행위 통제 관점에서 바라본 오픈소스 프로젝트 관리

Open Source Project Management-from a Behavior Control Perspective

조 준 기 (Jun Gee Cho) 미국 클레어몬트대학 대학원

요 약

상업용 소프트웨어에 견줄만한 수준의 오픈소스 소프트웨어들이 등장함으로 인하여, 오픈소스 (정보시스템 개발) 프로젝트에 대한 학계의 관심이 고조되고 있는 가운데, 오픈소스 프로젝트의 관리경영 측면에 관한 연구들은 막상 찾아보기 쉽지않은 상황이다. 이러한 현실에서 본 논문은 기존의 전통적인 정보시스템 개발관리에 적용되는 경영관리 이론들이 오픈소스 프로젝트에도 그대로 적용될 수 있는지를 살펴보고자 하는 학문적 관심에서 출발하여 사례연구 방법을 적용하여 행위통제의 관점에서 오픈소스 프로젝트들이 어떻게 관리되고 있는가 라는 문제를 연구하였다. 자료분석과 해석을 통해, 본 연구는 구체적인 사례들에 있어서 개별적인 차이들은 존재하지만, 전반적으로 오픈소스 프로젝트 관리에 있어 다양한 형태들의 행위통제가 이루어지고 있다는 사실을 확인하였다. 이러한 발견은 오픈소스 프로젝트는 그 특성상 행위통제가 적용되기 어렵다고 보는 기존 일부 연구자들의 주장과 배치되는 내용이다. 오픈소스 프로젝트 관리에 관한 보다 더 포괄적인 학문적 이해를 위해서는 향후 결과통제와 자기통제까지 포함하는 보다 더 확대된 연구가 필요할 것으로 보인다.

키워드 : 오픈소스, 통제, 행위통제, 프로젝트 관리, 사례연구

I. Introduction

With the arrival of commercially successful open source software packages whose quality is considered to be comparable to that of commercial software giants, open source software development has drawn attention by many software development entities (Bollinger *et al.*, 1999; McConnell, 1999; Paulson *et al.*, 2004). With respect to applying controls of ISD projects, however, there are a number of challenges posed by open source communities, including lack of mechanisms(e.g., rewards and punishments and structural hierarchy), supporting controls, and difficulty to define the boundary of control, especially in terms of controlees and project objectives.

Open source communities typically perform software development online using virtual channels. Unlike traditional ISD projects, the open source community does not allow exclusive privileges to organizational participants and major contributors. The principle of equality of membership is regarded as one

† I would like to thank Dr. Lorne Olfman for his helpful comments and suggestions on this work.

of the reasons behind difficulties of organizational control in open source software development(Jensen and Scacchi, 2005). Open source project communities are known to have a core-periphery structure with corresponding distinctive roles(AlMarzouq *et al.*, 2005; Crowston *et al.*, 2006; Iannacci, 2005). In addition, many open source projects are initiated by individual developers who have only ideas or prototypes. With the progress of a project, bugs are removed and features are added, making software more useful and less imperfect, yet the original goal becomes more complicated and multi-faceted.

Another understanding about open source software projects is that their activities rely on virtual communities to foster knowledge activities(Lakhani and Hippel, 2002). This view is supported by the fact that one of the motivations of participants is learning through participation(AlMarzouq *et al.*, 2005; Hippel and Krogh, 2003). Participants exchange knowledge through complex interactions in community practices.

These characteristics of open source software projects which are different from those of traditional software projects raise a question whether knowledge of traditional ISD project management is applicable in open source software projects, which comprise new contextual settings. A control mechanism, one of key concepts in the management of traditional ISD projects, is one of the factors which has attracted researchers' interests.

In an attempt to increase the knowledge about ISD management in a new emerging software development context, this study explored control mechanisms in open source projects. To narrow the research scope, this paper focuses on behavior control, which can be a meaningful research topic given that a number of researchers argue that socio-political and human aspects overwhelm structural aspects of open source projects(Lane and Basnet 2005). Some argue that behavior control is even infeasible in open source software ISD projects because behavior control potentially hinders participants' autonomy, which an important aspect of the innovation process relies(Xu *et al.*, 2005; Nidumolu and Subramani 2003).

The main research question of this study is whether open source software development projects construct any behavior control strategy. To find the answer, this study used a case study approach based on Yin (2003). This research aims at contributing to generalize our understanding about ISD project control, and based on such understanding, principles of control in one context may become reusable in another through a series of systematic interpretations of those principles.

