COMMUNITY SOURCE AS A NEW APPROACH TO ENTERPRISE APPLICATION DEVELOPMENT: EXPLORATION OF TECHNOLOGICAL AND MANAGERIAL ISSUES

by

Manlu Liu

____

A Dissertation Submitted to the Faculty of the

Committee On Business Administration

In Partial Fulfillment of the Requirements

For the Degree of

DOCTOR OF PHILOSOPHY

WITH A MAJOR IN MANAGEMENT

In the Graduate College

THE UNIVERSITY OF ARIZONA

2010

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Dissertation Committee, we certify that we have read the dissertation

prepared by Manlu Liu

entitled "Community Source As A New Approach To Enterprise Application Development: Exploration of Technological and Managerial Issues"

and recommend that it be accepted as fulfilling the dissertation requirement for the

Degree of Doctor of Philosophy

____ Date: 12/21/2009
J. Leon Zhao

____ Date: 12/21/2009
Jay F. Nunamaker

____ Date: 12/21/2009
Daniel D. Zeng

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

_________ _________________________________ Date: 12/21/2009
Dissertation Director: J. Leon Zhao

STATEMENT BY AUTHOR

SIGNED：Manlu Liu

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Dr. J. Leon Zhao, for his excellent guidance, care, and patience. He exposes me to a variety of interesting research topics and inspires me throughout every state of my research. I'm thankful for all he has taught me.

My thanks also goes to the members of my dissertation committee. Dr. Jay F. Nunamaker encourages me to see my research in a new light. Dr. Daniel D. Zeng always provides invaluable advice and strong support to my research. I would like to thank my minor advisor Dr. Keith G. Provan for his advice and guidance in my minor studies in Management &Organization.

I would like to thank all the faculty members and staff in the department; especially I would like to thank Dr. Paulo Goes for his care and support. I also would like to thank Dr. Brad Wheeler and his colleagues from the Kuali Foundation for their support of my research.

I would like to thank all of my classmates and my friends. The past five-year was full of their support and friendship.

Last but not least, I would like to gratefully thank my grandma, my parents and my younger brother for their unconditional love through this adventure. I would like to thank my husband (Saiwu), and my daughter (Jocelyn) for their understanding, support, smiles, kisses and love.

DEDICATION

This dissertation is dedicated

to my grandma

And

to my parents

TABLE OF CONTENTS

TABLE OF CONTENTS - CONTINUED

TABLE OF CONTENTS - CONTINUED

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Community-based open source (community source) has emerged as a new approach of developing enterprise software systems that requires participation and investments from partner institutions. This new approach provides the opportunity for institutions to pool resources together to achieve objectives that are hard to reach individually.

In my dissertation, community-based development of enterprise applications is examined in a real world project called Kuali through the following three perspectives: technology perspective, economic perspective and management perspective. Under the technology perspective, the main research question is “how do service-oriented architectures enable technology flexibility in community source?” The Kuali case provides the initial context for understanding the basic concepts and insights surrounding the technological issues in community source. Under the economic perspective, the main research question is “why are institutions interested in investing in community source?” The case study and analytical modeling are used to better understand the decision-making process in community source. Under the management perspective, two main research questions are studied. The first research question is “what are the factors affecting the project success in community source development?” The case study, the surveys and the interviews are conducted to test the community source project success model. The second research question is “what are the motivations for outsourcing software development in community source?” Based on the results of our interview analysis, we propose a

research framework for community source outsourcing.

In the long term, I would like to develop a generic framework for describing the process of acquiring application software via either commercial, open source, or home grown approach. My research will concentrate on comparing these three application development approaches with a special focus on community source.

The main contribution of my dissertation is to provide guideline for strategic planning and decision making in an institution. The finding in this dissertation research will provide significant insights on the selection of various software development approaches that are useful for both researchers and practitioners.

CHAPTER 1. INTRODUCTION

1.1. Community Source

Due to an increasingly competitive business environment, organizations demand customized application software that can meet their specialized and strategic requirements. If they cannot find suitable software in a commercial market, organizations feel compelled to develop the software in-house. According to the study of Perry et al. (2007), however, 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 alliances 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 a community-base open source, or “community source.” In community source, a community of partner organizations invests in and develops the EAS and this software is open source eventually.

A community source project is “an open source project that requires significant investments from institutional partners in both human resources and cash contributions” (Liu, Wang and Zhao, 2007). The community source approach is a unique form of open source approach. There are several important distinctions between the two. First, Community source requires a formal collaboration among multiple institutions via a virtual organization whereas the latter does not require so. Second, the application developers in a community source are employees designated to the project by the partner institutions. Third, the community source project requires significant initial investments by the partner institutions, but an ordinary open source project might not. Some of these three features might appear in other types of open source projects, but a community source project requires all of them. In summary, we define community source as a unique type of open source that depends on significant financial and other support from a community of institutions in the development and deployment of the enterprise application.

Community source is somewhere between commercial software and open source. As described by Raymond (2001), when developing commercial software, on one hand, vendors are organized as builders construct cathedrals; on the other hand, open source software is built when programmers work in a way similar to merchants in a bazaar, harnessing a range of approaches and agendas, 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 solutions in different locations and at different times.

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, designated to work on the project. It is worthy 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.

1.2. The Kuali Case

In order to better understand community source, we studied a real-world ongoing community source project 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.

1.2.1. Background of the Kuali

Kuali is a community source project to develop a comprehensive suite of administrative software that meets the needs of all Carnegie Class institutions. The word Kuali is the Malaysian term for “a humble utensil which plays the most important role in a successful kitchen and is used for frying, steaming, braising, blanching and many more cooking techniques and styles.” The Kuali project started with migrating Indiana University’s financial information system to a web-based platform in 2004. Indiana University and the University of Hawaii led the effort to build educational software under Kuali. Its partner institutions are colleges, universities, and interested organizations that share a common vision of open, modular, and distributed systems for their software requirements. There are currently around thirty development partners in the Kuali project including Carnegie Mellon University, Cornel 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 motivation behind the Kuali project is that existing financial systems used in universities are outdated and too difficult and time-consuming to maintain. The commercial ERP products for academic use are often excessively expensive and hard to be customized. The Kuali project provides an attractive alternative to this “buy or build” dilemma. The project pools institutional resources to develop an open source financial system, and finally reduces costs and minimizes risks.

