2005

Cohesion, Structure and Software Complexity: A Model of Open Source Software Development

Sherae L. Daniel
University of Maryland - College Park, sdaniel@rhsmith.umd.edu

Katherine J. Stewart
University of Maryland - College Park, kstewart@rhsmith.umd.edu

Cohesion, Structure and Software Complexity: A Model of Open Source Software Development

Sherae L. Daniel
University of Maryland, College Park
sdaniel@rhsmith.umd.edu

Katherine J. Stewart
University of Maryland, College Park
kstewart@rhsmith.umd.edu

ABSTRACT

As the use of open source software gains popularity, it is important to understand the factors that contribute to the success of open source software development projects. This research contributes to this understanding by developing a set of propositions about the organizational structure that supports open source software projects. We argue that in open source software development it is important to understand the existence and interaction of two related but distinct entities; the interest community and the software development group. We propose relationships between the development group, the interest community and software complexity in open source software development. Implications of the propositions for research and practice are discussed.

KEYWORDS

open source; organizational structure; group cohesion; software development

INTRODUCTION

Open source software (OSS) is software for which the source code is open for anyone to see, modify and use[1]. OSS is often developed by volunteers and the software is frequently made available without a licensing fee. The factors that lead to success for OSS development are of interest to both academics and practitioners. Corporate and government agencies are intrigued by the potential to gain software at lower costs. Several companies, including Oracle and IBM have begun to invest in OSS. Several governments, including Brazil and China, are endorsing open source operating systems[2]. In addition to practitioner interest, one of the many reasons OSS is interesting academically is because OSS developers are able to overcome many challenges. Two of the most frequently acknowledged challenges are the lack of monetary incentives for OSS developers and the geographical dispersion of developers (Raymond 2001). Given these challenges, research has sought to understand what leads volunteers to contribute to open source software development and what leads to the effective coordination of those contributions (Gallivan 2001; Hars and Ou 2002).

Research into the factors that lead to success for OSS development has been conducted at multiple levels of analysis. These include the individual developer, the development group and the open source community. At the level of the individual developer, research has considered motivations including indirect economic (Feller and Fitzgerald 2002; Lerner and Tirole 2002), social (Raymond 2001) and technological motivations (Lakhani and Hippel 2003; Raymond 2001). At the open source community level, it has been proposed that a community ideology enables the development of OSS (Hars and Ou 2002; Markus, Manville and Agres 2000; Raymond 2001). Research at the group level has considered the impact of trust, communication quality, adherence to the open source ideology (Stewart and Gosain 2001) and control structures (Gallivan 2001) on open source success.

This paper focuses on the project level, which is differentiated by the software that is produced by each OSS development group. Although other levels of analysis are important, from the user perspective, focusing on the software may be of greatest interest because this is the level that has the most direct impact on software users. Some characteristics of the software that may impact users are the complexity of the software and the potential for network externalities. Potential network externalities include the degree of compatibility of the software with a variety of other software, the level of adoption that the software experiences and the availability of support for the software.

---

[1] See <http://www.opensource.org > *(*accessed February 23, 2005*)*

[2] See <http://news.com.com/2100-1001-272299.html?legacy=cnet > (accessed February 23, 2005)

This paper argues that it is useful to conceptualize the organizational structure that supports a single OSS project as consisting of a development group and an interest community. By taking advantage of an active community of interested users who have access to the source code, successful open source development groups are able to compensate for the lack of an authority structure that provides deadlines and requirements. We suggest that effective use of this organizational structure influences the successfulness of open source projects.

The contribution of this work to the growing body of research on OSS projects is in its explicit attention to organizational structure as a key driver of project success. Understanding the organizational structure is essential to provide insight into the internal interactions of the open source project that contribute to success. This work draws on the social psychological and software development literatures to identify characteristics of the development group and interest community that are expected to lead to improved performance for an OSS project.

The rest of the paper will proceed as follows. In section two we discuss the entities that make up the proposed organizational structure that supports an OSS development project. Section three presents a model this organizational structure supports and section four explores the implications of this organizational structure for research and practice.

