Levels of Formality in FOSS Communities

Andrew Schofield[1], and Professor Grahame S. Cooper[2]

1   Salford Business School, University of Salford, Salford, M5 4WT, UK,
a.j.schofield@pgt.salford.ac.uk,
WWW Home page: http://www.postgrad.isipartnership.net/~aschofield/
2   School of Computing, Science, and Engineering, University of Salford,
Salford, M5 4WT, UK, g.s.cooper@salford.ac.uk,
WWW Home page:
http://www.cse.salford.ac.uk/profiles/profile.php?profile=G.S.Cooper

Abstract. One of the aspects of Free and Open Source Software (FOSS) which may act as a significant deterrent to its adoption, is the method used to collaboratively develop the software and provide support through the use of communities. It is not until this method is examined more closely that its many advantages can be realised. The method can, however, seem very disorganised especially when compared with traditional proprietary development styles. A key difference between these two development approaches lies in the management of projects, and perhaps as a consequence, in the level of formality in the community environment. This paper presents the results of empirical survey research investigating FOSS community participants' views on the level of formality in FOSS, and the way in which this affects both development and support provision activities. The paper then concludes by analysing what can be learnt from the participant's views.

1   Introduction

Despite the many success stories and research studies demonstrating the advantages and capabilities of FOSS, the stereotypical view of ideas and code being thrown around within disorganised communities, still has the potential to deter individuals and organisations from using or developing it. In actual fact, these communities are arguably the most important element of FOSS. Much research has been done on FOSS communities [1-4, 6-8, 10] demonstrating that there is a general framework that communities follow [8, 9]. However, differences in working methods and styles of approaches become very apparent when comparing FOSS communities. These are also seen as major differences between FOSS and proprietary closed-source development [5, 7].

This paper contributes to the knowledge in this area by presenting the results of empirical survey based research, which collected FOSS community participants' experiences and views on the level of formality within FOSS communities, and its

subsequent effect. As these communities are often quite complex, the research was split into two sections, one focusing on the support aspects of the communities, and the other focusing on development activities.

2   Research Method

As its primary data collection tool, the research used an on-line survey consisting of open questions designed to collect qualitative data. The sample set for the research consisted of Linux, BSD, and Open Source user and/or interest groups, hereafter referred to as Linux User Groups or LUGs. The research targeted LUGs within the UK, US, Italy, Germany and Canada, as these were the countries with the highest number of LUGs. In total 392 responses were received from the various countries, 48% of which were from developers. It must be pointed out that LUGs cannot be considered as absolutely representative of FOSS communities. However, the survey was particularly designed to collect members' perspectives of FOSS communities in general, and LUG members are also likely to participate in other communities as well. Additionally it should be noted that the sample consists of FOSS community members, and although some may have proprietary software experience, it should be recognized that a bias towards FOSS software will exist. The paper's purpose, however, is not to specifically compare the formality of FOSS and proprietary approaches, but rather to examine the effect of formality on FOSS communities.

3   Research Findings

3.1   FOSS Community Formality

The first section of the survey dealt with the general concept of formality in FOSS communities. Participants were asked their opinions about the level of formality in the working practices of FOSS communities, when compared to the proprietary software approach. The responses, grouped by topic, are summarised below:

- **Project/Community Dependant**: A significant number of respondents wrote that the formality of a project or a community is very specific, and that a generalised statement that describes all of them is not possible.
- **A Mixture of Elements**: Respondents pointed out that communities can be viewed as informal in terms of them being open for anyone to participate, however, in terms of the management of the final product, FOSS communities could be seen as very formal. Others pointed out that how formal a community is depends on how it is led, and that the formality of the methods used in a community project are needed only for project management purposes