Ⅱ. Previous Research

Regarding control over traditional ISD projects, a number of researchers have studied collective activities of a group unit(a team) including established procedures and standards for compensation and penalties on behaviors and outcomes(Choudhury and Sabherwal, 2003; Henderson and Lee, 1992; Kirsch, 1997), and based on the concept of managerial control. In the managerial discipline, researchers classify control into two modes: formal and informal(Ouchi, 1979, 1980; Eisenhardt, 1985; Manz *et al.*, 1987). They divide formal control into behavioral control and outcome control based on whether controlled tasks are observable(observability) or measurable(measurability). They view informal control as either clan control and self control. Informal controls are derived from organizational decisions when they lack of conditions for formal controls. Organizations utilize clan control on common values, cultures, and norms existing in the organizations, or self control based on characteristics(professional and non-routine tasks).

Following the concepts of managerial control, researchers perceived that organizations control ISD projects by combining control mechanisms(a portfolio) which are chosen based on contextual conditions before and during ISD projects(Kirsch 1996, 1997; Choudhury and Sabherwal, 2003). These researchers see task characteristics(observability, measurability, project size, etc.), role expectations, and project-related knowledge and skills as factors influencing organizational decisions on control mechanisms(Kirsch 1997). The contextual settings of these factors can be changed over time and across control mechanisms(Kirsch 1997, Choudhury and Sabherval 2003).

On the other hand, a number of researchers perceive open source ISD as an alternative methodology to traditional methodology of software development(e.g., Rapid Application Development) (Feller and Fitzgerald, 2000; McConnell, 1999). The basic idea behind open source ISD is nicely explained as follows: "*When programmers can read, redistribute, and modify the source code for a piece of software, the software evolves. People improve it, people adapt it, and people fix bugs.*"[1] The production of large scale and quality software packages such as Apache, Linux, and eClipse have drawn attention to open source software development.

Through an empirical observation, Raymond(1999) explains that open source development draws more eyes to watch source code, increasing the likelihood of discovering code bugs. This enables developers to concentrate more on code production than retrospective code checking, resulting in high quality source software.

Several studies present the observations of control of open source ISD directly and indirectly. Through a content analysis of a set of published case studies of open source software, Gallivan(2001) investigated the influence of control and trust in open source ISD performance. His findings show that control functions are an important criterion for efficiency. He interprets this finding by applying the structural concept of open source communities, which is observed to have a core-periphery shape(AlMarzouq *et al.*, 2005; Crowston *et al.*, 2006). Community members continuously move between core and periphery. Some researchers note that open source communities perform control over members through sanctions and rewards(Fang and Neufeld, 2006; Markus *et al.*, 2000). Raymond(1999) argues that reputation is the key behind such conformity to community rules.

The interpretive studies of control types explain that certain types of formal controls are not feasible to apply in open source projects(Lane and Basnet, 2005; Xu, *et al.*, 2005). Open source communities are perceived as virtual organizations(Gallivan, 2001). As the observation of behaviors is not feasible in this context, the application of formal controls is considered to be limited. Xu *et al.*(2005) argue that open source ISD is without behavior control. They understand that open source software development is an incremental innovation process based on participants' autonomy and initiatives, and pre-defined procedures impede such process.

Iannacci(2005) reports that there are three coordination mechanisms in the open source community: standardization, loose coupling, and partisan mutual adjustment. The open source community uses predefined procedures and routines as a standard. In a large scale ISD like Linux, multiple subgroups cooperate under independent authority. Participants in open source projects construct a social network while pursuing their local interests in the physical environments. Thus, the coordination occurs through con-

1) http://www.opensource.org/accessed on July 31, 2008.

tinuous adjustments of individuals between their localized interests(subgroups) and emergent structures constructed through a few particular projects that appeal to them.

Lattemann and Stieglitz(2005) suggest three types of governance exist in open source communities: direct, indirect, and social. Direct governance refers to control by monitoring behaviors(behavior control). Evaluation of outputs compared to intended goals (output control) becomes indirect governance. Social governance concerns conformity to community rules and cultures. Although Lattemann and Stieglitz's observations use many different terms to explain controls in open source software development, these controls are equivalent to control types of researches of traditional ISD projects. The goals and contextual situations of specific controls observed in their study are not different from those of traditional ISD projects.

Ⅲ. Research Approach and Methodology

To find out whether behavior control is exercised in the open source ISD context, this research collected and analyzed explicit data of activities of six selected open source projects from Sourceforge.net, which is the largest open source hosting service. The research process followed the case study guidelines outlined by Yin(1993, 2003) and Miles and Huberman(1984). This research searched for determinations of controls and related circumstantial conditions by observing publicly available communications of open source project teams. In order to focus on managerial project leaders' decisions to control project activities, this research concentrated on the communications of project leaders with other participants. Potential instances of critical transformations of structures and behaviors were determined based on key episodes such as debates, arguments, discussions, and resolutions appearing in multiple communication threads over time.