The initial mission of the Kuali project was to develop a baseline system for financial services. This mission has later been expanded to include other systems such as research administration and student management. Kuali Financial System (KFS) phase 2 was released in November 2007. Kuali Research Administration (KRA) is under development. Kuali Student Management (KS) was was just launched. In this paper, KFS project is used to conduct an empirical study to examine the project success in community source since KFS is almost completed.

1.2.2. Project Mission and Scope

The KFS project vision is to provide and maintain a richly featured financial system for use by its member institutions. The Kuali consortium will work to ensure the following criteria: The baseline of the new system is FIS; Its financial system modules meet GASB and FASB standards; the system enables a strong control environment; thoughtful and timely changes are made to keep with advances in both technology and business; No one member bears the bulk of the cost for, or reaps the overwhelming majority of the benefit/profit from system development; An efficient governance and administrative structure is created and maintained to support the member institutions with new or improved functionality, fixes, and service releases.

The project mission is to create a comprehensive suite of functionality to serve the financial system needs of all Carnegie Class institutions. The design will be an enhancement of the proven functionality of Indiana University's Financial Information System (IU FIS).

The project scope is Kuali Financial Systems release 1 and 2, which will be based on the existing FIS design. Conformance with FIS provides the fundamental basis for Kuali scope control.

1.2.3. Project Funding

Even though it received a start-up grant from the Andrew W. Mellon Foundation, the

Kuali project is mainly funded by partner institutions. There are two types of partners in Kuali: development partners and deployment partners. A development partner pays a partnership fee to join the project during the development stage and become involved in the system development. A development partner must contribute resources to the project at a value of a half million to one million dollars, consisting of 25% cash and 75% personnel costs. If an institution joins as a deployment partner, the institution needs to pay $25,000 per year to become involved in the deployment and customization efforts. Since Kuali is an open source project, institutions can decide not to join the project at all but simply implement and use the system for free.

1.2.4. Project Structure

A community source organization was created for the project and all development partners work together in the organization (Figure 1.1). The organization consists of the Kuali board, the functional council, the technical council, the project manager and staffs. The Kuali board is the final decision maker during the development stage of Kuali systems. The functional council and the technical council are responsible for the functional issues and technical issues of the Kuali organization, respectively. The Functional Council is composed of two or more user representatives from each of the partner institutions, but each institution has only one vote. The Functional Council meets via an audio telephone link for one hour every week. Every three months they have a face to face meeting for about two and a half full days. In these meetings they deal with issues of prioritizing functionality, doing a review of the resources, identifying where they are short on resources, and what functionality gets cut. The Functional Council makes all of those decisions. The Technical Council is composed of one representative from each of the partner institutions. The role of the technical council is to provide and enforce technical standards for the Kuali Financial Systems project. Because the systems development is spread out over the IT organizations of geographically dispersed institutions, each with its own development culture and practices, without uniform standards the resulting system would be difficult to support and maintain. The project

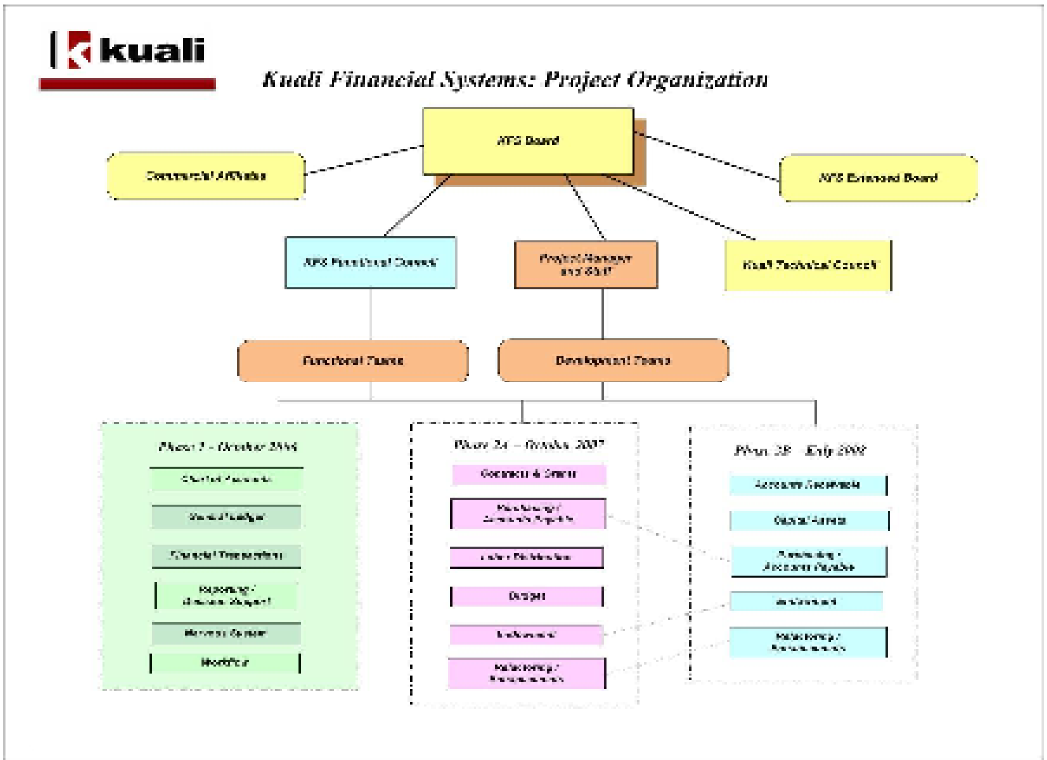

Figure 1 1 Project Organization for Kuali Financial Systems

manager is located at Indiana University, but his Kuali staff of developers is spread out over different locations. The developers are employees of and are paid by their own institutions. 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 within a certain time and on a certain budget. The Functional council, technical council and project manager all report to the Kuali board.

1.3. The Research Questions and the Research Methodologies

In this dissertation, community-based development of enterprise applications is examined through the following three perspectives: technology perspective, economic perspective and management perspective. The long term objective of this research is to develop new concepts and investigate issues in the development of enterprise applications through collaborative use of information technologies.

In particular, I am investigating issues in technology flexibility, IT investment, project success and outsourcing motivation. I have used several research methodologies including case studies, analytical modeling, interviews and survey as illustrated in Table 1.1 below.

