everyday work practices in their pursuit of an electronic business and ERP system implemented as free software. These beliefs and values enhance and motivate acceptance of outsiders' criticisms and resolution of conflict despite the distance separation and amorphous state of the contributor population.

11 References

Avison, D. E., & Myers, M. D. (1995). Information Systems and Anthropology: An Anthropological Perspective on IT and Organizational Culture. *Information Technology and People, 8*, (43-56).

Berquist, M. and J. Ljungberg, The power of gifts: organizing social relationships in open source communities, *Info. Systems J.*, 11(4), 305-320, October 2001.

Crowston, K., & Scozzi, B. (2002). *Exploring Strengths and Limits on Open Source Software Engineering Processes: A Research Agenda.* Paper presented at the *2nd Workshop on Open Source Software Engineering*, Orlando, Florida.

DiBona, C., Ockman, S., & Stone, M. (1999). *Open Sources: Voices from the Open Source Revolution*. Sebastol, CA: O'Reilly & Associates Inc.

Dubé, L., & Robey, D. (1999). Software Stories: Three Cultural Perspectives on the Organizational Practices of Software Development. *Accounting, Management and Information Technologies, 9*(4), 223-259.

Easterbrook, S. (Ed.). (1993). *CSCW: Cooperation or Conflict*. New York: Springer-Verlag.

Elliott, M. (2000). *Organizational Culture and Computer-Supported Cooperative Work in a Common Information Space: Case Processing in the Criminal Courts*. (Vol. Unpublished Dissertation). Irvine: University of California, Irvine.

Elliott, M. (2003). *The Virtual Organizational Culture of a Free Software Development Community.* Paper presented at the *3rd Workshop on Open Source Software*, Portland, Oregon.

Elliott, M., & Scacchi, W. (2002). Communicating and Mitigating Conflict in Open Source Software Development Projects, Working Paper: Institute for Software Research, University of California, Irvine, http://www.ics.uci.edu/~melliott/commossd.htm

Elliott, M., & Scacchi, W. (2003). Free Software: A Case Study of Software Development in a Virtual Organizational Culture. Working Paper: Institute for Software Research, University of Calfornia Irvine, http://www.ics.uci.edu/~wscacchi/Papers/New/Elliott-Scacchi-GNUe-study-DRAFT.pdf

Feller, J., & Fitzgerald, B. (2002). *Understanding Open Source Software Development*. N.Y.: Addison-Wesley.

Fielding, R. T. (1999). Shared Leadership in the Apache Project. *Communications of the ACM, 42*(4), 42-43.

K. Fogel (1999). *Supporting Open Source Development with CVS.* Scottsdale, AZ: Coriolis Press.

Easterbrook, S. M., Beck, E. E., Goodlet, J. S., Plowman, M., Sharples, M., & Wood, C. C. (1993). A Survey of Empirical Studies of Conflict. In S. M. Easterbrook (Ed.), *CSCW: Cooperation or Conflict?,* 1-68. London: Springer-Verlag.

Godwin, M. (1984). Nine Principles for Making Virtual Communities Work. *Wired,* 2.06, 72-73.

Gregory, K. (1983). Native-view Paradigms: Multiple Cultures and Culture Conflicts in Organizations. *Administrative Science Quarterly, 28*, 359-376.

James D. Herbsleb, Rebecca E. Grinter (1999). Splitting the Organization and Integrating the Code: Conway's Law Revisited. ICSE 1999: 85-95.

Hine, C. (2000). *Virtual Ethnography*. London: Sage.

Koch, S., & Schneider, G. (2000). *Results from Software Engineering Research into Open Source Development Projects Using Public Data*, Wirtschaftsuniversitat Wien.

Kollock, P. (1996). Design Principles for Online Communities, *The Internet and Society: Harvard Conference Proceedings,* http://sscnet.ucla.edu/soc/faculty/kollock/papers/design.htm.

Kollock, P., & Smith, M. (1996). Managing the Virtual Commons: Cooperation and Conflict in Computer Communities. In S. Herring (Ed.), *Computer-Mediated Communication: Linguistic, Social, and Cross-Cultural Perspectives*, 109-128, Amsterdam: John Benjamins.

Kollock, P., & Smith, M. A. (1999). Communities in Cyberspace. In M. A. Smith & P. Kollock (Eds.), *Communities in Cyberspace* (pp. 3-25). New York, NY: Routledge.