3.1 Pilot Study

PILOT STUDY APPROACH

Yin(1993, 2003) explains that the case study can be considered as a type of experiment. He advises researchers studying multiple cases to deal with each case as a subject of an experiment, and replicate the study of one case in analyzing another. For replication, I reviewed whether instances of behavior control uncovered in one case were to be found in other cases, and investigated whether conditions underlying those controls were also discovered in other cases(a theoretic replication). Based on results of previous studies about controls of outsourced ISD projects

〈Table 1〉 Data template of Behavior Control

Expected Instances	Revised Instances(based on Pilot Case)
Project plan, project meeting, progress report(check), walkthrough, documentation, standards, conference calls(e-mails), any collocation, quality assurance process, personnel(member) change	◦ Planning: Project plan or project roadmap, To-Do lists ◦ Documentation: FAQ, wiki, manuals, help ◦ Quality Assurance: a series of processes to improve quality(testing, releasing candidate), ◦ Rule Construction: standards(coding and naming convention for code submission), specific working procedures

(Choudhury and Sabherwal, 2003), I constructed a data template matrix(Miles and Huberman, 1994; Yin, 2003). The matrix included control practices exercised at the operational level, such as documents, scheduling, specification, and procedural protocols. Using a draft of the template, I conducted a pilot test by observing the public and longitudinal data of one open source ISD project.

<Table 1> displays the initial data template and its revision based on the analysis of the pilot case. The term "project roadmap" is frequently used to refer to the project plan of traditional ISD projects. Participants used both terms to indicate the abstract level of project plans. From the roadmap, readers can find information about variations of features of the upcoming versions. The project team produced a number of online documents. They are FAQs, a manual for developers, and a user guide. Because the project community has global participation, a few unofficial online documents were also available to serve specific foreign language speakers. Documents for developers detailed conventions on source code(e.g., a mixed use of Java and C++) and concepts of modules, and mandated code contributors comply with the specified standards.[2] Online documents also showed the project team's concerns about unacceptable behaviors such as vandalism in community resources.

3.2 Trust issues of the research

RELATED TRUST ISSUES

Most empirical researchers must face four quality issues of construct validity, internal validity, external validity, and reliability(Yin, 1993, 2003). According to Yin, internal validity is only applicable in explanatory and causal studies. External validity refers to the ability of the study's findings to be generalized. Reliability defines the level of effectiveness of a study's findings to be replicable. A research study using case methodology also should address general quality issues. This study follows Yin's advice to address quality issues(Yin, 2003). Thus, I utilized multiple sources of evidence and key informants' review of draft case study reports for construct validity, pattern-matching for internal validity, replication logic in multiple-cases for external validity, and database and script notes for reliability.

Construct Validity

Operational Measures

In this research, I presumed that contextual factors would have causal relationship with control practices. Using a data template <Table 1> that could limit the scope of the observation to a few specific aspects of cases, I explored patterns of controls in the data. The use of a data template and pattern matching of cases in this study allowed me to make inferences of the relationship between behavior control and participants' practices, rather than deterministic and strong causal relationships between them.

External Validity

For external validity issues, Yin(1993, 2003) suggests theory based research design and the use of analytic generalization in multiple cases. Unlike quantitative study, case research is not strong enough to present theoretic generalization(statistical generalization). Thus, statistical sampling is not appropriate in case studies. Instead, analytical generalization through the investigation of multiple cases is suggested in the case research. Following Yin's suggestion, I chose a multiple case study approach.

2) http://dev.fckeditor.net/wiki/CodingStyle accessed on July 31, 2008.

Reliability

Yin(2003) advises to address reliability issues by building a database and research protocols. The database and research protocols help the researcher and others replicate the case study research. This study collected public online data using a web-crawler and stored the collected data in a database and file folders. In addition, a case study protocol was produced and revised after studying the pilot case.

3.3 Case Selection

Basic criteria were drawn from the descriptive information of open source projects provided by Sourceforge.net, which had more than one million registered members as of May 2007(when the data collection took place). Although there are many alternative systems for communications to support open source ISD projects, such as code repositories and code compilation, the number of hosted projects is comparably limited to represent general features of open source projects.[3)]

Based on the descriptive information available about Sourceforge.net projects, six projects were selected using three criteria: software taxonomy, the size of teams, and download statistics. Statistics of software downloads and the size of project teams are perceived as potential measures to determine the success of an individual open source project(Crowston *et al.*, 2004). Sourceforge.net classifies hosted projects based on a software taxonomy which includes types such as communication, database, desktop, education, etc. The case projects selected for this research had more than five developers and at least 500,000 file downloads as of May 2007. These six projects belong to system or applied(office/business) groups of hosted projects. However, five projects were actually used in data analysis since the project administrator of one project refused to allow his interview session to be used.[4)]

3.4 Data Collection and Processing

SOURCES OF EVIDENCE

Among six recommended sources of data suggested by Yin(2003), this research focused on documentation and archival records of development activities. Documentation in this research included information available on the project website, various project related documents such as FAQs, guides and manuals for users and other developers, how-tos, policies, etc. Archival records were forum messages, mail messages, announcements, and trackers of bugs, features and patches. At the same time, I collected and examined records of development activities stored in the version control system of projects(e.g., CVS, SVN: subversion). Data of development activities included number of lines of updates and commits of source code by developers, and a timeline of case activities and issues.

