

Organizational Structure of “User Collaboration Community”: Insights from the Case of an Open Source Software Project

Jen-Fang Lee, Tzu-Ying Chan

National Cheng Chi University of Technology & Innovation Management, Taiwan

E-mail: jflee@nccu.edu.tw; g8359503@nccu.edu.tw

Abstract

Current literatures on organization innovation do not investigate the innovation model which is composed of “network between individuals”. Therefore, this study proposes the concept of the “User Collaboration Community”, tries to understand this new phenomenon by conducting OSS project is the subject of this analysis, borrows the observation variables and propositions adopted by Mintzberg on structures of the innovative organization. This study further infers the conceptual framework and a series of hypothesis on the organizational structure of the user collaboration community. We expect that the proposition of this hypothesis will function as a concrete description and presentation of the innovation model of the “User Collaboration Community”.

1. Introduction

Current literatures on organization innovation do not investigate the innovation model which is composed of “network between individuals”. How is the organizational structure of the innovation model operated? Can the current organization models and theories explain or describe the organization platform of the “user collaboration community”? Therefore, this research will pinpoint the deficiencies of these literatures and try to answer these questions. This study first conducts a brief review of some important scholar’s research frameworks and observation variables for innovative organizations. After a comparison, this study decides to adopt Mintzberg’s observation dimensions and propositions on the innovative organization as a basis for studying the formats of the organization platform of the user collaboration community, and also adopt three open source software projects as subjects for analysis.

This structure of this article is as follows: Section 2 is a literature review. Section 3 proposes the conceptual framework of the organizational structure of the user collaboration community. Section 4 is data collection and analysis, and Section 5 is the conclusion.

2. Literature review

The review of related literatures on organization innovation indicates that most literatures focus on innovation behaviors and determination factors for innovation adopted “inside an organization”, emphasizing long-term, cumulative, durable innovation within one organization. In other words, it is organization innovation at the firm level, such as self-organization form [20], heavyweight development team [2], spider web organization [21], hypertext organization [19], fractal organization, practice community [1,25], strategic community [22], and global research network [6,11], etc.; In addition, many literatures also discuss innovation models of network relationships formed by organizations and similar organizations in the external environment, focusing on flexibility and combination. Knowledge accumulates in each actor. The innovation is achieved through the combination of different actors. In other words, it is organization innovation at the industry level, such as virtual organization, network organization, boundless organization, etc.[4,7].

In regard to research fields, literatures on organization innovation did not investigate the innovation model composed of “network between individuals”. The most famous innovation model of this type is the operation of Linux community. How is the organizational structure of the innovation model operated? Can the current organization models and theories explain or describe the organization platform of the “user collaboration community”?

[26] indicated that individual variables, organization variables, and environment consequence variables all contribute to organization innovation. Of these three variables, the organization variable carries the most weight in explaining the organization innovation. In the organization variable, scholars often adopt “organizational structure” to study organization innovation. [1-7] thought that the essence of organizational design is the manipulation of a series of parameters that determine the division of labor and the achievement of coordination. He proposed nine major parameters for structural design.

Many scholars have proposed capabilities of organization innovation such as the degree of organization formalization, the degree of organization specialization, the degree of organization centralization, the entrepreneurship, etc. All of these are important variables that may affect organization innovation. From the literature review, it is not easy to come up with a consensus on organization innovation. In addition, [15] indicated that most studies on exploring the predicting factors for organization innovation focus more on the individual predictability of different variables, but less on the joint predictability of various variables. As a result, these studies could not provide an overall picture of organization innovation and inhibit the accumulation of research results on organization innovation. Configuration approaches, on the other hand, provided a solution for resolving the problems mentioned above and offered an alternative method for organization analysis. For those scholars who adopted configuration approaches, an organization is a combination of various factors that occurred simultaneously. Thus, parameters for organization design could not be selected independently [16,17]. Therefore, this study adopts the theory of organization structure proposed by [17], based on configuration approaches, especially his integrated analysis on innovative organization structure, as the main analytical framework of this study.