	Technology Flexibility	Investment Decision	Project Success	Outsourcing Motivation
Case Study	X	X	X	X
Analytical Modeling		X		
Interviews	X	X	X	X
Survey			X	

Table 1 1 Multiple Research Methodologies

1.3.1. The Technology Perspective

Under the technology perspective, technology flexibility issue is studied. The main research question is “how service-oriented architectures enable technology flexibility in community source.” The Kuali case provides the initial context for understanding the basic concepts and insights surrounding the technological issues in community source. Several hypotheses are generated based on a case study and a number of interviews (Liu, Wang and Zhao, 2007).

1.3.2. The Economic Perspective

The IT investment decision issue is the main focus under the economic perspective. The main research question is “why institutions are interested in investing community source.” We have done a case study and several interviews to better understand the decision making process in community source. The Black-Sholes Model is used as the utility function in the decision model. We have developed a game theoretic model to describe the interactive behavior of decision making of multiple institutions (Liu and Zhao, 2007, Liu, Zeng, and Zhao, 2008)

1.3.3. The Management Perspective

Project success issue and outsourcing management issue are the two main focuses under the management perspective. The first research question is “what are the factors affecting the project success in the community source development.” Again, we did a case study to describe the project development in Kuali. The project success model in community source is built on the well-known Technological-Organizational-Environment framework (Liu, Wheeler and Zhao, 2008, Liu and Zhao, 2008). The pilot study is conducted to test the community source project success model. The second research question is “what are the motivations for outsourcing software development in community source?” Based on the results of our interview analysis, we propose a research framework for community source outsourcing.

1.4. The Main Contributions

The main contribution of my dissertation is to provide guideline for strategic planning and decision making in an institution. The finding in this dissertation research will provide significant insights on the selection of various software development approaches useful for both researchers and practitioners.

1.5. The Dissertation Structure

This dissertation is organized around the four research questions under technology perspective, economic perspective and management perspective and the corresponding research methodologies. Chapter 2 presents the literature review on the following related areas: open source/community source, technology flexibility, service oriented architecture, workflow technology, decision analysis, option pricing model, the Technology-Organization-Environment Framework, research dependency theory, IS success and outsourcing software developments. Chapter 3 discusses technology flexibility in community source under the technology perspective. In this chapter, I describe how service centricity plays the important role in community source and the five components which enables flexibility in Kuali. Several propositions are generated by observing the Kuali case. In chapter 4, I examine the investment decision issue in community source under the economic perspective. A cooperative analysis model is set up and the payoff analysis for joining or not joining the community source as a development partner is conducted. Several propositions are generated based on the Kuali case and the analytical modeling. Chapter 5 and chapter 6 study the two research issues – project success and outsourcing motivations of community source under the management perspective. Chapter 5 examines the factors affecting the project success in community source development. The research model with four contexts is generated based on the Technology-Organization-Environment Framework and the hypotheses are proposed. The quantitative analysis and qualitative analysis are conducted to primary test the research model and hypotheses. The main findings are summarized in the end. In chapter 6, I identify the drivers and potential benefits of global outsourcing by using a case of the Kuali community source project, which could be a viable solution for the challenges faced by community source. Based on the results of our interview analysis, we propose a research framework and related propositions. Our empirical analysis has led to interesting insights that would warrant further more extensive studies on the relationship between community source and outsourcing. This dissertation concludes with the discussion of community source and the long term objectives of this research in chapter 7.

CHAPTER 2. LITERATURE REVIEW

The literature review has ten sections. The first section covers literature in community source and open source. The literature for the technology perspective is covered by the second section which includes technology flexibility, service-oriented architecture and workflow technology. The third section reviews the literature of the economic perspective including decision analysis, option pricing model. The literature of the management perspective is surveyed by the fourth section including the technology-organization-environment framework, resource dependency theory, IS success and outsourcing software development.

2.1. Community Source/Open Source

In the past decade, open source software (OSS) development has often been characterized as an innovative way to develop software (Dibona et al. 1999; Raymond 1999). It poses a serious challenge to the commercial software businesses that dominate software markets today (Vixie 1999). The challenge from open source is not the same posed by a normal commercial software competitor that operates according to the same rules but threatens to do it faster, better, cheaper by opening the source code developed by thousands of volunteers. The open source software defies the basic motivations, economics, market structure, and philosophy of the institutions that develop and market software (Mockus et al. 2002). More recent work on individual open source developers has analyzed the relationship between motivation and participation (Bagozzi et al. 2006; Roberts et al.

2006) and performance (Roberts et al. 2006, Hann et al. 2004) as well as the alignment of license choice to developer motivation (Stewart et al. 2005). Baldwin et al. (2006) have modeled participation in open source software (OSS) communities using private provision to public goods' models where contributing towards an open source project can be optimal even if contributors only maximize their own realized value of the software. Feller et al. (2000) studied the OSS development paradigm and derived a framework for OSS.

Community source becomes the most recent trend in the development of scalable and flexible information systems collaborated by multiple organizations (Liu, Wang and Zhao, 2007; Liu, Zeng and Zhao, 2008). Community source is a type of open source because it tends to make the source code of the resulting information system available to the public. Application software for higher education appears to be the next arena for open source efforts (Wheeler, 2004). Wheeler (2007) argued that developing a collaborative capability is not an option; it is a necessity for effective college and university IT organizations. In this paper, we study specifically a higher education community source case – Kuali Foundation.

2.2. Literature of the Technology Perspective

2.2.1. Technology Flexibility

Flexibility has emerged as a key competitive factor in many organizational endeavors. Nelson et al. (1997) defines technology flexibility as "the ability to adapt to both incremental and revolutionary change in the business or business process with minimal penalty to current time, effort, cost, or performance." A measurement framework for technology flexibility was proposed including such factors as modularity, change acceptance, and consistency in the structural flexibility dimension, and rate of response, expertise, and coordination of actions in the process flexibility dimension. Zhao (1998) differentiated between two software flexibility concepts: system adaptability and system versatility. The former is defined as the capability to modify the system to cope with major changes in business processes with little or no interruption to business operations, and the latter is the capability of the system to allow flexible procedures to deal with exceptions in processes and procedures. Deiters, Goesmann, and Löffeler (2000) dealt with the issue of flexibility in technical, organizational, and human perspectives. They classify flexibility into four dimensions: process flexibility, inter-organizational flexibility, flexible management and knowledge, and flexible task allocation. Byrd et al. (2000) focused on the IT infrastructure flexibility and attempted to develop a valid and reliable instrument for IT infrastructure flexibility.