While reviewing the collated messages by topic, I marked relevant information on the margin space of transcripts when control practices specified in the data template were encountered. Sequences of controls were analyzed to get insight about management

3) http://www.ibiblio.org/fosphost/exhost.htm accessed on July 31, 2008; http://en.wikipedia.org/wiki/Comparison_of_free_software_hosting_facilities accessed on July 31, 2008; http://en.wikipedia.org/wiki/Portal:Free_software accessed on July 31, 2008.

4) The Claremont Graduate University Institutional Review Board approved the research approach, which informed all interviewees that they were free to drop out of the project at any time.

of open source ISD projects and structural changes over time. After reviewing the communication contents, the marked contents were assembled in a spreadsheet file for each case. For data reduction(Miles and Huberman, 1994), these spreadsheet data were further reduced into a table containing the chronicle data of control practices of each case <Figure 1>.

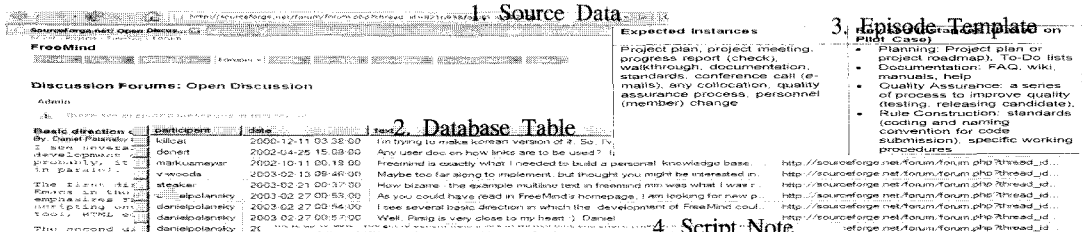
Ⅳ. Data analysis and Discussions

4.1 Overview of Case Projects(see 〈Table 2〉)

'FreeMind' is an open source mind mapper(or a cognitive mapping tool). Its features include hierarchical representation of data, basic drawing functions(lines connecting contents), and folding of contents. The project team focuses on developing 'FreeMind' as a knowledge management tool. 'FreeMind' was originally aimed at being developed as an editor, a mind mapper, a collaboration tool, and a web browser. It is the mind mapper, however, which remains as the main feature of the current version of 'FreeMind'. Users of 'FreeMind' have various contextual backgrounds and purposes: personal use, project management, system development, education, and so forth.

'FreeNAS' is an open source NAS(Network Attached Storage) server based on the free BSD operating system. It supports various network protocols(CIFS: Samba, FTP, NFS, AFP, RSYNC, SSH, Unison, and iSCSI) for data transfer over networks. It provides the features of RAID(Redundant Array of Independent Drives) 0, 1, and 5, JBOD, disk encryption, web-based configuration, and SMART(Self-Monitoring Analysis and Reporting Technology). The project team recently became a multi-developer team. The motivation of this project was to develop a NAS server for personal purposes in order to provide features not supported by existing free and commercial NAS applications.

'KeePass'('KeePass' Password Safe) is a free, multi-platform, and open source password manager work-



〈Figure 1〉 Data Collection

〈Table 2〉 An Overview of Focused Data Captured from Case Projects

Case Projects	Period of data	Total Messages(participants*)	Selected Data	Other data source
FreeMind	June 2000~May 2007	14,249(4,046 topics) by 1200 participants	2,020(220 topics), 227 participants	260 mail messages, statistics of file repository(CVS), project documents, wiki
FreeNAS	October 2005~May 2007	11,682(2,550 topics) by 1800 participants	1,317(364 topics), 264 participants	329 mail messages, statistics of file repository(CVS), project documents
KeePass	November 2003~May 2007	12,943(1,990 topics) by 500 participants	3,743(449 topics), 143 participants	Statistics of file repositories (CVS and SVN), project documents
PHPMyAdmin	June 2000~May 2007	13,402(3,000 topics) by 700 participants	3,573(1,130 topics), 301 participants	Statistics of file repository (SVN), project documents
WebCalendar	June 2000~May 2007	32,264(7,504 topics) by 2,300 participants	3788(657 topics), 349 participants	Statistics of file repository (CVS), project documents

Note) * Project involves many anonymous participants. Thus, the actual number of participants is larger than the number provided in the table.

ing on Windows-based computers. It secures users' data with the most advanced security algorithms such as AES(Advanced Encryption Standard) and Twofish. Users of 'KeePass' can manage personal information of numerous online accounts with passwords in a single secure database. Despite its many users, contributors, and long history, the project leader keeps the projects as a single developer project.