- **Informal Interaction**: Many respondents wrote that the interactivity side of FOSS communities is fairly informal (i.e. discussion forums, etc). Others added that formality depends on how well one knows the project leader, and presumably the other community members.
- **Depends on the Projects' Complexity**: Many respondents stated that the formality of a community depends on the project's size, scope, complexity, and maturity. Consequently, a large project requires a high degree of formality to keeps things under control. However, for each of these large projects, there are many smaller sub-projects that are far less formal.
- **Theory Versus Practice**: Many respondents wrote that in both the case of FOSS and proprietary software, formal rules and guidelines may be set down, but are seldom followed. Additionally, community based and company based software development often operate with the same rules, but it is how these rules are enforced that differentiates the two types.
- **The Freedom of FOSS**: Some respondents stated that FOSS is more informal because of its underlying ethos. They stated that those involved in FOSS do not want to be restricted to a formal and controlled system.

3.2   Effects of Formality on FOSS Support Activities

Survey participants were then asked to comment on how the level of formality in FOSS communities affects support provision.

- **A Deterrent to the use of FOSS**: Respondents posted that the unwritten rules and etiquette used in FOSS forums could make people feel unwelcome. Many also felt that frequent arrogance among knowledgeable community participants leads to 'newbies' being unwelcome.
- **Project Dependant**: Some wrote that the effect would depend on the size of the project. Large projects having good support because of the number of people involved, and smaller projects having poor support as the fewer members will have less time free to provide support.
- **Formality Improves Support**: Several respondents observed that communities with strict and formal working practices have very good support, particularly documentation. Likewise, those less formal communities were often found to be lacking in support.
- **Ease of Using Forums**: Other respondents felt that the informal nature of FOSS community forums facilitates interaction. Contrary to the comments left above, it was suggested that the equality and lack of a hierarchy, would make members willing to help one another. The importance of interplay between experts and 'newbies' was also emphasised as a positive factor.
- **Direct Contact with Programmer(s)**: Several participants wrote that FOSS communities allow direct communication to the actual writer of the code. This clearly has advantages as no-one could provide better support, and the questions asked might also stimulate ideas on further code development.

- **Enjoyment of Support Forums**: Comments were left stating that the informal nature of support forums is enjoyable, but that anyone requesting support in an inappropriate manner is likely to receive an unhelpful reply.

3.3   Effects of Formality on FOSS Development Activities

Finally, respondents were asked to comment on how the level of formality affects software development within FOSS communities.

- **The Big Picture**: Without some control or steering group, respondents believed that programmers would do what they personally think is best, which may not be what is best for the overall project.
- **Very Informal Management**: Several respondents stated that projects with an extremely relaxed approach lead to either unusable products or a forking of the projects, leading to several very similar products.
- **Natural Formality**: Some respondents wrote of projects adapting and finding their own level of formality. One respondent referred to this as a community's *"natural formality"*.
- **Development Feedback**: Several survey respondents pointed out that the detection of bugs, and even design ideas can originate from questions asked on a support forum, and how this is supported by an informal environment.
- **Higher Formality for Larger Projects**: Many respondents stated that they felt larger projects required a more formal structure to manage all the code submissions, while others felt that this was not a problem with the use of versioning software.
- **Openness in Development**: It was pointed out that the open approach can lead to arguments and disagreements, especially in very informal and undisciplined projects.
- **Informal Development Rules**: Although most respondents felt that at least some formal structure was required, some felt that the lack of rules was a good thing, allowing FOSS development to release leading edge software quicker. This was also facilitated by the unrestricted communication and the lack of red tape.
- **Informality Breeds Innovation**: Many respondents thought that the freedom of an informal environment helps developers to be dynamic, innovative, and have the freedom to experiment, and that a formal environment stifles creativity.
- **More control needed**: Although it was a minority view, some felt more control or planning was needed. They felt there was too much discussion about software functionality and project direction, and that FOSS communities often have problems related to clashing egos and methods.
- **Deterrents to Involvements**: A disadvantage pointed out was that informal environments could deter those who have previously worked in a more structured environment. Others however, felt that it would encourage participation