COMPONENTS OF THE OSS PROJECT

Prior research has suggested several ways to categorize the various participants in OSS projects. Ye and Kishida, from their study of the GIMP project, describe eight types of participants in OSS projects including the project leader, the core members, the active developers, the peripheral developers, bug fixers, bug reporters, readers and the passive users (Ye and Kishida 2003). Similarly, Krogh et. al. classify individuals related to an OSS project as list participant, joiner, newcomer and developer (Krogh, Spaeth and Lakhani 2003). There have also been several proposed schemes to understand the different types of developers. Some schemes are based on developer motivation (Hann, Roberts and Slaughter 2004) while others are based on experience and expertise on the project (Edwards 2001). In this research we generalize from prior research and consider OSS project participants to either be a part of the development group or the interest community.

The advantage of studying these two entities, the development group and the interest community, is that doing so provides a parsimonious schema that may be applied to the majority of OSS projects regardless of their size or history. More fine grained schemes developed from exploration of individual OSS projects, while they provide more detail about these projects, may not be appropriate for many of the smaller OSS projects. Many OSS projects do not have enough members to fulfill the multiple roles that more detailed membership categorization schemas describe. We believe the proposed two level structure is an appropriate alternative when attempting to understand the factors that lead to success for a diverse set of OSS projects.

It is important to consider these two entities distinctly even though the boundaries between them are somewhat permeable. The members of the development group and the interest community can participate in some of the same functions including documentation, bug reporting and bug fixing. Participants can be members of either or both entities and they may change their membership status with either entity at any time. However, even though there is some ease in moving between entities, many users never become developers (Von Hipple and Von Krogh 2003). Developers and members of the interest community generally have different skills and purposes for interacting with the OSS project. Further, the skill set required to become an active developer is not easily attained and the development group is often a much smaller set of participants compared to those in the interest community. Because of the different size of the entities and the different skill set and motivations of their members, the development group and the interest community are expected to impact the OSS project in different ways.

The Development Group

The OSS development group members for one project are identified by their authority to add written code to the primary distribution of the project releases (i.e. CVS write access) and a pattern of regular written code contributions to the development of the source code. The individuals in the development group often use computer mediated forms of communication. The OSS developers are typically volunteers, but are sometimes paid. OSS development groups have been successful in developing large software projects such as Linux and the Apache Server. In many ways the OSS development group is very similar to other computer mediated software development groups. However, the differences from other development groups are important to understanding the antecedents of success for OSS projects.

OSS development groups are different from other software development groups because their members often expend effort and develop software without immediate financial incentives. This lack of financial incentives may exacerbate difficulties in control and coordination mechanisms for OSS development groups (Gallivan 2001). Without control mechanisms, development tasks may be left undone because not all programming tasks are of the same level of interest to developers

(Weinberg 1998). This can be problematic because the lack of interest by developers does not necessarily overlap with a lack of importance to users.

Another difference in OSS development groups is that they often do not have an organizational sponsor to provide a sense of direction via documented deadlines or requirements defining the expected outcome of the project (Scacchi 2002). This lack of focus may lead to challenges for development and challenges for sustained adoption by users. Many large software development projects require cooperative, continued efforts of multiple developers toward a common goal. If projects change directions frequently it may be difficult to acquire and maintain the level of cooperation and commitment necessary to develop large software projects. Similarly, without a specified goal or set of requirements, potential users may find it hard to identify OSS projects as useful.

Possibly the most significant difference between OSS development groups and other development groups is the interaction the OSS development group has with an associated interest community. Although many software applications, including proprietary software, have associated interest communities, the interest communities associated with OSS development projects are different because of the interest community's access to the source code and their potential direct impact on the direction of the project. Because of the interest community's access to the source code, the interest community can not only provide user testing and feature requests; it can also find and correct errors in the source code. The open source literature proposes that OSS is "better" because of the potential for many people to examine the source code (Raymond 2001). The interest community also may be able to help by offering suggestions for features, which helps to ensure that the software product developed meets the needs of a large consumer base. The impact of user requests is more direct in an OSS environment because the requests do not have to go through company management approval. This paper proposes that understanding how characteristics of the associated interest community interact with the characteristics of the software development group may be an important unexplored relationship in the study of OSS development.