Our work under the technology perspective extends the literature by examining the relationship among three related characteristics, namely technology flexibility, system extensibility, and software customizability in community source. Essentially, we find that technology flexibility is a continuum of two stages of implementation. At the lower level, software can be customized without making major changes to the system code, leading to software customizability, and at the higher level, the system can be extended by either modifying the code or by adding new components. The key issue is about how emerging technologies such as service centric computing and business process automation can facilitate those capabilities.

2.2.2. Service Oriented Architecture

One of the emerging technologies used in community source is service-oriented architecture, which enables better system extensibility. This is a critical feature that makes community source a viable development methodology. Service Oriented Architecture (SOA) was first proposed by Schulte and Natis (1996). They specified SOA as "a style of multi-tier computing that helps organizations share logic and data among multiple applications and usage modes." SOA is commonly defined as a software architecture that allows the use of loosely-coupled software services to meet the needs of business processes and users (en.wiki.org). A service-oriented architecture defined by IBM contains four core components including business services, integration services, enterprise service bus, and infrastructure services, which work together to provide the capability for on-demand business (Keen et al. 2004). Typically, SOA consists of composable services, which are used to support the operations of a specific enterprise (Cherbakov et al. 2005).

Service-oriented computing is a broader term referred to as the computing paradigm that utilizes services as fundamental elements for developing applications (Papazoglou and Georgakopoulos 2003, Huhns and Singh 2005). Service-oriented computing is described by some researchers as the use of service-oriented architecture for implementing web services.

As indicated in the literature, service-oriented computing is process-driven. This is because business processes are a flexible way to organize and integrate enterprise applications at the business level (Zhao et al. 2005). At the computational level, service orchestration is the main function of the enterprise service bus, the coordinator of the functional modules under the service centric infrastructure.

2.2.3. Workflow Technology

Workflow technology has been widely adopted by organizations to automate business processes, reduce cycle time, and support e-business. Workflow Management Systems (WFMS) remove the process dependency by separating the process logic from the application logic, which makes information systems more flexible and adaptive to process changes (Stohr and Zhao 2001). Many approaches have been proposed to make WFMS even more flexible. Casati et al. (2000) proposed a rule-based workflow modeling method to allow a high degree of flexibility during workflow design and great adaptability in terms of process exception handling. Document-driven workflow is another such effort, which is different from production workflow systems (Wang and Kumar 2005). The latter models the workflow as an execution sequence of tasks while the former designs a process by specifying the data dependences among the documents. As a result, document-driven workflow is very suitable for less-structured, cooperative, and ad hoc workflows.

Today, many companies are investing a great amount to expose their business operations as web services in order to offer easy integration with a wide range of third-party applications. Web services choreography and orchestration technologies have been developed to connect individual web services to form higher value-added processes

(Peltz 2003). Workflow technologies based on the workflow standards such as Business Process Execution Language (BEPL) have helped organizations achieve better process flexibility (Weerawarana 2005). As we will illustrate later, the real world community source project - Kuali leverages workflow technology to enhance the flexibility and extensibility of the system.

2.3. Literature of the Economic Perspective

2.3.1. Decision Analysis

Decision analysis approaches have been used in IS field in the last decade (Kimmle 1972, King et al. 1975), and much progress has been made in recent research. Robert (2003) applied cooperative game theoretic models for decision-making in contexts of library cooperation. Ekstrom et al. (2005) construct a real options model to measure the value of flexibility in IT investment and use a decision analysis model to evaluate the value of the pilot project. Thomaidis et al. (2006) presents a fuzzy multi - criteria decision-making approach to the evaluation of IT projects.

In the study under the economic perspective, we apply a cooperative decision analysis framework for the community partnership decision. This type of analysis is different from a full-fledged strategic decision analysis based on game-theoretic frameworks, where decisions from involved parties are inherently intertwined and can only be made jointly through various equilibrium concepts. Although in future research, we plan to perform such game-theoretic analysis to analyze community partnership decisions, this study focuses on a simpler class of models in which for one participating organization, the behavior/decisions of other parties are known in a probabilistic sense. In this sense, our work can be classified as a game-against-nature formulation (Varian 1992). We note that performing cooperative decision analysis is nontrivial and can deliver important insights as to why certain organizations behave as observed.

2.3.2. Option Pricing Model

Investing in community source can be viewed as a real option for potential future benefit. In recent years, the use of Real Options Analysis (ROA) to support IT investment decisions in various application settings has been investigated in MIS literature. There are different option pricing models such as Margrabe's exchange option model, the Binomial model, the Black-Scholes (BS) model, and their extensions. One of the basic models for pricing financial options is the Black-Scholes model (Black et al. 1973, Hull 1993). It is widely used because of its simplicity. Benaroch and Kauffman (1999) examined the theoretical basis for applying option pricing model to IT evaluation. Moreover, they illustrated the feasibility of using the BS model to analyze a real deferral option. They argued that IT infrastructure projects may involve a "wait-and-see" component that directly follows the logic of ROA.

2.4. Literature of the Management Perspective

2.4.1. The Technology-Organization-Environment Framework

The Technology-Organization-Environment (TOE) framework was originally developed by DePietro, Wiarda and Fleischer, based on meta-analyses of prior organizational studies, to address the context in which technological innovation decision is made by an organization (DePietro et al., 1990). Technological context, organizational context, and environmental context influences the adoption and implementation of a firm's technological innovation.

The TOE framework has been examined by a number of studies on various IS domains. For instance, the adoption of electronic data interchange (EDI) has been extensively studied in the last decade (Mukhopadhyay et al. 1995). An EDI adoption model including three aspects (technological factors, organizational factors and environmental factors) was developed by Iacovou et al. (1995) based on seven case studies. Further, Kuan and Chau (2001) developed a perception-based TOE framework with six factors as EDI adoption predictors and confirmed that this framework was useful for understanding the adoption of technological innovation. This TOE framework has been extended to the e-business domain and open source domain. For example, Chau et al. (1997) developed an adoption model for open systems, which tied seven factors together representing three major contexts of open systems adoption: (1) external environment, (2) characteristics of the open systems technology innovation, and (3) organizational technology. In addition, Zhu et al. (2002) built a conceptual model for electronic business adoption incorporating six adoption facilitators and inhibitors.

In summary, the TOE framework has been well studied and shown to be a promising modeling technique. Thus, we adopt this theoretical framework and extend it to the community source domain, which has not been done in the literature.

