

COMMUNITY-BASED OPEN SOURCE

The Phenomenon and Research Opportunities

Manlu Liu and Qiang Tu

Rochester Institute of Technology, New York, U.S.A.

Keywords: Community source, System development, Collaboration, Virtual team, Kual.

Abstract: Community-based open source (community source) development has emerged as a new way of developing enterprise applications, leading to a unique type of open source practice involving collaboration from multiple organizations in a virtual environment. In this paper, we introduce the concept and motivation of community source and address several research directions in community source under multiple perspectives. We examine a real world case, the Kual community source project, to help better understand this new phenomenon. The long term research objectives are discussed at the end of the paper. This paper facilitates a general understanding of the emerging community source development landscape. We believe that the issues presented in this paper can attract more researchers to study this new area and help organizations make better decisions in IT investments.

1 INTRODUCTION

Due to an increasingly competitive business environment, organizations demand customized application software that can meet their specialized and strategic requirements. Sometimes organizations feel compelled to develop the software in-house since they could not find the suitable application software in a commercial market. According to the study of (Perry et al. 2007), the overall cost of in-house development is so much higher than that of buying commercial software that many organizations cannot afford the former approach. Furthermore, it is impractical for many organizations to achieve all the necessary competence for building software themselves. As a result, they seek out strategic alliance to jointly develop their desired software. Strategic alliance is a formal relationship between two or more parties to pursue a set of agreed upon goals or to meet a critical business need while remaining independent organizations (Yoshino et al. 1995). A new approach of enterprise application software (EAS) development is applying strategic alliance to the development mechanism of open source software (Agerfalk et al. 2008). The approach is referred to as community-base open source or "community source" (Wheeler, 2004). In community source, a community of partner organizations invests and develops the EAS and this

software is open source eventually.

The community source is a unique form of open source approach. It is somewhere between commercial software and open source. As described by Raymond (2001), when developing commercial software, vendors are organized as builders constructing cathedrals based on blueprints; but on the other hand, open source software is built when programmers work in a way similar to merchants in a marketplace, harnessing a range of approaches and agenda, taking input from diverse types of people scattered across the world, and being open to new ideas and new participants. A community source organization can be described as a shopping mall where a virtual organization consisting of participating institutions, each of which commands its own employees. Community source creates a formal virtual organization where multiple partner institutions collaborate with each other to develop custom software solution in different locations and at different time.

In community source, rather than relying on commercial software vendors, the development partner institutions pool their resources together to develop EAS. This pooling of resources gives the organizations better control over the software development and community source thus offers an innovative approach to developing open-source EAS.

By becoming a development partner in community source, the institution can influence the application features, complete the deployment sooner, minimize the total cost of ownership, and reduce the uncertainty of the system value. This incentive is the main driving force for institutions to make community source projects succeed. The motivation for institution joining in community source development will be further explained in the Kuali case section.

It is worthwhile to study community source since it offers an innovative way to develop enterprise applications and it can potentially break the dominance of commercial software in enterprise applications.

This paper is structured as follows: we start with introducing a real world community source project to help better understand the community source approach. Then we outline the related research directions under the technological, the economic and the managerial perspective respectively and discuss the potential research opportunities. We present the future trend of community source and the value of research in community source in the discussion section. The long term objectives for community source research are pointed out at the end of this paper.

2 THE KUALI CASE

In order to better understand community source, we studied a real-world ongoing community source project of called Kuali (www.kuali.org). The Kuali case offered us a great opportunity to study the research issues of community source in a higher education setting.

Kuali is a consortium of universities to develop an open source financial service system, starting with the conversion of the Indiana University's Financial Information System to the web in 2004 (www.kuali.org). The original motivation of the project is that existing financial systems used in the universities are outdated and too difficult to maintain. The commercial products are often too expensive and hard to customize; some institutions paid tens of millions to ERP vendors for software and installation, but still need to build 15% of all the features they need to handle specific financial transaction needs. Some schools and colleges found that they need to operate expensive "shadow systems" to provide needed features that are absent in currently available ERP packages. The alternatives to buying off-the-shelf packaged

financial software are equally daunting and can only be considered by the largest institutions. The Kuali project provides an attractive alternative to the "buy or build" dilemma. It pools institutional resources to develop an open source financial system, thus dramatically reducing the cost of managing fiscal data and processes in higher education.