3. Conceptual Framework

This study is trying to answer the question "Can current organization models or theories explain or describe the organizational structure of user collaboration community?" Therefore, this research tries to utilize Mintzberg's observation and inference on innovative organization to construct the conceptual structure of this research. Even though Mintzberg used business as the analysis unit, this study uses the individual network as the analysis unit, and this study believes that there will be no difference in the

operation model for the innovative organization caused by the difference of participants. The induction of the conceptual framework is listed below:

4. Data Collection and Analysis

The operation platform of OSS project organization not only helps clarify the innovation model of "the organization platform of user collaboration communities", it further provides detailed process data regarding the internal operation of the project. For scholars who study organization innovation, this is a tremendously valuable database. It makes up the insufficiency of the research method of subjective recognition or pure description caused by the difficulties of obtaining data.

This research adopts three open source software projects as the subjects of case study. These projects are Linux kernel, Apache, and SFS (School Free Software). Because each of the three projects has a corresponding web site[26,27,28], this research uses documents on the web site, file records, and participation in observation as major manners of data collection. Linux[3,5,8,10,13,14,23,24] and Apache [9,12,18] focus mainly on the analysis of secondary data. However, SFS interviews related community members who participate in the operation of a project. It also uses VBA program language to automatically grab CVS and BUGDB[30] data for analysis.

SFS projects are briefly described as follows: A grammar school teacher in Taiwan, Ying Guan Chen, developed a curriculum management software and allowed everyone to use, revise, and pass around the software for free. So far there are teachers in more than sixty schools participating in the revision of the software, and there are more than two hundred schools, even schools in Hong Kong and mainland China, trying to install the software.

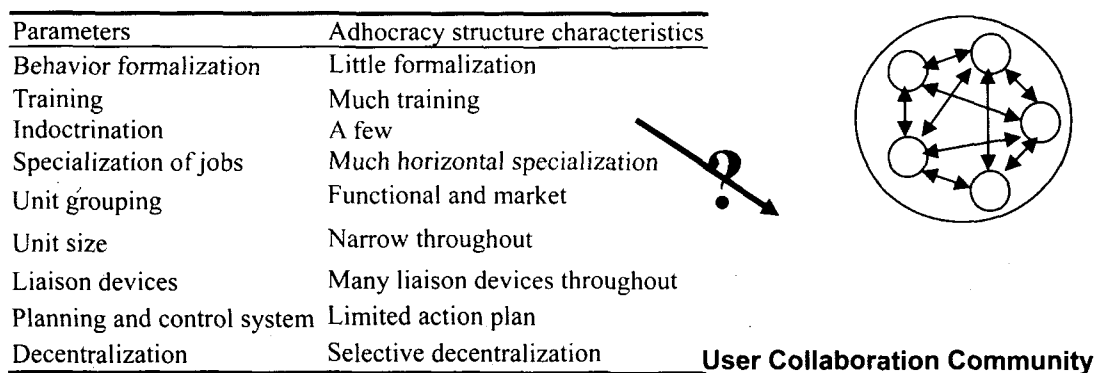


Figure 1. Framework of organizational structure analysis for "user collaboration community"

[17] felt that innovative organization is a highly organic structure with very few formalized behaviors. Members of this organization all receive formal and strict training to perform assigned work.

From table 2, we can tell that the Identifying of the development work of the OSS communities is proposed or reported by the communities. Nobody will assign anybody to perform what task. Whoever feels like performing the task may choose to be responsible for what they specialize or what they prefer. Therefore, there is no need to give orders, and there is no clear division of responsibilities. It is impossible to have a clear picture of division of works or to organize a group. As a result, it is impossible to have any formal or written documents to explain or to operate the activity procedures of members of the organization.

Hypothesis 1 The user collaboration community has the characteristics of a low degree of behavior formalization.