'PHPMyAdmin'(or PMA) is a tool to administer MySQL with web interfaces. The application was developed in PHP script language. The features of 'PHPMyAdmin' are bound to the MySQL database. The main usage of 'PHPMyAdmin' is found in web hosting environments by supporting an online access to database configuration over the network. Because of the close attachment to MySQL, the progress of PHPMyAdmin is tightly coupled with updates of the base application. Major tasks of PHPMyAdmin are supporting the compatibility of features in accordance with changes of MySQL and the development of its own new features. The project is recognized as a 'community project' by the MySQL AB.[5)]

'WebCalendar' is a web-based calendar application for a single or a group of users in an intranet. It also functions as a scheduler and an event calendar viewable for web visitors. It is programmed in PHP script language. Due to its high quality and usefulness, the 'WebCalendar' has been embedded in a number of open source MRBS(Meeting Room Booking System), groupware tools, and numerous personal and community web sites. The major issues and decisions of the project were open for discussion with projects participants including users. When issues such as the decision for the integration of platforms or the future direction of the project were raised, the

5) http://www.mysql.com/accessed on July 31, 2008.

project team gathered opinions of each participant and asked the participants to vote on the possible options.

4.2 Data Analysis and Discussion

<Table 3> summarizes the collected data project by project. To verify the data, instances observed in threads of messages were matched up with the data in other data sources including online documents(manuals, FAQ, or wiki) at project websites.

Overall, instances of practices of behavior controls were observed in the five cases including planning, documentation, quality assurance, and rule construction, which were specified as potential measures in the data template. Detailed practices vary project by project. This is quite contrary to the assumption of some researchers who have argued that behavioral control is infeasible in open source project management as it potentially hinders participants' autonomy on which an innovation process relies(Xu *et al.*, 2005; Nidumolu and Subramani 2003).

One distinctive finding is that FreeNAS showed relatively fewer instances of practices of behavior control. Given that control practices shown in <Table 3> actually have been gradually implemented over time, the difference seems to be attributable to FreeNAS's relatively short project history; the control strategy of FreeNAS appears to be still under development.

In terms of project planning, project teams were observed to create two types of plans: project roadmap(or future plan) and To-Do list. Project roadmaps, which were generally observable in the selected cases, presented abstract levels of future plans. Project plans, which focused on features and tasks, integrate individual developers' task determination in an auton-

〈Table 3〉 Observed Instances of Control Practices

Controls	Cases(Projects)				
Instances	FreeMind	FreeNAS	KeePass	PHPMyAdmin	Web-Calendar
Planning: Project plan(project roadmap)	To-Do List	project roadmap, To-Do list	To-Do list	No formal project roadmap, To-Do list	Future plan, To-Do list
Documentation: (manuals, FAQs, and wiki)	Manuals Change log wiki FAQs, bug list	Change log Bug report Guide for developers	FAQ Online help (no user participation)	Change log Wiki FAQ Manual(developers' documentation)	Manuals wiki FAQ
Quality assurance	RC(Releasing Candidate) testing		RC testing	RC testing	RC testing
Rule construction: Standards(code convention, industry standard) or Specific working procedures	coding convention, translation procedure, recommended tools, expression notations		translation procedure	Industry standard(W3C), Internal policy, Coding convention, Translation procedure	Industry standard(IETF RFC 2445), Translation procedure

omous manner. The existence of such autonomy, however, did not seem to hinder the functionality of planning in open source project management, and is different from traditional ISD management, which is top down rather than bottom-up. The observed comments of an administrator of WebCalendar supported this observation. He explained the relationship between software quality and autonomy by saying that unlike commercial software projects in which tasks are delegated top-down, developers in open source ISD projects choose what they want to work on, which has positive effects on development. He also noted that as developers determine tasks based on personal interests(or passions), they are likely to go an extra mile to make that component better, which is less likely to happen in traditional commercial settings.

To-Do lists were generally observed in file repositories or released packages, describing detailed tasks at the present stage(or version). They combined tasks determined by an individual developer. To-Do lists with file repositories were observed to function as a source to deliver information about the status of software under development. One frequently witnessed issue in communication threads was developers' concerns about limits of resources. Because providing a response to every inquiry and service request was practically infeasible, developers tried to structure repeating practices as shown in the inquiries of 'how to become a listed developer' and 'how to contribute.' This provides inquirers with better access to corresponding information or to be asked to read manuals or wikis instead of receiving direct guidelines. For instance, feature requests, one of the critical inquiries of open source ISD projects, were observed to be often rejected in an early stage of projects, raising complaints about administrators' methods of project management. To address such issues, open source project teams allowed participants to access file repositories(CVS and SVN) and lists of tasks(To-Do lists), which enabled them to acquire an up-to date snapshot of the project information and to decide whether their feature requests were appropriate and not-overlapping.