The initial mission of the Kuali consortium was to develop a baseline system for financial services and has now expanded to other systems such as research administration and student management. Currently, there are three main application modules in Kuali: the Kuali Financial Services (KFS), the Kuali Research Administration (KRA), and the Kuali Student System (KS). The KFS project was just completed. The KRA and KS project are scheduled to be released after 2010.

The Indiana University and the University of Hawaii started to lead the effort to build KFS in 2004. In March, 2005, Kuali project got a start-up grant \$2.5M from the Andrew W. Mellon Foundation. The follow-up funding was invested from participating institutions which are referred as the development partners of KFS. KFS project is mainly funded by its partner institutions. In April 2006, Kuali project announced the availability of the Kuali Test Drive, which enables institutions to explore KFS. In October, 2006, Kuali foundation released KFS phase I. KFS Phase II was released in July 2008. KFS project was completed in 2010 and have grown to around thirty development partners including Carnegie Mellon University, Cornell University, Massachusetts Institute of Technology, the University of Arizona, and University of California. One of the very important components of the Kuali effort is the Kuali Commercial Affiliates program. Commercial Affiliates provide for fee guidance, support, implementation, and integration services related to the Kuali software. Affiliates may offer packaged versions of Kuali that provide value for installation or integration beyond the basic Kuali software. Affiliates may also offer other types of training, documentation, or hosting services.

The development team in community source is under a formal organizational structure. Figure 1 shows the complexity of the organization structure. The development partners work in a project organization. The Kuali Board is the final decision maker during the development of Kuali. The Kuali functional council and Kuali technical council take care of the functional issues and technical issues of Kuali, respectively. The Extended board, Kuali investors, Kuali functional council, Kuali technical council and the project manager report to the Kuali

board. The project manager supervises the project staff and development staff. The project manager is managing an organization that is virtual in every aspect. The task of the project manager is to complete a project of size and complexity on time and on budget.

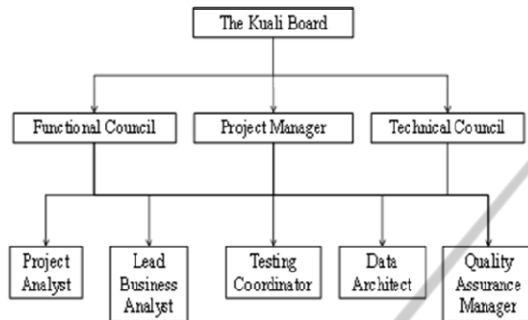


Figure 1: KFS Organizational Structure.

In order to ensure flexibility and extensibility, Kualu uses the most up-to-date approaches and technologies such as open source, workflow, and service-oriented architecture.

As we can see from this case study, the Kualu project provides an attractive alternative to the “buy or build” dilemma. The Kualu project pools resources from multiple universities together to develop an open source system which can meet universities’ special requirements, and finally reduces costs and minimizes risks. In the Appendix, we compare community source with the other two main approaches: traditional open source and commercial software and summarize their characteristics. Comparing with the traditional open source approach, Kualu has better system quality control since the system is developed under a formal organization and the system adopters could have strong technical support due to the commercial affiliates in Kualu. Comparing with commercial software, Kualu satisfies the special requirements for the high education and dramatically reduce the development costs and development time by sharing resources among multiple institutions.

3 RESEARCH DIRECTIONS

In a community source project similar to Kualu, multiple institutions make significant investment in developing the system. Development teams acquire, maintain, analyze, and formalize software requirements from multiple stakeholders that can have varying and possibly conflicting needs. The

research directions in community source could be generated from multiple perspectives.

We outline several research issues for community-based development of enterprise applications under three perspectives: the technological perspective, the economic perspective and the managerial perspective. We discuss unique research opportunities along with those research directions.

3.1 The Technological Perspective

We address two research issues under technological perspective, including technology flexibility, workflow modelling paradigms.