- **Communication Leads to Results**: Several wrote that FOSS communities facilitate communication and that the informal approach makes it easier for developers to work together.
- **Formal Practices at the Right Time**: Some respondents felt that FOSS development should adopt the most suitable level of formality for the phase of the development. As a project grows, and more become involved, a *"benevolent dictator"* is needed. Some referred to this as a Darwinist approach, because only the *'fittest'* submissions are accepted.
- **Disadvantages of Voluntary Work**: Some survey respondents felt that more effort is put into developing software that is fun to write, rather than the more mundane or boring applications. To quote one respondent *"that's why we only have 3 office suites but about 42,000 music players"*.
- **Informality is Good for Growth**: A few respondents wrote that although informal approaches are good for recruiting new members, they are often then less motivated to work than in a formal environment.

4   Conclusions

The results allow us to further define the concept of formality with regards to FOSS. We can first separate formality into more specific important factors, which we shall call 'managerial formality' and 'cultural formality'. Managerial formality refers to aspects of formality which are for the purposes of organisation and structure. These manifest themselves as rules and regulations concerned with the support and particularly the development of FOSS. Many respondents wrote of the importance of formal management, particularly for large projects with many people involved and during phases of the development cycle where decisions about the direction of the development are made. Cultural formality refers to the level of formality which exists between the participants of the community. This is evident from the discussion boards and mailing lists of a community and is defined by the members themselves, with a possible influence from the managerial formality. The essential difference is that the former is imposed, or perhaps suggested, by the community leaders, while the latter comes into being or develops from the personalities and actions of the participants.

From the analysis of the responses, it seems that the predominant view is that managerial formality improves both support and development activities. Nevertheless, many survey participants warned of the dangers of an environment which was too formal. Likewise, cultural formality promotes freedom and innovation, but can be off-putting to 'newbies' or those used to formal practices.

5   References

1 Ghosh, R., A., Glott, R., Krieger, B., Robles, G. (2002), "Survey of Developers", Free/Libre and Open Source Software: Study and Survey, International Institute of Infonomics, University of Maastricht, The Netherlands, Available at: http://www.infonomics.nl/FLOSS/report/ , (Accessed May 2006)

2 Hann, I. H., (2004) "Why Developers Participate in Open Source Software Projects: And Empirical Study", Twenty-Fifth International Conference on Information Systems.

3 Hertel, G., Niedner, S., Herrmann, S. (2003), "Motivation of Software Developers in Open Source Projects: An Internet-based Survey of Contributors to the Linux Kernel", Research Policy, Special Issue on Open Source Software Development, Available at: http://opensource.mit.edu/papers/hertel.pdf ,(Accessed February 2004)

4 Lakhani, K. R., Wolf, R.G. (2003), "Why Hackers Do What They Do: Understanding Motivation Effort in Free Open Source Software Projects", MIT Sloan School of Management Working paper, Available at: http://freesoftware.mit.edu/papers/lakhaniwolf.pdf, Accessed (February 2004)

5 Moody, G. (2001), "Rebel Code: How Linus Torvalds, Linux and the Open Source Movement Are Outmastering Microsoft", The Penguin Press, England

6 Oh, W., Jeon, S., (2004) "Membership Dynamics and Network Stability in the Open-Source Community: The Ising Perspective" Twenty-Fifth International Conference on Information Systems.

7 Pavlicek, R. C. (2000), "Embracing Insanity: Open Source Software Development", Sams Publishing, USA

8 Scacchi, W., Feller, J., Fitzgerald, B., Hissam, S., Lakhani, K., (2005) "Understanding Free/Open Source Software Development Processes", Available at: (http://www.ics.uci.edu/~wscacchi/Papers/New/SPIP-FOSS-Intro-Dec2005.pdf) (Accessed December 2005)

9 Schofield, A., Mitra, A. (2005), "Free and Open Source Software Communities as a Support Mechanism", UK Academy of Information Systems conference 2005, Newcastle, UK

10 Zhang, W. & Storck, J, (2001) "Peripheral Members in Online Communities", Americas Conference on Information Systems, Boston, USA