Documentation was another important issue in the observations of the cases. Change log, manuals, FAQs, wiki, and web pages were common forms of documents observed in various sources such as communication threads, project site, and file repositories. The observed cases involved many non-developer participants who were not necessarily to be listed as a member. Documentation was observed to be a major area where these non-developers could contribute followed in degree by localization(translation) and user support. Documentation was dealt with by developers in the very early stages of the projects, and evolved into collaboration tools(i.e., wiki) between developers and participants. Documentation seems to aim at reducing developers' workload to respond to requests of information and services.

Project documents also contain various conventions for third party developers(related to add-ons/ plug-ins architecture). Due to limited resources(developers), project teams seemed to try to balance between their capabilities and enrichment of features. The project teams implemented open architecture using add-ons and plug-ins to seek for such balance. With open architecture, developers in the team were observed to focus only on core features and source code quality while achieving functional richness through collaborations with developers who did not belong to the project teams.

Quality assurance turned out to be another important instance of behavior control observed in the data. Observed cases showed a development cycle composed of two distinctive periods: the long-term

〈Table 4〉 Procedures for Translation

FreeMind	Procedures(where to get source code, how to post the result, and how to become the listed member), recommended tools, expression notations, specification of files[6]
KeePass	Standard, procedures(how to get source code and how-to), notations, recommended tools[7]
PHPMyAdmin	Procedures(what file to translate, how to submit the processed data, and how to become the listed member) and mailing list[8]
WebCalendar	Procedures(what file to translate, detailed guidelines, and how to submit the results)[9]

period for feature development, and the short-term period for releasing packages. In the long-term feature development period, developers created new features, and fixed the reported bugs of previous versions with no time constraint. In the short-term releasing period, however, the observed cases intensively mobilized resources including users. In general, the observed project teams released a number of releasing candidates(RC) and requested users to test these. During the releasing period, three projects(KeePass, PHPMyAdmin and WebCalendar) froze development of new features and emphasized the improvement of quality of features on hand.

The selected project cases are also observed to produce guidelines and standardized procedures, to some degree, from coding conventions to the compliance with industrial standards(PHPMyAdmin for W3C standard and PHP, WebCalendar for iCalendar and IEFT RFC 2445). Communications messages regarding translation and participation into code development exemplified the use of structured guidelines and procedures in the observed cases. Translation(localization) was observed to be the most active project task following source code development. In most cases, project teams required translators to comply with a set of guidelines generally including a way to access source files, instructions, and to manipulate translated files. <Table 4> shows procedures and standards formalized in the project teams.

Guidelines of procedures sometimes included requests to comply with releasing schedules, how to cooperate with other translators for specific language, and how to get information and announcements of the project. Except KeePass, which was a single developer project, project teams had procedures to accept qualified developers as core members. Through a set of procedures, developers were required to learn project tasks, and to demonstrate their qualifications in searching and fixing bugs. A FreeMind developer put it this way: "The best method is as follows: publish your development plan in our Open Discussion forum and discuss with others, provide first patches to FreeMind from your developers directly to me, get involved as official developers some time later."

In terms of functional purposes, instances of behavioral control practices can be classified into four

6) http://freemind.sourceforge.net/wiki/index.php/Translation accessed on July 31, 2008.

7) http://keepass.info/translations_devinfo.html accessed on July 31, 2008.

8) http://www.phpmyadmin.net/documentation/#faq7_2 accessed on July 31, 2008; http://sourceforge.net/mailarchive/forum.php?forum_name=phpmyadmin-trk-translat accessed on July 31, 2008.

9) http://www.k5n.us/webcalendar.php?topic=FAQ#faq_11accessed on July 31. 2008.

sub sections: information sharing, standardization of practices, revision of source code architecture, and supporting systems.

Information sharing includes plans, daily updates of source code, project documentation, and communication. It functions to streamline information requests such as repeated requests of user support, status of incorporation of contributions(but, information sharing does not necessarily bring all participants into decisions on issues). Standardization of practices include conventions, task guidelines, and statements of qualification. Following established procedures and standards, participants seem to rely less on direct controls and commands, which enhance self-control of participants. Although details vary project by project, all observed cases showed revisions of source code architecture such as split language section, add-ons or plug-ins to comply with the evolution of team structure(related to roles of members such as translation) and users' demands for functional diversities.

In addition, the cases utilize various supporting systems. When interactions were limited, developers maintained direct contact with user participants for problem reports(i.e., bugs), user support, etc. As interactions increased, developers relied more on supporting systems and requested user participants to utilize supporting systems. Besides system components provided by the hosting service(Sourceforge.net), the observed cases adopted various systems such as a wiki and a file repository system(Subversion: SVN).

Various instances related to behavior control observed in this study are structured forms of repeating practices to handle overload due to the openness of the resources of the source code and developing activities. The observed cases have experienced increasing overload due to increased interactions over time such as requests of enrichment features, use supports, task guidelines, etc., which does not seem to be easily handled with the limited capabilities of the various projects.