2.4.2. Resource Dependency Theory

Organizational success in resource dependency theory (RDT) is defined as organizations maximizing their power (Pfeffer 1981). Research about power within organizations began by Weber (1947) and then generalized by Selznick (1949). RDT characterizes the links among organizations as a set of power relations. RDT has been applied to IT research in recent years. DeSanctis et al. (1994) pointed out that organizational information systems can have significant implications for organizations. Information technologies change control and coordination activities, shape the form of organizational coupling, and influence strategic coordination. Fulk et al. (1990) introduced new work practices, new responsibilities, and shifts to the organization's structure as new IT systems implemented. Tillquist et al. (2002) presented a new representation methodology for analyzing information technology and organizational dependency. In the study under the management perspective, we apply RDT to the understanding of issues in successful management of community source.

2.4.3. IS Success

Most literature on IS success focused on IS deployment success. The Delone and McLean (D&M) IS Success Model received the wide popularity in IS success literature, which was proposed by DeLone et al. (1992) as a framework for conceptualizing and operationalizing IS success. DeLone et al. (1992) introduced a comprehensive taxonomy to present a integrated view of the concept of IS success. DeLone et al. (2002) updated their original success model based on changes in the role and management of information systems and included arrows in the updated D&M IS Success Model to demonstrate associations among success dimensions in a process sense. However, little literature studies IS development project success. In community source, system development is more unique than system deployment since it is a multiple-institution work in a virtual environment. The success of system deployment in community source relies heavily on the success of system development.

2.4.4. Outsourcing Software Development

Outsourcing may be viewed as a natural step in the evolution of a business (Davis et al. 2006). An organization can outsource a wide selection of business activities, particularly those IS/IT related such as software development. The typical process of software outsourcing is that the software is developed by external vendors, purchased by clients, and then customized and integrated with other internal systems (Davis et al. 2006). Significant portions of a firm's software development activities have been routinely outsourced for years (McFarlan et al. 1995). There are no signs that this trend of IS outsourcing will cease (Beverakis et al. 2009). In fact, programming and software development continues to move out of internal IT department into outsourced companies (King 2008). According to a recent report, programming, software developing and testing, and software maintenance is among the most outsourced works of firms (Aspray et al., 2006).

The sourcing decision of software development is basically a "make or buy" issue. Prior to the commencement of a customized development project, the organization must decide whether to develop the software internally or to outsource it to external developers (Wang et al. 1997). Aside from the obvious benefit of cost savings, previous studies have identified two other primary motivations for outsourcing software development, to leverage the IS expertise of external vendors and to focus more attention on the organization's core competencies (Beverakis et al. 2009). Subsequently, the expected strategic benefits from outsourcing are: 1) improved IS development and 2) increased operational efficiency and effectiveness and thus enhanced overall performance (Beverakis et al. 2009; DiRomualdo et al. 1998). In the study under the management perspective, we examine the motivations for outsourcing the software development in community source.

CHAPTER 3. THE TECHNOLOGY PERSPECTIVE

3.1. Introduction

Organizations are facing an environment of ever increasing turbulence and change on a global scale. To survive under global competition, they need to meet more and more challenging requirements, such as shorter product cycle times and continuous costs reduction. One way to meet the new challenges is to improve the flexibility of enterprise systems, thus making the organization more adaptive and agile to unexpected changes. In particular, increased flexibility can give an organization more competitive advantage through faster response to varying customer needs and environmental conditions. As companies automate their business processes through workflow technology and electronic business, the flexibility of business process implementation can also greatly influence the organization's capacity for change. As has been long recognized by researchers (Nelson et al. 1997), companies have been using emerging technologies to enable flexibility in the enterprise computing environment. Most recently, a new set of information technologies collectively referred to as service-oriented computing have been adopted to support more enterprise agility (Zhao, Tanniru, and Zhang, 2007).

In this study, we examine the importance of service-centricity in open source development, in the context of Kuali. Kuali is an ideal case for studying service-centric open source development. It started with a focus on developing a university financial information system and has now extended the system scope by including research administration and student systems. The goal of Kuali is to combine resources from multiple universities to minimize development risk and financial burdens. Kuali is developed as an alternative to expensive and rigid systems such as Financial ERP modules offered by commercial software vendors. As a community source project, Kuali 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. 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. As we illustrate in this study, Kuali takes advantage remarkably of these emerging technologies to enable the strong system flexibility. In fact, we believe that Kuali is a good representative of a recent trend in combining service centricity with open source development to build scalable and flexible information systems.

All in all, service centricity and open source involve innovative concepts and practices that are very new to many. As such, there are a lot of confusion among both researchers and practitioners. In this study, we investigate the concepts and issues surrounding these two important topics based on the recent literature. Further, we examine the case of Kuali and its technological components, including service-oriented architecture, document workflow, and customization techniques.

3.2. Service-centric Community Source

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. In order to ensure flexibility and extensibility, Kuali uses the most up-to-date approaches and technologies such as open source, workflow, and service-oriented architecture. Figure 2 illustrates the Kuali system architecture that shows how the enterprise service bus is used to support service-oriented architecture. Currently, there are three main application modules, financial services, research administration, and student system.

At the center of the Kuali system is the Kuali enterprise service bus (KSB), the core concept of service-oriented architecture that coordinates the interactions between an application module and the various lower-level system services such as workgroup service, user service, and legacy wrapper service. KSB includes a service registry, remote access protocols, and service invocation procedures. Each application module carries its own documents, business objects, business rules, data dictionaries, and other local services. This service-centric architecture makes it relatively easier to enhance the application modules and add new modules. As shown in Figure 3.1, Kuali also has a modularized middleware infrastructure called Rice, sitting between the server and applications. Rice consists of four client-side-components: Kuali Enterprise Service Bus, which includes a service registry, remote access protocols, and service invocation procedures; Kuali Nervous System, which provides reusable code and shared services to enforce system development standards; Kuali Enterprise Workflow, which facilitates the routing and approval of transactions throughout the university; and Kuali Enterprise Notification, which provides a single list for all university-related communications.