Mackenzie, A., Rouchy, P., & Rouncefield, M. (2002). *Rebel Code? The Open Source 'Code' of Work.* Paper presented at the Open Source Software Development Workshop, February 25-26, 2002, Newcastle-upon-Tyne, UK.

Martin, J. (2002). *Organizational Culture: Mapping the Terrain*. Thousand Oaks: Sage Publications.

Mockus, A., Fielding, R., & Herbsleb, J. (2002). Two Case Studies on Open Source Software Development: Apache and Mozilla. *ACM Trans. Software Engineering and Methodology,* 11(3), 309-346.

Mockus, A., Fielding, R. T., & Herbsleb, J. (2000). A Case Study of Open Source Software Development: The Apache Server. *Proc. 22nd Intern. Conf. on Software Engineering*, 263-272, Limerick, IR.

Nardi, B., Whittaker, S., Bradner, E. Interaction and Outeraction: Instant Messaging in Action. (2000). Proceedings CSCW 2000.

Noll, J., & Scacchi, W. (1999). Supporting Software Development in Virtual Enterprises. *Journal of Digital Information,* 1(4), http://jodi.ecs.soton.ac.uk/.

Olsson, S. (2000). *Ethnography and Internet: Differences in Doing Ethnography in Real and Virtual Environments.* Paper presented at the IRIS 23, Laboratorium for Interaction Technology, University of Trollhattan Uddevalla.

Ott, J. (1989). *The Organizational Culture Perspective*. Pacific Grove, CA: Brooks/Cole.

Pace, R. C. (1990). Personalized and Depersonalized Conflict in Small Group Discussions: An Examination of Differentiation. *Small Group Research, 21*(1), 79-96.

Pavlicek, R. G. (2000). *Embracing Insanity: Open Source Software Development*. Indianapolis, IN: SAMS Publishing.

Raymond, E. S. (2001). *The Cathedral & The Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. Sebastopol, CA: O'Reilly & Associates.

Robey, D., & Azevedo, A. (1994). Cultural Analysis of the Organizational Consequences of Information Technology. *Accounting, Management, and Information Technology,* 4(1), 23-37.

Sawyer, S. (2001). Effects of Intra-Group Conflict on Packaged Software Development Team Performance. *Information Systems Journal, 11*, (155-178).

Scacchi, W. (2002a). *Open EC/B: A Case Study in Electronic Commerce and Open Source Software Development* (Technical Report). Irvine, CA: University of California, Irvine.

Scacchi, W. (2002b). Understanding Requirements for Developing Open Source Software Systems. *IEE Proceedings - Software, 149*(2), 24-39.

Schein, E. (1990). Organizational Culture. *American Psychologist, 45*, 109-119.

Schein, E. H. (1991). The Role of the Founder in the Creation of Organizational Culture. In P. J. Frost, L. F. Moore, M. R. Louis, C. C. Lundberg, & J. Martin (Eds.), *Reframing Organizational Culture* (pp. 14-25). Newbury Park, CA: SAGE Publications, Inc.

Schein, E. H. (1992). *Organizational Culture and Leadership*. San Francisco: Jossy-Bass.

Smith, A. D. (1999). Problems of Conflict Management in Virtual Communities. In M. A. Smith & P. Kollock (Eds.), *Communities in Cyberspace* (pp. 134-163). New York, NY: Routledge.

Stallman, R. (1999a). *Free Software Foundation Brochure*. Cambridge, MA: Free Software Foundation.

Stallman, R. (1999b). The GNU Operating System and the Free Software Movement. In C. DiBona, S. Ockman, & M. Stone (Eds.), *Open Sources: Voices from the Open Source Revolution* (pp. 53-70). Sebastopol, CA: O'Reilly & Associates, Inc.

Strauss, A. L., & Corbin, J. (1990). *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*. Newbury Park, CA: Sage Publications.

Trice, H. M., & Beyer, J. M. (1993). *The Cultures of Work Organizations*. Englewood Cliffs, NJ: Prentice Hall.

Williams, S. (2002). *Free as in Freedom: Richard Stallman's Crusade for Free Software*. Sebastopol, CA: O'Reilly & Associates.

Yin, R. K. (1994). *Case Study Research, Design and Methods.* 2nd ed. Newbury Park: Sage Publications.