The Interest Community

We define the open source interest community for a given project as the set of individuals who have an active interest in the software developed by the project and participate in some computer mediated communication related to the project. Individuals interested in the project can be identified through participation on list serves, public forums or other similar computer mediated communication vehicles. The interest community often consists of potential and current users of the software under development and developers and users of related software. Examples of related software are competing software products that provide similar services or software that works in conjunction with the software that the project is developing. The interest community members often exchange information about acquiring the software developed by the project and using the software developed by the project. Sometimes members of the interest community submit bug reports, request additional features or submit corrections for bugs (Raymond 2001).

Identification of the interest community associated with a particular project is an important factor in understanding the success of an OSS project. There are many characteristics of an interest community that may impact the OSS development process, including the number of participants in the community, the number of lurkers in the community and the level of participation across participants in the community. Understanding the behavior of this interest community is important because the function that the interest community fills can be very important in the software development process.

The literatures on open source and proprietary software development provide evidence of the importance of code reviewers other than core developers in the software development process. Raymond suggests that the large number of reviewers of the source code is an important way OSS development is able to find and correct errors and ultimately develop better code (Raymond 2001). Participation in proprietary software development by users of the software under development has been found to be an important component in software development (Barki and Hartwick 1994; Cushing 1990; Hunton and Beeler 1997). This further supports the proposal that identifying the community of individuals who provide this function in OSS projects will be an important factor in determining the success of the OSS development process.

MODEL DEVELOPMENT

Software Complexity

There are many ways of defining success in the OSS context. These include the level of adoption of the software, the popularity of the software, the complexity of the software, and the amount of active participation in the software development project. The key outcome of interest here is software complexity, because the complexity of the software is expected to be an antecedent of the other success measures.

Low levels of complexity leads to minimal errors and ease of comprehension and maintenance (Curtis, Sheppard, Milliman, Borst and Love 1979). Because it will be easier for open source developers to understand and then contribute to the projects with lower complexity, projects with lower complexity may have more developer participation. Added developer participation may increase functionality and reduce the number of errors (Raymond 2001). Additional functionality and fewer errors should make the software more appealing to users. Ease of comprehension will allow many people to understand how to support the software and fix errors in the software and give the users many options from which to get help using and maintaining the software. These factors combine to show how software complexity is expected to lead to developer participation, popularity of the software and user adoption.

Group Cohesion

The group characteristics examined here as a driver of the complexity of software within a particular OSS project are the levels of group cohesion in the development group and the interest community. Shaw describes group cohesion as "the degree to which members of the group are attracted to each other (Shaw 1981)." Group cohesion has also been described as a psychological force that binds people together (Keyton and Springston 1990).

Members of highly cohesive groups identify with the group (Shaw 1981). In identifying with the group they also identify with the successes of the group. Because a low level of complexity in a piece of software is a mark of success for the group and may lead to other forms of success such as popularity of the software, cohesive open source software development groups may work harder to attain software with low levels of complexity. This added focused effort by the developers is expected to lead to software with low levels of complexity.

Group cohesion has been found to lead to increased participation (Evans and Dion 1991; Yoo and Alavi 2001). In this context more participation could lead to more design choices from which to choose the best design solution. Because group cohesion leads to more solutions from which to choose, highly cohesive groups are expected to have a better chance of choosing the best design solution. Software with low complexity is a good solution because it is easier to understand and is therefore easier to maintain.

P1: The cohesion of an OSS development group leads to the development of lower complexity software.

The interest community faces challenges to becoming an effective part of the OSS development process that are similar to those faced by the development group. The interest community must sustain participation in order to be effective in the open source development process. Group cohesion has been found to lead to increased participation and commitment to stay in a group (Evans and Dion 1991; Yoo and Alavi 2001). We expect the cohesion of the interest community to lead to participation in the interest community. The interest community cannot generally directly impact the software complexity for the project(an exception is when a member of the interest community occasionally fixes a bug). In general, because they do not have direct authority to change the official project codebase, the contribution of the interest community can only moderate the impact of the efforts of the development group on the software complexity.

Past research on the relationship between group cohesion and performance outcomes has been inconsistent (Lott and Lott 1965; Stogdill 1972). Although many studies have found a positive relationship between group cohesion and performance, some studies have not. In their study of the impact of group cohesion on the productivity of programmers, Gowda and Chand found the relationship between group cohesion and productivity of programmers to be insignificant with negative coefficients (Gowda and Chand 1993). Shaw suggests that this inconsistency may be due to effects of motivation (Shaw 1981). Evidence of the effects of motivation through goals on the relationship between group cohesion and performance was found in a study done by Klein and Mulvey (Klein and Mulvey 1995). In the OSS context a cohesive interest community may provide motivation for the development group by focusing their efforts on specific goals when the community requests additional features and bug fixes. This input may be especially important in motivating the development group because they may lack formal goals, timelines, and authority structures as discussed above (Gallivan 2001; Scacchi 2002).

