

The Virtual Organizational Culture of a Free Software Development Community

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

Abstract

Open source software development (OSSD) projects are growing at a rapid rate. Current researchers in OSSD processes have focused on the quantitative side of OSSD projects. Only a few researchers have gone beyond the quantitative approach to focus on the socio-technical perspective of OSSD. More studies are needed using a socio-technical perspective to develop empirically grounded understandings of the social circumstances involved in OSSD in virtual organizations. In particular, a deeper understanding of the work culture of the open source movement would be beneficial to the research community and to managers of open source projects. In this paper, I present findings from a virtual ethnography of a free software project showing how the cultural beliefs of the free software movement influence software development practices.

1. Introduction

Open source software development (OSSD) projects are growing at a rapid rate. The SourceForge Web site estimates 500,000+ users with 700 new ones joining every day and a total of 50,000+ projects with 60 new ones added each day. Thousands of OSSD projects have emerged within the past few years [3, 16] leading to the formation of globally dispersed virtual communities [11]. 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 [18,19]. Working together in a virtual community in non-collocated environments, 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, electronic

mailing lists, and more [19]. How do these developers concur over long distances in environments where people rarely if ever meet face to face in a virtual organization? How does the work culture of a virtual OSSD community influence software development processes? In this paper, I present empirical findings of a virtual ethnography showing the influence of a virtual organizational culture on OSSD in a free software project. The work culture of the free software development community ties the contributors together with an *esprit-de-corps* that perpetuates the community.

Current researchers in OSSD processes have focused on the quantitative side of OSSD projects, such as aspects of developer defect density, core team size, and other variables [15]. Only a few researchers have gone beyond the quantitative approach to focus on the socio-technical perspective of OSSD [6,7,8,13,19]. More studies are needed using a socio-technical perspective to develop empirically grounded understandings of the social circumstances involved in OSSD in virtual organizations. In particular, a deeper understanding of the work culture of the open source movement would be beneficial to the research community and to managers of open source projects.

Popular literature has described open source developers as members of a “geek” culture [16] notorious for nerdy, technically savvy, yet socially inept people, and as participants in a “gift” culture [17] where social status is measured by what you give away. However, no empirical research has been conducted to study open software developers as virtual organizational cultures [14,21] with beliefs and values that influence their work. Researchers have theorized the application of a cultural perspective to understand IT implementation and use [1] but few have applied this to the workplace itself [4,5]. In this paper, I present findings from a free software project whose purpose is to build a free enterprise resource planning (ERP) system. I show how the work culture of

this community influences software development decisions such as tool choice. This study is part of an ongoing comparative study of various types of open software communities [6,18,19] including both free software projects and open source projects. A portion of the results are presented here. For a more detailed account of the study and its results, see [7].

2. Free versus open source software

Free software developers work in globally disbursed virtual communities with few face-to-face meetings, utilize informal requirements gathering [19], and practice software development techniques that veer from typical software development practices [12]. It is important to distinguish between the terms free software [22] and open source [3] distributing software. Free software refers to software that is open to anyone to copy, study, modify, and redistribute [22]. The Free Software Foundation (FSF), founded by Richard M. Stallman (known as RMS in open source communities) [22] in the 1970s, advocates the use of its GNU General Public License (GPL) as a copyright license which creates and promotes freedom. A popular term heard in the free software community is “Think free speech, not free beer”. It is used to emphasize the importance of the defense of freedom, not just the ideal of promoting software that is free of cost.

The term open source was coined by a group of people concerned that the term “free software” was anathema to businesses. This resulted in the formation of the Open Source Initiative (OSI) (<http://www.opensource.org>), a non-profit corporation dedicated to managing and promoting the Open Source Definition for the good of the community. The major difference between free software movement and the OSI is in the licensing requirements. The OSI promotes more liberties with open source licensing than the FSF.

3. GNU Enterprise

The research site is the free software development community known as *GNU Enterprise*, or more simply, GNUe (<http://www.gnuenterprise.org>). GNUe is a meta-project of the GNU (<http://www.gnu.org>) Project, GNUe is an international virtual organization for software development [2] based in the U.S. and Europe that is developing a free, open source ERP systems and related E-Business capabilities. The plans are for GNUe to consist of customized shared applications; a set of tools to implement a full ERP system; and a general community of support and resources for developers using GNUe tools.