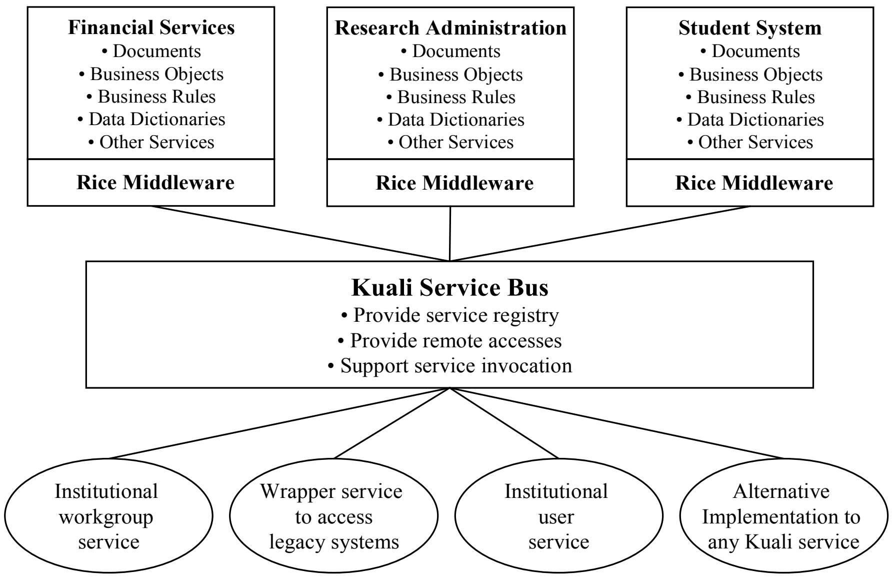

Figure 3 1 Service-Oriented Architecture of Kuali

3.3. Enabling Flexibility in Kuali

As a community source project, Kuali must balance among varying requirements from many universities. Thus, it is imperative to make the Kuali system flexible in terms of both customizability and extensibility. In this section, we take an in-depth look at how Kuali achieves flexibility in this unique environment.

3.3.1. Label Customization

An important aspect of flexibility is to enable the customization of the look and feel of its user interface. Specifically, Kuali allows its adopters to rename any label on all forms (Figure 3.2) without the need to write code.

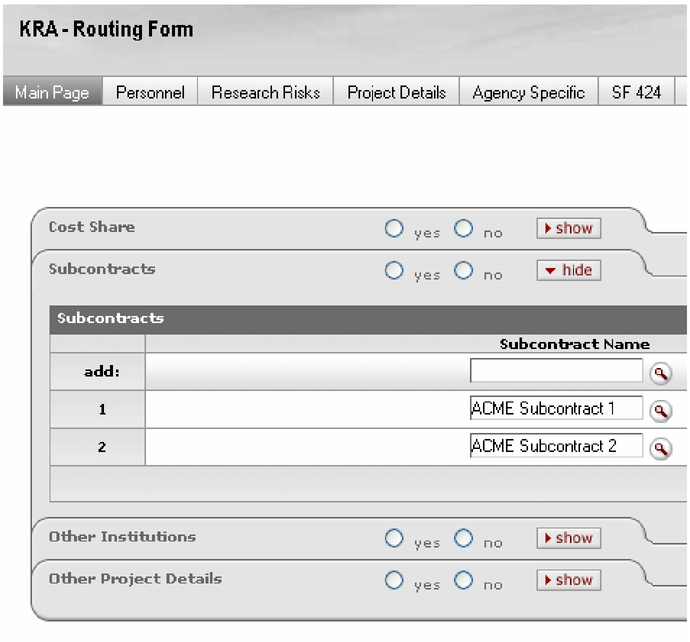

Figure 3 2 Kuali Forms and Labels

Kuali supports label customization by means of its Data Dictionary, which can be specified using XML files. For example, account.xml (Figure 3.3) holds various attributes about a Business Object “Account Number”. This customizability allows an institution to change the user interface to fit its unique lingo without changing the lower-level programs.

```
<attribute name="accountNumber" forceUppercase="true">
<label>Account Number</label>    <shortLabel>Account</shortLabel> <maxLength>7</maxLength>
<validationPattern>   <alphaNumeric   exactLength="7"   allowWhitespace="false"   allowUnderscore="false"   />
</validationPattern>
```

```
<required>true</required> <control>    <text size="10" /> </control>
<summary>Account Number</summary> <description>Identifier for a pool of funds.</description>
</attribute>
```

Figure 3 3 Data Dictionary Attribute Definition

3.3.2. Modification and Addition of Document Types

Kuali organizes its financial documents into a hierarchy of document types; as of today, it has 139 types on its website. Figure 3.4 illustrates a number of document types in the financial transactions category. Document types are a critical concept for Kuali since its business process evolves around the routing of documents.

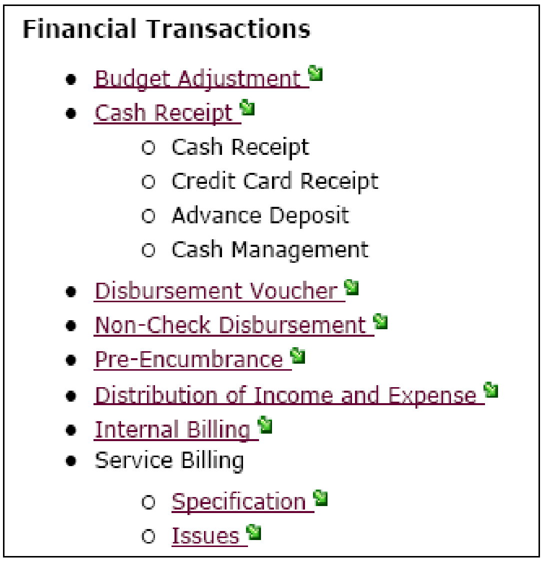

Figure 3 4 Kuali Document Type Examples

Kuali allows its adopters to modify and add new document type via XML files. As shown in Figure 3.5, the Account Maintenance Document type is defined via a XML (only partially show for simplicity). The components of a document type include business rules, authorization, locking keys, and so on.

```
<dictionaryEntry>
    <maintenanceDocument>
     ...
     <businessRulesClass>org.kuali.module.chart.rules.AccountRule</businessRulesClass>
     <documentAuthorizerClass>org.kuali.core.document.MaintenanceDocumentAuthorizerBase</document
AuthorizerClass>
     <authorizations>
     <authorization action="initiate">
     <workgroups> <workgroup>kualiUniversalGroup</workgroup> </workgroups>
     </authorization>
     </authorizations>
     <documentTypeName>KualiAccountMaintenanceDocument</documentTypeName>
     <documentTypeCode>ACCT</documentTypeCode>
        <label>Account Maintenance Document</label>
        <shortLabel>AcctMaintDoc</shortLabel>
        <summary>Account maintenance document</summary>
        <description>Document used to create or update Account objects</description>
        <lockingKeys>        <field        attributeName="chartOfAccountsCode"/>        <field
        attributeName="accountNumber" /> </lockingKeys>
        …
        </maintenanceDocument>
</dictionaryEntry>
```

Figure 3 5 AccountMaintenanceDocument.xml

3.3.3. Workflow Customization

Workflows in Kuali can be customized to suit the needs of different universities. Essentially, the Kuali workflow is document-based and allows the modification of document routing without having to write any code. As given in Figure 3.6, the Kuali workflow engine specifies the generic actions on financial documents including initial actions, approval actions, workflow processor, and post processor. Specific actions on each financial document are specified within the document type itself such as document initiator, organizational unit, and document approver. The Kuali workflow uses XML-driven routing capabilities, and routing rules can be defined or modified with respect to any document type without requiring a modification to the Kuali source code.