P2: The relationship between the development group cohesiveness and software complexity is moderated by the cohesiveness of the interest community.

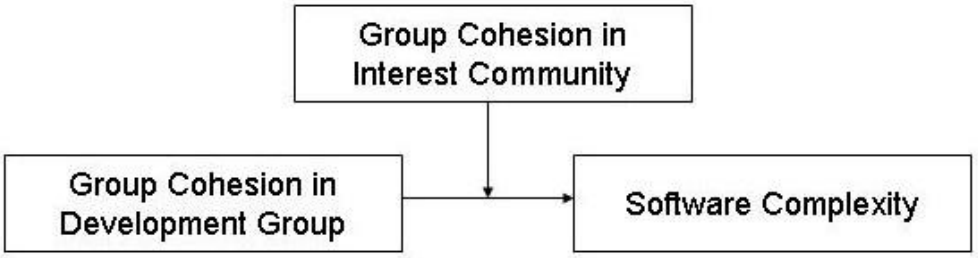

Figure 1

Proposed Operationalization of the Model

A sample of OSS projects from the soureforge.net website can be used to find empirical validation for this model. The sample should be random, but the projects should be required to operate under an OSS approved licenses and be active during the time of the study.

A secure Web-based survey can be used to get feedback from participants in the sample of projects to understand the level of cohesion in the development group and interest community. Group cohesiveness can be measured using Evan and Jarvis' Group Attitude Scale (Evans and Jarvis 1986). This scale has been used in prior research on computer mediated groups (Alavi, Wheeler and Valacich 1995; Yoo and Alavi 2001). This scale measures the affective aspect of group cohesiveness (Evans and Dion 1991). An aggregation of responses (Guzzo, Yost, Campbell and Shea 1993; Jarvenpaa, Knoll and Leidner 1998) to get measures of cohesion at the group and community level can be used. As is appropriate for this method, analysis should be conducted to ensure that respondents to the survey do not differ substantially from non respondents.

Some studies have begun to look at software complexity in the open source context (Paulson, Succi and Eberlein 2004). Complexity has been explored using several measures including lines of code, coupling and cohesion.

The size of the development group and the size of the interest community should be used as controls in this model. Size is an important control variable in this model because size has been linked to lower levels of cohesiveness (Shaw 1981).

IMPLICATIONS FOR RESEARCH AND PRACTICE

Implications of the Organizational Structure

Modeling the organizational structure that supports OSS development as separate but related development groups and interest communities advances the research on the antecedents of OSS success in several ways. This model highlights linkages to prior research, enables the examination of more relationships and allows for precise measurement of constructs.

By separating the interest community from the development group we facilitate the use of prior research in the understanding of the factors that lead to success for OSS development. There are well developed and distinct literatures for virtual communities (e.g. Blanchard and Markus 2004) and software development groups (e.g. Curtis et al. 1979). These literatures provide different measures of success for development groups and interest communities. For instance an important success variable for a development group's success is software complexity as an indicator of software quality (Banker, Datar, Kemerer and Zweig 1993), but this is not an appropriate measure for many virtual communities that are not associated with OSS projects. Similarly, because the coordination burden is high for development groups (Carmel 1999), the communication technology needs of the development group and the interest community are different. The separate literatures present models of the effectiveness of different communication technologies for the two different types of entities. The software development literature offers insights into software development which can be adapted to the context of the OSS development groups to understand the factors that lead to success for OSS development groups. The virtual community literature can similarly be adapted to understand the antecedents to success for the open source interest community. By modeling the development group and interest community separately we hope to encourage future researchers to pay attention to the unique needs of these entities and thereby leverage prior research to understand how they impact the success of the open source project.

Another benefit to identifying the development group and the interest community as separate but related entities is that it presents an opportunity for the exploration of many new relationships. We can explore the main effects between characteristics of these two different entities and the many dimensions for success in the OSS context, such as popularity or software complexity. We can also explore how different features of the interest community, such as the size or the cohesion

of the community, impact the development group characteristics and the development group's relationship to performance variables. The model presented in this paper is one of many models that this conceptualization enables. We encourage future researchers to consider others.