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. Companies from Austria, Argentina, Lithuania, and New Zealand support paid contributors but most of the contributors are working as non-paid participants.

4. Data

I present the results of a qualitative study of the GNUe community and its work culture conducted using grounded theory [9]. The purpose of the study is to research what processes are used for OSSD and what socio-technical structures evolve to support free/open software development work practices. Data gathering has entailed a virtual ethnography over the past year. Data collection includes the content analysis of IRC archives; kernel cousins archives (summary digests of IRC and mailing list archives); IRC archives; mailing list archives; email interviews; Web site documents and observations; and personal interviews taken place at two open source conferences. Two cases from IRC and mailing list archives of the GNUe virtual community at work are analyzed:

1. A newcomer (outsider) who criticizes the choice of a non-free graphics tool to create a Web site screenshot and causes a debate over tool choice. This debate takes place for one day on the IRC with the graphics creator arguing against the exclusive use of free tools for GNUe software development. Finally, fellow contributors help find a free graphics tool which the creator agrees to use to replace the offending non-free graphic.
2. A group of insiders (frequent contributors) debate the issue of using non-free software to develop documentation. A frequent contributor argues for the exclusive use of free tools for documentation creation. The debate lasts three days without a change to documentation. The insider making the complaint agrees to drop the issue.

5. Organizational Culture and GNU Enterprise

Much like societal cultures have beliefs and values manifested in norms that form behavioral expectations, organizations – both typical and virtual – have cultures that form and give members guidelines for “the way to do things around here.” An organizational culture perspective [4,20] provides a method of studying an

organization's social processes often missing in a quantitative study of organizational variables.

Similar to business organizations with a founder who leads the organization's culture [20], the free software movement, with RMS as its founder, has inspired the creation of virtual organizations with beliefs and values of the free software movement.

In this study, organizational culture is viewed as an emergent phenomenon manifested in an organization's work practices, norms, artifacts, and symbols. Culture is treated as a metaphor for organization, not as a discrete variable within an organization. The virtual community *is* the organizational culture. The GNUe virtual organization is presented as a subculture of the FSF inculcating the beliefs and values of the free software movement into their everyday work practices in free software development of GNUe. An Integration perspective [14] is used to show how the beliefs and values of the free software movement bind contributors together in the spirit of developing a free software project.

A cultural analysis includes the identification of content themes, common threads of concern that are reflected in interpretations of several cultural manifestations [14]. Content themes can include beliefs or tacit assumptions as well as attitudes or values. They may be espoused (e.g. belief in free software) or inferred by a researcher (e.g. value in community and cooperative work). Cultural manifestations of a culture include rituals, stories, humor, jargon, physical arrangements, and formal structures and policies, as well as informal norms and practices [14]. For an interpretation of a virtual culture, beliefs and values are analyzed in terms of the following manifestations: electronic artifacts, formal policies, and informal norms and work practices. Table I shows a summary of the beliefs and values used in the analysis. For this paper, I elaborate on the belief in freedom as a content theme and show its manifestation in the choice of tools for GNUe software development. For a detailed analysis of all GNUe cultural beliefs and values, see [7].

6. Belief in Freedom

Belief in freedom is a recurring theme throughout the data. In fact, the free software movement is claimed by RMS to be inspired from the ideals of 1776: freedom, community, and voluntary cooperation (<http://www.gnu.org>). In case two, during a lengthy heated debate regarding the use of free versus non-free software to develop documentation, one GNUe developer claimed that he is working on GNUe for the "freedom of his son". The GPL was designed by RMS "to uphold and defend the freedoms that define free software – to use the

words of 1776, it establishes them as inalienable rights for programs released under the GPL. It ensures that you have the freedom to study, change, and redistribute the program, by saying that nobody is authorized to take these freedoms away from you by redistributing the program under a restrictive license (<http://www.gnu.org>)".

This deep belief in freedom that drives the FSF instantiates itself in the belief in using free software and the belief in the freedom of choice of job assignments.