[17] felt that when an organization cannot standardize its behavior, it will assure people who perform the tasks possess the knowledge and skills required for the tasks through intense training beforehand. In the three projects, the phenomenon of “standardization of skill” is not discovered. However, in order to allow beginners to continuously and periodically participate in the operation of the project and also to assure participants’ withdrawal from the project will not impact the operation of the project, the OSS project utilized informal training system to assure participants in the project possess a certain degree of knowledge and skills. For example, the design of property right system allows source to be shared, studies, verified, and passed on among the communities. In addition, the matched screening mechanism allows the project to verify the participants’ knowledge and skills. It makes the participants’ performance visible and transparent. The examination process of the program is similar to the peer evaluation in the scientific field.

Hypothesis 2 The user collaboration community has an informal training system.

[17] argued that “indoctrination” is considered to be a substitute of “behavior formalization” and “training”. OSS community is composed of a group of members dispersed everywhere in the world. For most of them who are not compensated for their work, it is very important that whether they share a common belief. It relies on volunteers’ self-encouragement. It utilizes a complicated but obscure monitoring and encouragement system to reinforce its norms and

also allows everyone to contribute. It compensated the participants with peer recognition, but not money. Earning peer recognition is an important motivation of participating in open source. The desire of maintaining and strengthen their own reputation is the key mechanism to assure the progress.

Hypothesis 3 The informal social control system for the platform of the user collaboration community is very active.

[17] felt that there is a high degree of multiple horizontal specialization between members of an innovative organization. It is common to group experts in each field based on their specialized functions, and they then perform in the manner of a project team. Such organizations rely heavily on project teams and task teams and tend to result in numerous teams. However, the size of each team is small.

[17] thought that in order to allow the operation of an innovative organization, the organization has to maintain a variety of members, and each member has to specialize in his own specialty. Their responsibilities are few and the scope is narrow. The core developers in the Linux and the Apache projects devote only in several modules. It shows a high degree of specialization. However, a half of the core developers in the SFS project participate in the addition and the revision of 50% of the modules. The different results are probably caused by the small difference of the technology among each module and the development history of the project is short.

Hypothesis 4 The organization platform of the user collaboration community has the characteristics of a high degree of multiple horizontal specialization

From the three projects, we discover that the OSS project not only separate groups based on the functions, such as groups of core development and testing, when it constructs the project, it also tries to allow the product structure reflect on the project structure. [17] felt that the project team of the innovative organization should be kept at a very small scale due to the coordination. It indirectly leads to a large number of projects. From the three projects, we find that most projects are divided into numerous modules and the number of participants in each module is not large. [18] had the same research result. OSS is a big project. Therefore, it is divided into smaller sub-projects for easier management.

Hypothesis 5 The organization platform for the user collaboration community proceeds in the manner of a project team.

Hypothesis 6 There are many project teams in the organization platform for the user

collaboration community, but the scale of each team is very small.

[17] felt that there are various liaison mechanisms within teams or between teams in an innovative organization to promote mutual adjustments. Power is separated selectively in these teams, and the styles of the power division are selective vertical and horizontal decentralization. Planning and control systems are the grassroots model, and the information flow and the strategy process are very flexible and very informal.

Because there are numerous project teams in the OSS project, it is necessary to have a communication mechanism to coordinate the operation of each team. In the three projects, we discover that in addition to coordinate the operation of the project through abundant collaboration development tools, these projects also have coordination staffs such as project leaders, testing managers, and product managers.

Hypothesis 7 There are many liaison mechanisms in the organization platform of the user collaboration communities

Even though the mechanism of the release of decision-making power is different among each project, they all release the decision-making power to community developers who are highly familiar with each module. From table 3, we can tell that power is also divided among team members. The addition and revision of files and modules are generated jointly in a certain format through the coordination and cooperation of the communities. The principle of the division of decision-making power is totally based on the characteristic of the decision.

Hypothesis 8 The organization platform of the user collaboration communities have the characteristics of selective vertical and horizontal decentralization

Summarizing the development flow of the three projects (table 2), we discover that the plan and the control system of these projects is grassroots model. It is different from the strategy setup manner of the traditional greenhouse model. The web site of the project is an open testing ground. It provides a space that nobody can plan and control. In addition, from the summary of the comparison of the origin and the current condition of the development (table 1), we can tell that these projects were not developed based on the current objective. Follow the growth of user community the direction of the development software functions tends to be beyond the expectation of the original founder. It derives many applications that are not expected originally.