Finally, consideration of the interest community and development group as separate entities will allow empirical studies to better understand the inner workings of the OSS development project. Measuring the development group and the interest community together may bias results. Because of the larger number of individuals often found in the interest community any simple aggregation will over represent the interest community. One other empirical concern is that research that attempts to empirically study this model should use caution in assigning individuals to the interest or development group. Individuals may move between the two entities and this should be considered as research progresses in this area.

Implications for Practice

This work offers insight for corporate and government sponsors of OSS development. By understanding OSS projects as being composed of two separate entities, sponsors of OSS projects are better able to identify ways to improve the process of OSS development. Sponsors of open source projects can improve OSS development by using the different functions of the two entities to choose and implement appropriate communication systems for both. Sponsors can also improve development by consulting this organizational structure when some projects are not performing well. Perhaps the OSS development group is performing well, but the project is not meeting expectations because the interest community is not fulfilling its critical role in the process. Sponsors of OSS projects can also consider a similar case in which the interest community is functioning well, but the development group needs additional resources.

CONCLUSION

This research contributes to the growing body of literature that seeks to understand the factors that lead to success in the OSS context. It does so by describing an organizational structure for open source projects that enables the production of low complexity software through OSS development. This organizational structure consists of a development group and an interest community. A discussion of the unique nature of OSS development groups is undertaken to understand the distinct challenges and opportunities in this context. This discussion is used to develop a theoretical conceptualization of the relationship between an open source development group and a related interest community for a single project. Specifically, we underscore the importance of the interest community and identify the way the interest community impacts the software development process. We argue that the relationship between the cohesion of the development group and software complexity for the project is moderated by the cohesion of the interest community. Finally, we discuss implications of this model for research and practice.

ACKNOWLEDGMENT

The authors would like to thank Ritu Agarwal for helpful comments on previous versions of this manuscript.

REFERENCES

1. Alavi, M., Wheeler, B.C., and Valacich, J.S. "Using IT to Reengineer Business Education: An Exploratory Investigation of Collaborative Telelearning," *Management Information Systems Quarterly* (19:3) 1995, pp 293-312.
2. Banker, R.D., Datar, S.M., Kemerer, C.F., and Zweig, D. "Software Complexity and Maintenance Costs," *CACM* (36:11), November 1993, pp 81-94.
3. Barki, H., and Hartwick, J. "User Participation, Conflict, and Conflict-Resolution - the Mediating Roles of Influence," *Information Systems Research* (5:4) 1994, pp 422-438.
4. Blanchard, A., and Markus, M.L. "An Experienced 'Sense' of a Virtual Community: Characteristics and Processes," *The Data Base for Advances in Information Systems* (35) 2004, pp 65-79.
5. Carmel, E. *Global Software Teams: Collaborating Across Borders and Time Zones* Prentice Hall, Upper Saddle River, NJ, 1999.
6. Curtis, B., Sheppard, S.B., Milliman, P., Borst, M.A., and Love, T. "Measuring the Psychological Complexity of Software Maintenance Tasks with the Halstead and McCabe Metrics," *IEEE Transactions on Software Engineering* (5:2) 1979, pp 96-104.
7. Cushing, B.E. "Frameworks, Paradigms, and Scientific Research in Management Information Systems," *Journal of Information Systems* (4) 1990, pp 38-59.
8. Edwards, K. "Epistemic Communities, Situated Learning and Open Source Software Development," *Epistemic Cultures and the Practice of Interdisciplinary Workshop*, Norwegian University of Science and Technology, 2001.