6.1 Belief in free software

The belief in free software is a core motivator of free software developers. GNUe developers show a strong belief in free software extolling its virtues on its Web site and in daily activity on the IRC logs. This belief is manifested in electronic artifacts such as the Web pages, source code, software design diagrams, and accompanying articles. The GNUe Web site advertises that it is "a free software project with a corps of volunteer developers around the world working on GNUe projects". The GNUe software is licensed under the GNU General Public License (GPL) (<http://www.gnu.org/copyleft/gpl.html>). The belief in free software is manifested formally, through the rights and imperatives afforded in the GPL that one realizes if employing free software, and informally in the moral imperatives that contextualize the software development work practices of GNUe contributors. Throughout the digests and IRC logs, there are numerous references to the importance of adhering to the principles of free software. In the following excerpt from a GNUe IRC log, the belief in free software prompts an outsider to inquire as to why a screen image was made with a non-free piece of software. The informal practice of readily accepting outsiders to the GNUe processes facilitates these types of reviews.

An outsider (JimB who happens to be a member of the European free software foundation) questions the use of a non-free Adobe software to create a graphic and calls it "quite shocking". Surprisingly, core contributors discuss it with JimB and eventually, after a day-long IRC debate, a solution is found to recreate the graphic with free software (Names are camouflaged to protect participants' identity).

<JimB> Hello. Several images on the GNUe website seems to be made with non-free Adobe softwares, I hope I'm wrong: it is quite shocking. Does anybody now more on the subject ?

<JimB> We should avoid using non-free software at all cost, am I wrong ? Anyone awake in here ?

In the second case, Bill, a frequent contributor (insider), brings up the subject of creating documentation using non-free tools. His problem is that in order to use the required free package, lyx, to read and edit the original documentation, he would have to also install some non-free graphics tools. He brings it up here and then for the next three days argues vociferously for the use of docbook (a non-free tool). He is outnumbered and finally, gives up the argument.

<Bill> lyx requires non-free software...should that be acceptable for a GNU project?...

<John> Bill: 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). pick one :)

<Bill> use docbook then

7. Conclusion

Conclusions from this study indicate that the recording and archiving of GNUe IRCs contribute to the mitigation and resolution of conflict while, at the same time, contribute to the persistence and renewal of cultural beliefs and values. The following quote illustrates the importance of the IRC to the GNUe culture:

"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." (GNUe core contributor, email interview (2002))

Both cases show the power of the **belief in freedom (free software, free choice)** that deeply influences contributors' decisions. While a slight delay in work occurs while contributors debate issues on IRC, the debates result in **reinforcement of cultural beliefs in free software**. The beliefs in free software and freedom of choice serve as motivations to join a free software project and, at the same time, contribute to the building of the community that perpetuates the GNUe software project.

This research shows that free software or OSSD projects develop rich organizational cultures in their virtual communities. To preserve the presence of online communities like GNUe, several structural features are important [10] ongoing interaction, identity persistence, and knowledge of previous interactions. For the perpetuation of a free software development community, I

propose that persistence of rich cultural beliefs and values in the work itself is also needed.

Acknowledgements

The research described in this report is supported by grants from the National Science Foundation #IIS-0083075 and #ITR-0205679. No endorsement implied. Mark Ackerman at the University of Michigan Ann Arbor, Les Gasser at the University of Illinois, Urbana-Champaign, Walt Scacchi, 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 described in this paper.

8. References

1. Avison, D. E. and M. D. Myers (1995). "Information systems and anthropology: an anthropological perspective on IT and organizational culture." Information Technology and People, 8: 43-56.
2. Crowston, K. and B. Scozzi, (2002). "Exploring Strengths and Limits of Open Source Software Engineering Processes: A Research Agenda". 2nd Workshop on Open Source Software Engineering, May 25, 2002, Orlando, Florida.
3. DiBona, C., S. Ockman, et al. (1999). Open Sources: Voices from the Open Source Revolution. Sebastopol, CA, O'Reilly & Associates Inc.
4. Dube, L. and D. Robey (1999). "Software Stories: Three Cultural Perspectives on the Organizational Practices of Software Development." Accounting, Management and Information Technologies, 9(4): 223-259.
5. Elliott, M. (2000). "Organizational Culture and Computer-Supported Cooperative Work in a Common Information Space: Case Processing in the Criminal Courts." Ph.D. Dissertation, Information and Computer Science Dept., University of California, Irvine.
6. Elliott, M. and W. Scacchi (2002). "Communicating and Mitigating Conflict in Open Source Software Development Projects." <http://www.ics.uci.edu/~melliott/commossd.htm> <http://www.ics.uci.edu/~melliott/commossd.htm>, Institute for Software Research, University of California, Irvine
7. Elliott, M. and W. Scacchi (2003). "Free Software: A Case Study of Software Development in a Virtual Organizational Culture." <http://www.ics.uci.edu/~melliott/reports/Elliott-Scacchi-FinalRep-04.htm>. Institute for Software Research, University of California, Irvine.