Hypothesis 9 The planning and control system of the organization platform of the user collaboration communities is the grassroots

model.

5. Conclusion

This research tries to use Mintzberg's observation and inference on innovative organization to explain the organizational structure of OSS project. This article discovers that even though the analysis unit of OSS project is a community composed of individuals, but not manufacturers, the organization operation model still conforms to Mintzberg's inference on the organization structure of which the mission is to perform innovation.

Acknowledgements

The financial support from National Science Council of Taiwan is gratefully acknowledged (the project NO: NSC 92-2416-H-004-021)

References

- [1] Brown, J. & Duguid, P. (1991), "Organizational Learning and Communities of Practice: Toward a Unified View of Working, Learning, and Innovation", *Organization Science*, 2(1): 40-57
- [2] Clark, K. & Wheelwright, S. (1999), "Organizing and Leading 'Heavyweight' Development Teams", *California Management Review*, 34(3): 9-28.
- [3] Dafermos, George (2001), "Management and Virtual Decentralized Networks: The Linux Project", Durham Business School M. A. Thesis
- [4] Davidov, W.H. & Malone, M.S (1992), *The Virtual Corporation*, Harper Business, New York
- [5] Dempsey, B. J., D. Weiss, P. Jones & J. Greenberg (1999). 'A Quantitative Profile of a Community of OpenSource Linux Developers', School of Information and Library Science, University of North Carolina at Chapel
- Hill, SILS Technical Report, TR-1999-05.
- [6] Gassmann, O. & Zedtwitz, M. (1998), "Organization of Industrial R&D on a Global Scale", *R&D Management*, 28(3): 147-161
- [7] Hedbeag, B., Dahlgren, G., Hansson, J. & Olive, N. (1997), *Virtual Organizations and Beyond: Discover Imaginary Systems*, Wiley, London.
- [8] Jae Yun Moon, Lee Sproull, (2000), "Essence of Distributed Work: The Case of the Linux Kernel", 5, No. 11, First Monday, November 2000. www.firstmonday.dk/issues/issue5_11/moon/
- [9] Jagielski, Jim (1999), "The Apache Success Story - Exploring an Open Source Development Process", <http://www.newarchitectmag.com/archives/1999/10/jagielski/>
- [10] Koght, Bruce & Anca Metiu (2001), "Open Source Software Development and Distributed Innovation", *Oxford Review of Economic Policy*, 17(2): 248-264.
- [11] Kuemmerle, W. (1997), "Building Effective R&D Capabilities Abroad", *Harvard Business Review*, 75(2): 61-70
- [12] Lakhani, Karim & von Hippel (2003), "How open source software works: "free" user-to-user assistance", *Research Policy*, 32:6,923-943
- [13] Lee, Gwendolyn K. & Robert E. Cole (2000), "The

Linux Kernel Development as a model of knowledge creation”, working paper
 [14] MacCormack (2001), *Product-Development Practices That Work: How Internet Companies Build Software*, MIT Sloan Management Review, Winter
 [15] Meyer, A. & Goes, J.(1988), “Organizational Assimilation of Innovations: A Multilevel Contextual Analysis”, *Academy of Management Journal*, 31(4): 897-923
 [16] Meyer, A., Tsui, A. & Hinings, C. (1993), “Configurational Approaches to Organizational Analysis,” *Academy of Management Journal*, 36(6): 1175-1195
 [17] Mintzberg, H. (1979), *The Structuring of Organizations*, Englewood Cliffs, NJ., Prentice Hall
 [18] Mockus, A., Fielding, R.T. & Herbsleb, J. (2000), “A case study of open source software development: the Apache server”, Proceedings of the 22nd International Conference on Software Engineering (ICSE 2000), Limerick, Ireland. June 4-11, ACM Press. p.263-272
 [19] Nonaka & Konno, N. (1993), “Knowledge-Based Organization”, *Business Review*, 41(1): 59-73
 [20] Quinn, J. B.; Baruch, J. J.; & Zien, K. A. (1997),