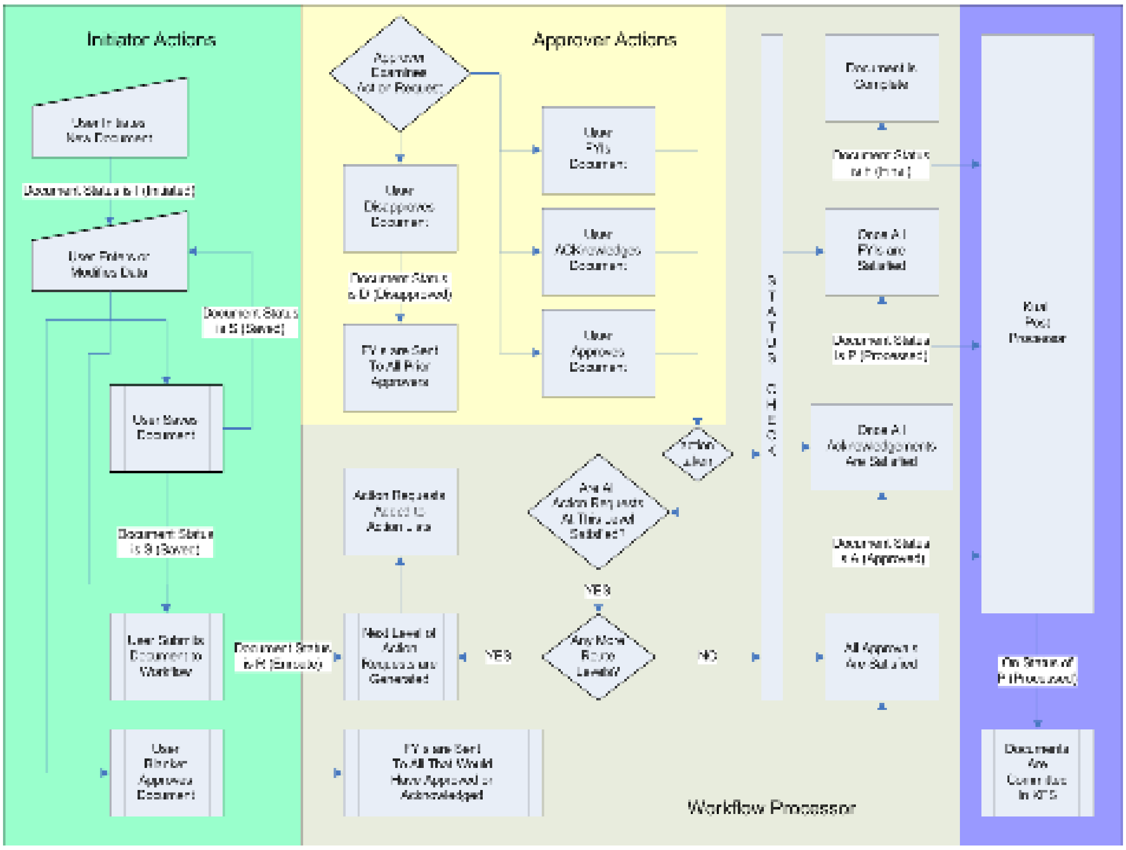

Figure 3 6 Kuali Workflow

3.3.4. Code Modification

This level of extensibility is inherent to the open source paradigm of Kuali. As an open source system, Kuali “allows its users to: 1) freely access, install, and run the software for any purpose; 2) modify the original software; 3) redistribute copies of the original or modified programs; and 4) share modifications with the community.” Further, as a community source system, Kuali “is based on many of the principles of open source development efforts, but community source efforts rely more explicitly on defined roles, responsibilities, and funded commitments by community members than some open source development models."

Kuali is distributed according to the Educational Community License version 1.0, which grants the permission to use, copy, modify, merge, publish, distribute, and sublicense this original work and its documentation, with or without modification, for any purpose, and without fee or royalty to the copyright holder(s). The adopters are required to notify any changes or modifications to the original work.

3.3.5. Addition of New Modules

As discussed in an earlier section, Kuali started as a financial system and has evolved to an umbrella suite of administrative higher education systems. It does so by providing a common architecture that can be extended to incorporate new applications. The Kuali framework improves productivity and consistency across the Kuali applications and creates a reusable development environment.

The service centric approach of Kuali makes it possible to "develop once and use anywhere and on any platform". This reduces the dependency of various development efforts and therefore simplifies the system architecture. Furthermore, workflow customization via business rules discussed previously makes it easier to create new application modules in a document-based routing. This further highlights the business value of service-oriented architecture, including the Kuali Rice Middleware approach.

3.4. Propositions

From the Kuali case, we can generalize several observations into a number of propositions, which will need to be validated in future research. Next, we state the propositions and explain them briefly without proofs.

Proposition 1: Service–oriented architectures enhance system extensibility.

As said previously, the Kuali's system architecture is service-oriented. Its enterprise service bus provides the necessary facilities to enable various application modules to access enterprise services without being hardwired to these services. This helps the Kuali adopters extend the system by adding new modules while reusing much of these enterprise services such as user services such as user directories and institutional services such as data dictionary.

Proposition 2: Workflow automation enables model-based application development and system maintenance without having to revise lower-level code.

Model-based applications can be modified by revising the process models without having to change any program code. The Kuali case indicates that workflow can help model the routing and authorization tasks and make it easier for an organization to streamline its processes and tasks. This feature is more pronounced in a community base project like Kuali in which the system is designed with built-in flexibility.

Proposition 3: Community source requires more flexible software architectures in order to meet the needs of the community.