V. Conclusion and Future Research

The main research question of this study is whether open source software development projects construct behavior control strategies. Through observations of longitudinal data, this research found evidence to support a positive answer. Project teams were observed to exercise various managerial practices related to behavior control as an attempt to handle overload due to the openness of the resources. This finding is against suggestions(Lane *et al.*, 2005; Xu *et al.*, 2005) that formal control, especially behavior control, is not feasible in open source projects due to potential interference between authority of participants and pre-specified guidelines.

Various practices related to behavior controls observed in this research can present possible options to open source project practitioners who experience corresponding issues or problems such as how to handle overloads. Given that there are successes and failures in open source projects, deeper investigations on behavior control practices in those projects have to be followed in further researches to provide more systematic guidelines for better management of open source ISD projects. In this regard, the research scope also has to be expanded to other types of controls, including outcome control and self-control.

It also has to be noted that this study has some limitations. First, this is a case study of which findings need further verification through more studies for general use. Second, the scope of this research is limited to only one aspect of control. Thus, it can-

not be ruled out that there are possible alternative explanations about instances of behavior control practice observed in this study. Third, the data template to capture operational measures was developed based on studies about control in traditional ISD projects. Although this research conducted a pilot study to adjust the template to open source ISD context, there can still have misinterpretation of meanings of terms and underlying contextual interrelationships among measures.

References

AlMarzouq, M., L., Zheng, G., Rong, and V., Grover, "Open Source: Concepts, Benefits, and Challenges", *Communications of AIS*, Vol.16, 2005, pp. 756-784.

Bollinger, T., R., Nelson, K., Self, and S., Turnbull, "Open source methods: peering through the clutter", *IEEE Software*, Vol.16, No.4, 1999, pp. 8 -11.

Charette, R. N. "Why Software Fails", in: *Spectrum (IEEE)*, 2005, pp. 42-49.

Choudhury, V., and R., Sabherwal, "Portfolios of Control in Outsourced Software Development Projects", *Information Systems Research*, Vol. 14, No.3, 2003, pp. 291-314.

Crowston, K., H., Annabi, J., Howison, and C., Masano, "Towards a portfolio of FLOSS project success measures." In ICSE Open Source Workshop, 2004.

Crowston, K., K., Wei, Q., Li, and J., Howison, "Core and periphery in Free/Libre and Open Source software team communications", 39th Hawaii International Conference on System Sciences, Hawaii, 2006.

Curtis, B., H., Krasner, and N., Iscoe, "A field study of the software design process for large systems", *Communications of ACM*, Vol.12, No.3, 1988, pp. 346-371.

Davenport, T. H., and L., Prusak, Working Knowledge Harvard Business School Press, Boston, Massachusetts, 1998.

Eisenhardt, K. M., "Control: Organizational and Economic Approaches", Management Science, Vol.31, No.2, 1985, pp. 134-149.

Elam, J. J., and D. B., Walz, "A Study of Conflict in Group Design Activities: Implications for Computer Supported Cooperative Work Environments", 21st Annual Hawaii International Conference on System Sciences, 1988.

Fang, Y., and D. J., Neufeld, "Should I stay or should I go? Worker commitment to virtual organizations", Hawaii International Conference on System Sciences, IEEE, Hawaii, USA, 2006.

Feller, J., and B., Fitzgerald, "A Framework analysis of the open source software development paradigm", ICIS, Brisbane, Queensland, Australia, 2000, pp. 58-69.

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*, Vol.11, 2001, pp. 277-304.

Guinan, P. J. Patterns of excellence for IS professionals: an analysis of communication behavior ICIT Press, Washington, D. C., 1988, pp. ix, 173p.

Henderson, J. C., and S., Lee, "Managing I/S Design Teams: A Control Theories Perspective", *Management Science*, Vol.38, No.6, 1992, pp. 757-777.

Hippel, E. v., and G. v., Krogh, "Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science", *Organization Science*, Vol.14, No.2, 2003, pp. 209-223.

Iannacci, F., "Coordination Processes in Open Source Software Development: The Linux Case Study", 2005.

Jarvenpaa, S. L., and D. E., Leidner, "Communication and Trust in Global Virtual Teams", *Organization Science*, Vol.10, No.6, 1999, pp. 791-815.

Jensen, C., and W., Scacchi, "Collaboration, Leadership, Control, and Conflict Negotiation and the Netbeans.org Open Source Software Development Community", 38th Hawaii International Conference on System Sciences, IEEE, Hawaii, 2005.

Kirsch, L. J., "Portfolios of Control Modes and IS Project Management", *Information Systems Research*, Vol.8, No.3, September 1997, pp. 215-239.

Kirsch, L. J., "Deploying Common Systems Globally: The Dynamics of Control", *Information Systems Research*, Vol.15, No.4, 2004, pp. 374-395.