8. Feller, J. and B. Fitzgerald (2002). Understanding Open Source Software Development. New York, Addison-Wesley.
9. Glaser, B. and A. L. Strauss (1967). The Discovery of Grounded Theory: Strategies for Qualitative Research. New York, Aldine.
10. Kollock, P. (1996). "Design Principles for Online Communities". The Internet and Society: Harvard Conference Proceedings, also available at <http://sscnet.ucla.edu/soc/faculty/kollock/papers/design.htm>.
11. Kollock, P. and M. A. Smith (1999). Communities in Cyberspace. In M. A. Smith and P. Kollock (eds.), Communities in Cyberspace. New York, NY, Routledge: 3-25.
12. Kotoyna, G. and I. Sommerville (1998). Requirements Engineering: Processes and Techniques. New York, NY, John Wiley and Sons, Inc.
13. Mackenzie, A., P. Rouchy, et al. (2002). "Rebel Code? The Open Source 'Code' of Work." Open Source Software Development Workshop, Newcastle-upon-Tyne, UK.
14. Martin, J. (2002). Organizational Culture: Mapping the Terrain, Thousand Oaks: Sage Publications.
15. Mockus, A., R. T. Fielding, and J. Herbsleb. (2002). Two case studies of Open Source software development: Apache and Mozilla. ACM Transactions on Software Engineering and Methodology, 11(3):309-346.
16. Pavlicek, R. G. (2000). Embracing Insanity: Open Source Software Development. Indianapolis, IN, SAMS Publishing.
17. Raymond, E. S. (2001). The Cathedral & The Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary. Sebastopol, CA, O'Reilly & Associates.
18. Scacchi, W. (2002a). "Understanding the Social, Technological, and Policy Implications of Open Source Software Development." NSF Workshop on Open Source Software, February 2002, <http://www.ics.uci.edu/~wscacchi/Papers/New/OSS-Policy.pdf>
19. Scacchi, W. (2002b). "Understanding Requirements for Developing Open Source Software Systems", IEE Proceedings – Software, 149:24-39.
20. 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 and J. Martin (eds.), Reframing Organizational Culture. Newbury Park, CA, SAGE Publications, Inc.: 14-25.
21. Schein, E. H. (1992). Organizational Culture and Leadership. San Francisco, Jossey-Bass.
22. Stallman, R. (1999). The GNU Operating System and the Free Software Movement. In C. DiBona, S. Ockman and M. Stone (eds.), Open Sources: Voices from the Open Source Revolution. Sebastopol, CA, O'Reilly & Associates, Inc.: 53-70.
23. Williams, S. (2002). Free as in Freedom: Richard Stallman's Crusade for Free Software. Sebastopol, CA, O'Reilly & Associates.

Table I. GNUe Virtual Organizational Culture

Content Themes	Informal Practices and Norms	Formal Practices	Electronic Artifacts
Belief in Free Software	Open disclosure; Impromptu design/code reviews; Exclusive use of free software; IRC discussions	GPL; Scheduled IRC meetings; CVS for Code mods and releases;	Free source code and documentation; website
Belief in Freedom of Choice	Informal management; Debates over choice of free versus non-free OSSD tools;	No formal timeline; Double Choco-latte (in-house project management system)	Use of free and non-free (i.e. Windows) for software development
Value in Community	Ideal of community-building; Immediate acceptance of outsider contributions		Archived IRC logs; kernel cousins (digests of IRC and mailing lists)
Value in Cooperative Work	Mitigation and resolution of conflict over IRC		