Innovation Explosion, The Free Press, N.Y.
 [21] Quinn, J.B., Philip, A. & Syndey, F. (1996), “Management Professional Intellect: Making the Most of the Best.”, *Harvard Business Review*, 74(2): 71-80
 [22] Storck, J. & Hill, P. (2000), “Knowledge Diffusion Through Strategic Communities”, *Sloan Management Review*, 41(2): 63-74
 [23] Torvalds, L. & Diamond, D. (2001). Just for Fun
 [24] Tuomi, Ilkka (2002), *Networks of Innovation - Change and Meaning in the Age of the Internet*, Oxford University Press
 [25] Wenger, E. & Snyder, W. (2000), “Communities of Practice: The Organizational Frontier”, *Harvard Business Review*, 78(1): 139-145
 [26] Wolf, R.A. (1994), “Organizational Innovation: Review, Critique and Suggested Research Directions”, *Journal of Management Studies*, 31(3): 405-430
Web sites
 [27] Linux-kernel project <http://www.kernel.org>
 [28] Apache project <http://www.apache.org>
 [29] SFS project <http://sfs.wpes.tcc.edu.tw/index.php>
 [30] SFS bug track system http://cvs.tnc.edu.tw/bugtrack/login_page.php

Table 1 The Origin and Description of Open Source Software Communities

	Linux Kernel	Apache	SFS
Starting Time	1991-	1995-	2000-
Technology	Operation system	Web server	Information system
Characteristics			(Curriculum management)
Project Origin	Torvalds developed end virtual program to connect newsgroups of Usenet.....	Behlendorf gathered several people to develop revised program for Http web server...	Ying Guan Chen developed a system for books loan service
Licensing Agreement	GPL	BSD	GPL
Community Size (The result of innovation?)	The operation system that has the highest growth rate. Support more than ten different types of CUP. It is good for multiple applications. Better than commercial software in safety, efficiency, and reliability. Evolved more than a hundred versions. There are more than four thousand web sites in the world.	Killer Application(67%) may cross platforms to be operated on about thirty operation systems. (IIS supports only one platform. The market share is 21%)	More than 200 schools installed the system. It even prevails in Hong Kong and mainland China. It also generates community communication stations for other five counties and cities and SFS version for vocational schools. More than 60 schools participate in the revision of the software.

Table 2 The Development Process of Open Source Software

	Linux kernel	Apache	SFS
The recommendation of core members and their responsibilities	Developed from only Torvalds to several hundred developers, assistants. Once a person's reputation is established, the revised program will be passed to this person and this person will gradually become a famous person for studying a specific problem and gradually picks up the tasks of core development teams.	Twenty-five members of the board (AG) are all volunteers without receiving any compensation. They possess the authority of CVS. The core development team includes AG and other active developers.	Started with only Ying Guan Chen to 15 people. They possess CVS authority. Most of them participate in the project through the recommendation of core members.