Lakhani, K.R., and Hippel, E.v. "How open source software works:"free" user-to-user assistance", *Research Policy*, Vol.1451, 2002, pp. 1-21.

Lane, M., and P., Basnet, "Informal Control in Open Source Projects: An Empirical Assessment", 16th Australian Conference on Information Systems, Sydney, 2005.

Lattemann, C., and S., Stieglitz, "Framework for Governance in Open Source Communities", 38th Hawaii International Conference on System Sciences, Hawaii, 2005.

Manz, C. C., K. W. Mossholder, and F. Luthans, "An Integrated Perspective of Self-Control in Organizations," *Administration and Society*, Vol.19, No.1, 1987, pp. 3-24.

Markus, M. L., B., Manville, and E, A., Carole "What Makes a Virtual Organizational Work?", *MIT Sloan Management Review*, Vol.42, No.1, 2000, pp. 13-26.

McConnell, S., "Open Source Methodology: Ready for Prime Time?", *IEEE Software*, Vol.16. No.4, 1999, pp. 6-8.

Miles, M. B., and A. M., Huberman, An Expanded Sourcebook: Qualitative Data Analysis SAGE Publications, Thousand Oaks, London, New Delhi, 1994.

Nidumolu, S. R. and M. R., Subramani, "The matrix of control: Combining process and structure approaches to managing software development", *Journal of Management Information Systems*, Winter 2003.

Ouchi, W. G., "A Conceptual Framework for the Design of Organizational Control Mechanisms", *Management Science*, Vol.25, No.9, 1979, pp. 833-848.

Ouchi, W. G., "Markets, Bureaucracies, and Clans", *Admin. Science Quarterly*, Vol.25, No.1, 1980, pp. 129-141.

Paulson, J. W., Succi, G., and Eberlein, A. "An empirical study of open-source and closed-source software products", *IEEE Transactions on Software Engineering*, Vol.30, No.4, APR 2004, pp. 246-256.

Pearlson, K. E. and C. S., Saunders, Managing and Using Information Systems: A Strategic Approach, (2nd ed.) John Wiley and Sons, New York, NY, 2004.

Piccoli, G., A., Powell, and B., Ives, "Virtual team: team control structure, work processes, and team effectiveness", *Information Technology and People*, Vol.17, No.4, 2004, pp. 359-379.

Robey, D. and D., Farrow, "User Involvement in Information-System Development a Conflict Model and Empirical-Test", *Management Science*, Vol.28, No.1, 1982, pp. 73-85.

Raymond, E. S., The Cathedral and The Bazaar O'Reilly and Associates, Inc., Sebastopol, CA,

USA, 1999.

Townsend, A. M., S. M., DeMarie, and A. R., Hendrickson, "Virtual teams: Technology and the workplace of the future", *IEEE Engineering Management Review*, Vol.28, No.2, 2000, pp. 69-80.

Xu, B., Y., Xu, and Z., Lin, "Control in Open Source Software Development", 11th American Conference on Information Systems, Omaha, 2005.

Yin, R. K., Applications of Case Study Research, Thousand Oaks, London, New Dehli, 1993.

Yin, R. K., Case Study Research, Design and Methods, (3rd ed.) SAGE, Thousand Oaks, London, New Dehli, 2003.

Information Systems Review
Volume 10 Number 2
August 2008

Open Source Project Management-from a Behavior Control Perspective

Jun Gee Cho*

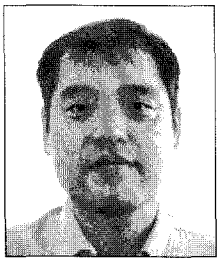
Abstract

With the successful arrival of quality free/open source software, open source ISD(Information System Development) projects have been drawing attention from academic researchers. However, there have been few efforts to examine the managerial aspect of open source ISD projects. This study use a case research methodology to explores the management of open source projects, especially from the perspective of behavior control. Through data analysis and discussion, the study found that various practices related to behavior control were exercised to obtain participants' shared goals, although specific instances vary in each case. This finding is contradictory to the views of some researchers who suggested open source ISD projects lack behavior control. For more comprehensive understanding, however, future research should also includecontrols of open source projects in terms of outcome control and self-control.

Keywords: Open Source, Control, Behavior Control, Project Management, Case Study

* Claremont Graduate University, Claremont, California, USA

◐ 저 자 소 개 ◑



조 준 기 (jun.cho@cgu.edu or jun.cho@gmail.com)

연세대학교 공과대학 금속공학과 졸업(공학학사)미국 템플대학교 대학원 MBA/ MS in eBusiness 졸업(경영학석사) 미국 클레어몬트대학 대학원 Information Systems and Technology(박사, 2008년 8월 졸업예정)

논문접수일 : 2008년 06월 19일 **게재확정일 : 2008년 08월 02일**
1차 수정일 : 2008년 07월 27일