3.1.1 Technology Flexibility

Community source is a unique form of open source in which the community must balance the various, sometimes even conflicting, requirements from all development partners. As such, community source is more challenging than a general open source project in which variations in requirements are not a major issue (Liu, Wang and Zhao, 2007). In our view, community source has more stringent requirement in flexibility in order to deal with complex requirement analysis and change management. Recently emerged technologies such as web services, service-oriented architectures, and workflow automation can help make a system more customizable. For example, Kualu takes advantage remarkably of these emerging technologies to enable the strong system flexibility. In fact, we believe that community source is a good representative of a recent trend in combining service centricity with open source development to build scalable and flexible information systems. As such, how technology flexibility is achieved through service-centric community source becomes a very interesting research issue.

3.1.2 Workflow Modelling Paradigms

Currently, the Kualu platform specifies workflows by means of workflow rules in the context of financial documents. As such, these workflow rules are also called routing rules. These routing rules seem to work fine in the context of financial services since the workflows are relatively simple with typically three to five steps. However, it is not clear if routing rules would be appropriate in the modules which consists of many processes (Liu, Wang and Zhao, 2007). Specifying these workflows using routing rules could result in lengthy XML scripts that are

very difficult to write and maintain. In addition, the control flow of those processes is not explicitly modeled when using rule-based and document-driven workflow modeling, which makes it difficult to understand the underlying process logic. This raises a research issue on when a particular workflow modeling paradigm should be used.

3.2 The Economic Perspective

The community source approach provides a viable alternative to vendor-provided packaged solutions by combining effectively the benefits of in-house development and outsourcing. One critical decision for prospective community source partners is whether or not it should pay a significant amount of partnership fee to join the development community, which we refer to as “the community source investment decision” or “the community partnership decision” (Liu, Zeng and Zhao, 2008). As in general, the resulting software package of a community source project is available as an open source free of charge, it seems counter-intuitive for an organization to be willing to make major investments. As shown in many real-world examples, many organizations have already invested heavily in various kinds of community source projects. This poses an interesting and timely research challenge as to gaining an in-depth understanding of economic incentives behind these decisions and further developing an actionable decision framework for the community partnership decisions.

3.3 The Managerial Perspective

In community source, multiple institutions collaborate closely and work in a virtual environment to develop the system. Eventually the system needs to meet the requirement from multiple institutions. We can see that project management issue becomes a critical issue of developing the system successfully in community source.

We address two research issues under managerial perspective: project success, outsourcing.

3.3.1 Project Success

In community source, multiple institutions make investments and develop systems together. The whole development process requires close collaboration among partner institutions. The developers in community source are full time employees in partner institutions, who are designated to work for the community. The

community source organization is a virtual organization consisting employees from the partner institutions. Community source development is therefore very different from in-house software development and traditional open source development. Achieving project success in community source poses new challenges that are not found in the development of commercial software or conventional open source software such as Linux (Liu, Wheeler and Zhao, 2008).

Since community source is a new approach of system development, understanding how to achieve project success in community source development is a very interesting research topic. However, the related issues have not been well studied in the literature. Several key research questions motivate our work: (1) which framework can be used as a theoretical basis for studying project success in community source development? (2) What factors can be identified within the theoretical framework? (3) How do these factors affect project success in community source development?

3.3.2 Outsourcing

In community source, rather than rely on commercial software vendors, the user organizations pool their resources together to develop EAS. This pooling of resources gives the organizations better control over the software development and community source thus offers an innovative approach to developing open-source EAS. Developers in a community source project are employees of the partner institutions who are designated to work on the project. Due to this unique feature, however, community source projects face a number of challenges. On one hand, it is optimal for the community source project to attract as many participating organizations possible in order to share resources, reduce costs, and minimize risks. On the other hand, the management of community source project becomes increasingly complicated and difficult with the increasing number of participating institutions.

One of the solutions to those issues might be global outsourcing of software development in the community source project (Liu et al., 2010). The primary research questions of this study could be: What are the motivations for global outsourcing of software development in community source, and what are the potential benefits of global outsourcing of software development in community source?