9. Evans, C.R., and Dion, K.L. "Group Cohesion and Performance: A Meta-Analysis," *Small Group Research* (22:2) 1991, pp 203-216.
10. Evans, C.R., and Jarvis, P.A. "The Group Attitude Scale: A Measure of Attraction to Group," *Small Group Research* (22:2) 1986, pp 203-216.
11. Feller, J., and Fitzgerald, B. *Understanding Open Source Software Development* Addison-Wesley, London, 2002.
12. Gallivan, M.J. "Striking a Balance between Trust and Control in a Virtual Organization: A Content Analysis of Open Source Software Case Studies," *Information Systems Journal* (11:4) 2001, pp 277-304.
13. Gowda, R., and Chand, D. "An Exploration of the Impact of Individual and Group Factors on Programmer Productivity," *ACM Conference on Computer Science*, 1993, pp. 338-345.
14. Guzzo, R.A., Yost, P.R., Campbell, R.J., and Shea, G.P. "Potency in Groups: Articulating a Construct," *British Journal of Social Psychology* (32:1) 1993, pp 87-106.
15. Hann, I.-H., Roberts, J., and Slaughter, S. "Why Developers Participate in Open Source Software Projects: An Empirical Investigation," *International Conference on Information Systems*, Washington, D.C., 2004.
16. Hars, A., and Ou, S. "Working for Free? Motivations for Participating in Open Source Projects," *International Journal of Electronic Commerce* (6:3) 2002, pp 25-39.
17. Hunton, J.E., and Beeler, J.D. "Effects of User Participation in Systems Development: A Longitudinal Field Experiment," *MIS Quarterly* (21:4), December 1997, pp 359-388.
18. Jarvenpaa, S., Knoll, K., and Leidner, D. "Is Anybody Out there? Antecedents of Trust in Global Virtual Teams," *Journal of Management Information Systems* (14:4) 1998, pp 29-64.
19. Keyton, J., and Springston, J. "Redefining Cohesiveness in Groups," *Small Group Research* (21:2) 1990, pp 234-254.
20. Klein, H., and Mulvey, P. "Two Investigations of the Relationships among Group Goals, Goal Commitment, Cohesion, and Performance," *Organizational Behavior and Human Decision Processes* (61:1) 1995, pp 44-53.
21. Krogh, G.v., Spaeth, S., and Lakhani, K. "Community, joining, and specialization in open source software innovation: a case study," *Research Policy* (32) 2003, pp 1217-1241.
22. Lakhani, K., and Hippel, E.v. "Community, Joining, and Specialization in Open Source Software Innovation: A Case Study," *Research Policy* (32:6) 2003, pp 923-943.
23. Lerner, J., and Tirole, J. "Some simple economics of open source," *Journal of Industrial Economics* (50:2) 2002, pp 197-234.
24. Lott, A.J., and Lott, B.E. "Group Cohesiveness," *Psychological Bulletin* (64) 1965, pp 259-309.
25. Markus, M.L., Manville, B., and Agres, C.E. "What makes a virtual organization work?" *Sloan Management Review*) 2000, pp 13-26.
26. Paulson, J., Succi, G., and Eberlein, A. "An Empirical Study of Open-Source and Closed-Source Software Products," *IEEE Transactions on Software Engineering* (30:4) 2004, pp 246-256.
27. Raymond, E.S. *The Cathedral and the Bazaar: Musing on Linux and Open Source by an Accidental Revolutionary* O'Reilly, Sabastopol, CA, 2001.
28. Scacchi, W. "Understanding the requirements for developing open source software systems," *IEE Proceedings on Software* (149:1), February 2002, pp 24-39.
29. Shaw, M.E. *Group dynamics: The psychology of small group behavior*, (Third ed.) McGraw-Hill, New York, 1981.
30. Stewart, K.J., and Gosain, S. "An exploratory study of ideology and trust in open source development groups," *22nd International Conference on Information Systems*, New Orleans, LA, 2001.
31. Stogdill, R.M. "Group productivity, drive and cohesiveness," *Organizational Behavior and Human Decision Processes* (8) 1972, pp 26-42.
32. Von Hipple, E., and Von Krogh, G. "Open Source Software and the Private-Collective Innovation Model: Issues for Organization Sciene," *Organization Science* (14:2) 2003, pp 209-223.
33. Weinberg, G. *The Psychology of Computer Programming (Silver Anniversary Edition)* Van Nostrand Reinhold, New York, 1998.
34. Ye, Y., and Kishida, K. "Toward an Understanding of the Motivations of Open Source Software Developers," *International Conference on Software Engineering*, Portland, OR, 2003, pp. 419-429.
35. Yoo, Y., and Alavi, M. "Media and Group Cohesion: Relative Influences on Social Presence, Task Participation, and Group Consensus," *Management Information Systems Quarterly* (25:3) 2001, pp 371-390.