Identifying of development work	Induce communities' opinions on tasks to be done through discussion groups.	Although AG puts authority limitation in the program to control all revisions on the program, the whole development process is open. Other people may provide feedback on errors or propose directions for improvement.	Communities report errors or propose directions for improvement through news group, mailing list, and Bugzilla.
The assignment and operation of development work	Nobody assigns whom to do what. Anybody who feels like performing the task may handle a certain module.	Anybody may provide patches, develop their favored or specialized modules, or respond other people's questions through mailing list, news group or BUGDB	List problems to be solved through Bugzilla, news group and mailing list Anyone may propose program modules or patches; however, only core members may put source directly in CVS.
Writing programs and testing the program voluntarily	Writing programs and testing the programs voluntarily.	Volunteer developers write programs and test the programs voluntarily on the server to make sure the operation is error free.	Developers voluntarily test the program they write and eliminate the bugs they discover.
Examination	Each of the assistants is responsible for a certain specific dimension of the core program. They screen and examine the patch programs transmitted from other developers, then forward the result to Torvalds	More core developers will examine all changes. Other community members who are not in the core development teams may also participate in debugging. The changes made on the stable releases have to be examined by other developers before the commitment. However, changes made on the development releases will be examined after the commitment.	Nobody will examine the programs in the "program code communication center" before uploading, but core members will examine and test these programs afterwards. If the function of these programs are good, the developers will include these programs as part of the main version. Currently utilize Bugzilla to feedback, control, and manage the progress of debugging.
Releases management	Core programs have parallel release structure that can be classified into hacker version and common user version. These programs are released in an accelerated revision cycle model. Several new versions are released in a month. Revisions are uploaded and released once it is completed.	There is no fixed schedule for the releasing of new versions. Core development team takes turns to be the manager of a certain version. After all approved revised programs are adopted in a new version, and before the version is released, developers are not allowed to make any changes to the depository of the source.	There are stabilized version, development version, and testing version. Once the development version is assured to have no error, it will be released to the stabilized version. The developer personally determines the development condition of a module. In its early days, patches are published periodically for users to download. Currently, the upsfs is adopted and is set to revise every three hours by all schools simultaneously.

Table 3 The Task Division Condition of Open Source Communities

	Linux	Apache [18]	SFS
Revision of programs	<p>_ Even though there are many people, their contribution to the innovation of products is extremely different. (Several thousand teams develop the program, several hundred thousand people work on debug report. However, not many people really contribute tremendously to the core program. There are about ten core developers in the top level.) (2% of the contributors provide 50% of related information.)</p>	<p>_ Even though there is a large number of participants, the change or the addition of program codes is seldom contributed by non-core developers. The addition and maintenance of new functions are almost operated by the core development team. (The top 15 program developers participate in the revision of about 83% of the patch program codes, 88% of new added program codes.)</p> <p>_ In addition to core development team, few developers may propose changes regularly.</p> <p>_ Developers strongly need a certain format to coordinate the generated products. In addition, nobody can individually write the complete program codes for a certain module. (42 files have more than 30 changes. More than 10% changes are made by developers for at least 2 of 40 files (at least f of 20 files).)</p>	<p>_ Core members are in charge of the revision of most programs. (14% of the developers provide 74% of all modules. The founder, Mr. Chen, contributed 23 modules, only 2% of all modules)-Program communication center (Of the 11 developers, one developer contribute 71% of the modules. Next is Ying Guan Chen. Overall speaking, 5 developers participated in the revision of 50% of the modules)</p> <p>-CVS _ There are frequent revising activities for modules and files. (More than 14% of the files are revised over 5 times and 56% of the files are changed over 2 times. Analyzing based on number of changes of the modules, about 43% of the modules are changed over 50 times, 98% of the modules are changed more than 10 times, 2% of the modules are changed under 10 times.)</p> <p>_ Collaboration process (84% of the modules are developed through the collaboration of more than two people.)</p>
Report on errors	N/A	<p>_ Other non-core teams, the large number of user communities, are responsible for the important role of system testing. (Of the top 15 PRs, only 3 are core development teams. There are 213 articles from the top 15 PRs, which is only 5% of all PRs.)</p> <p>_ Debugging through frequent or more efficient methods. It attains lower poor quality ratio. (50% of the PRs solve the problem on the same day, 75% of the PRs solved the problem in 42 days, 90% of the PRs solved the problem in 140 days.)</p>	<p>_ Error reports are mostly operated by other non-core members. (The PR contributed by the core members (only 5 people) is only 23% of all the PR.)</p> <p>_ Communities assume a very important role in finding system errors. (The types of report bugs are mostly error messages. Problems on structure, database, and program code are the second most frequent problem. Problems on "operation" and "installation" are only 12% of all messages.)</p> <p>_ Communities assume an important role in suggesting system functions. (The most common errors reported by communities are "insufficient function". It takes up 34% of all messages.)</p> <p>_ Bugs may be discovered timely and the problems may be solved through the platform developed by the collaboration and cooperation of communities. (The longest time to open a bug is 8.03 days, and the average time to solve a problem is 4.28 days.)</p>