4 DISCUSSIONS

The greatest value of community source is the leveraging of resources of the partners and the community for shared value creation. In the past, when Cornell University spent \$500,000 for some system, the investment provided no advantage to San Joaquin Delta College; Or when Rutgers University developed a clever piece of cross-language middleware code, Indiana University did not benefit; When Cambridge University developed a teaching tool, U.S. and other institutions gained little (Wheeler, 2007). The community source approach changes the situation in which organizations invest huge money in IT but do not or seldom share among each other. All of organizations in community source are mutually using other people's money to get and sustain the systems they need. Community source are foremost a coordination mechanism for institutional, corporate, and personal investments of resources, ideas, and talent.

Although community source faces lots of uncertainty and might not become popular in the long term for some reason. For example, the commercial software might decide to lower the price due to the competition from the community source. However, as long as community source makes the impact to the society, the research in community source has the unique value to the application software development and the result of studies will make contribute to both theory and practice. The collaborative approach to solving problems faced by institutions with an open source license has the highest probability of benefiting the most peoples and organizations. The essence of community source approach is that the institutions develop the collaborative capability that extracts the greatest value from open source.

5 LONG TERM OBJECTIVES

Since community source becomes a new approach for enterprise application development, there is a clear need for a theoretically rigorous and practically relevant framework for understanding value create and recreate through enterprise application development under community source environment. The long term research objectives should be proposing the enterprise application development cycle as a collaborative capabilities theory for measuring, predicting, and understanding an

institution's ability to create value for community through enterprise application development. Eventually, a generic framework should be developed for describing the process of acquiring application software via either commercial, open source, or home grown approach.

REFERENCES

- Agerfalk, P. J., Fitzgerald, B., 2008. Outsourcing to an Unknown Workforce: Exploring Open sourcing as a Global Sourcing Strategy. *MIS Quarterly* (32:2), June, pp 385-409.
- Liu, M., Zeng, D. D., Zhao, J. L., 2008. A Cooperative Analysis Framework for Investment Decisions in Community Source Partnerships. *Proceedings of the Americas Conference on Information Systems*, August 14 - 17, Toronto, Canada.
- Liu, M., Wang, H. J., Zhao, J. L., 2007. Achieving Flexibility via Service-Centric Community Source: the Case of Kuali. *Proceedings of the Americas Conference on Information Systems*, August 9-12.
- Liu, M., Wheeler, B., Zhao, J. L., 2008. On Assessment of Project Success in Community Source Development. *Proceeding of International Conference on Information Systems*.
- Liu, M., Wu, X., Zhao, J. L., Zhu, L., 2010. Outsourcing of Community Source: Identifying Motivations and Benefits. *Journal of Global Information Management*, 18(4).
- Perry, R., Quirk K., 2007. An Evaluation of Build Versus Buy for Portal Solutions. *IDC White paper*, sponsored by IBM, December.
- Raymond, E., 2001. *The Cathedral and the Bazaar*. Sebastopol, CA: O.Reilly Media., Inc..
- Wheeler, B., 2004. The Open Source Parade. *EDUCAUSE Review*, pp. 68-69.
- Wheeler, B., 2007. Open Source 2010: Reflections on 2007. *EDUCAUSE Review*, pp. 49-52.
- Yoshino, M. Y., Rangan, U. S., 1995. *Strategic Alliances – An Entrepreneurial Approach to Globalization*. First Edition. Harvard Business School Press.

APPENDIX

Comparison among Community Source, Traditional Open Source and Commercial Software

	Community Source	Traditional Open Source	Commercial Software
Examples	Kuali	MySQL	SAP
Builders	Organized teams of formal employees from multiple partner institutions	Loosely connected volunteer developers	System development departments in software companies
Users	Development partners, deployment members, and others	Organizations or individuals who use the software for free	Organizations or individuals who pay for the software
Characteristics	The application development requires initial investments which are shared by the partners; the result is open source.	The software is free for adopters; the developers are volunteers.	The users need to buy the software; dedicated investment is required to develop the software.
Initial investment	Cost sharing among development partners	Low	Purchasing price
Customization	The system is designed to meet the requirements of development partners	The system is designed to meet some generic requirements	Standard package
Quality control	Strong	Weak	Strong
Development and deployment cycle time	Short: system is design to meet user requirements in the first place; developers from partner institutions are well trained from the development process.	Long: due to uncertainties in development process and required system customization after adoption.	Long: due to required system customizations after adoption.
Technical support after adoption	Strong	Weak	Strong