The Kuali case demonstrates the need for close and intensive interactions between the functional aspects and the technical aspects on the regular basis because varying needs of multiple institutions make it very difficult to get the specifications right. This is not as essential in other types of open source projects when the functional requirements are relatively stable.

To validate these propositions, we need to develop additional and more specific hypotheses that are based on measurable parameters. This will be done as part of our future research.

3.5. Conclusions

3.5.1. Summary

In this study, we examined the need for technology flexibility and how to achieve it in the context of the Kuali project, a multi-university and multi-year project by bringing an existing system to the web. We demonstrated how Kuali utilizes a service-oriented architecture and the community source approach to minimize project risk and maximize cost sharing in enterprise application development. At the core of the flexibility endeavor is the five types of customization in Kuali. It is clear that service centricity is the key ingredient in the Kuali recipe for achieving flexibility. In addition, rule-based workflow management, XML-based document specification, and the Rice middleware approach are necessary ingredients as well. Based on the Kuali case, we summarized our observations in the form of research propositions. The important contribution of this study is the study of community source in the context of a large-scale project.

We will monitor the development and deployment of the Kuali system and continue with the research ideas we have presented in this study. Validation of the propositions we outlined in this study is the future research we plan to pursue.

3.5.2. Future Research Directions

Next, we outline several research directions related to the Kuali case, including community source, workflow modeling paradigms, and software adoption methodology. We discuss unique research opportunities along with those research directions.

- **Workflow Modeling Paradigms**

Currently, the Kuali 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 other Kuali modules. For instance, the Research Administration Module, currently under development, contains very complex workflows. The COEUS workflow (http://www.princeton.edu/~orpa1/coeus/) (Figure 3.7) consists of over twenty processes, each of which could include many steps as illustrated in Figure 8. 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

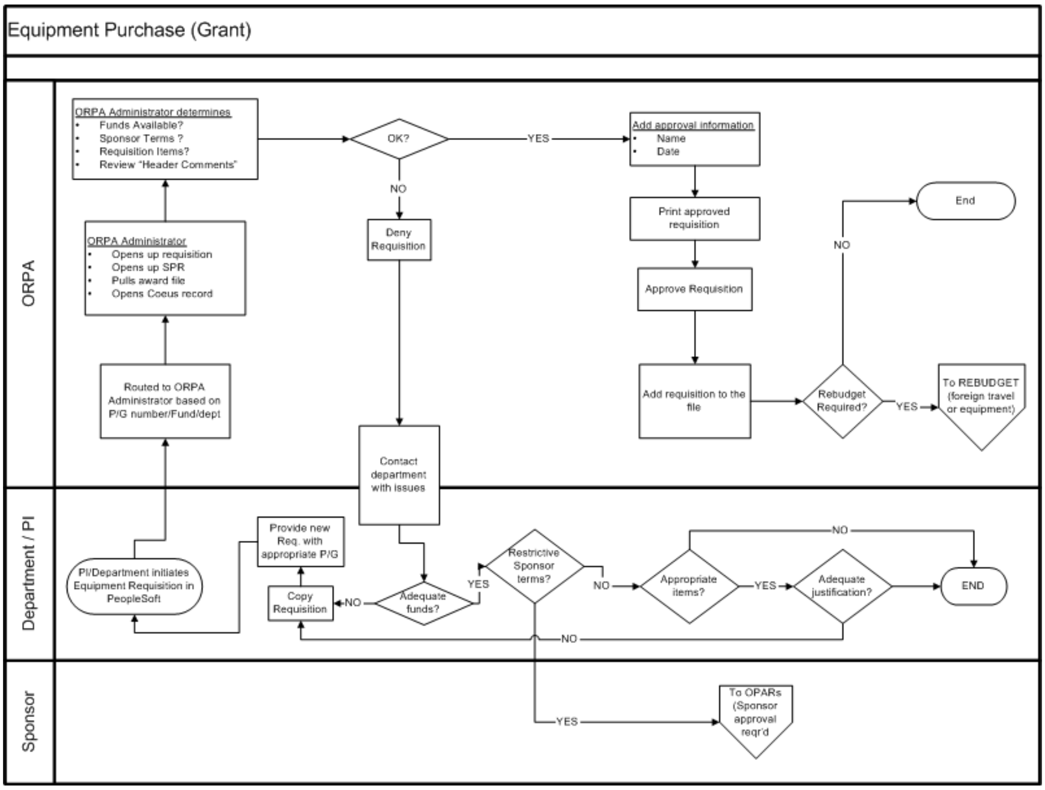

Figure 3 7 Equipment purchase process in research administration

underlying process logic. This raises a research direction on when a particular workflow modeling paradigm should be used. Specific research issues include the development of criteria on how to choose between control-flow-driven workflow modeling and rule-based and document-driven workflow modeling (Wang and Kumar 2005), mechanisms and procedures to extract implicit control flow from workflow rules, and rule languages that are capable of modeling more complex workflows (van der Aalst et al. 2003).

- **Adoption Methodology for SOA-based Software**

As demonstrated with the Kuali case, enterprises need to make software adoption decisions amidst recent technology advances such as service-oriented architectures and workflow automation. For example, how the universities other than Kuali development partners know Kuali is the right system or not for them is a non-trivial problem. As such, it is interesting to study how to describe the fitness of the system capabilities and enterprises needs. In order to study this issue, we need to understand how to formally specify the capabilities of the system, how to formally specify the needs of enterprises, and how to measure the fitness. Currently there is no formal methodology that allows enterprises to specify their system needs and compare with software capabilities in a manner similar to conceptual database design via ER diagram. In other words, there should be a software adoption methodology to describe the fitness of the system capabilities and enterprises needs.

A related concept is business modeling (Cherbakov et al. 2005). Business modeling is an emerging area of research. Researchers have studied the difference between a business model and a business process model (Gordijn et al. 2000), how related project activities can be modeled consistently to support the reengineering effort of the same business entity (Gulla et al. 2000), and how to develop business models to support the development of complex systems (Petrovic et al. 2001). Our focus in this direction is to develop a methodology for adopting software systems for given business requirements.

CHAPTER 4. THE ECONOMIC PERSPECTIVE

4.1. Introduction

The community source approach provides a viable alternative to vender-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.” As in general, the resulting software package of a community source project is available as an open source free or charge, it seems counter-intuitive for an organization to be willing to make major investments. Yes, 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.

The main research question to be examined in this study is “what motives partnering institutions to invest in community source development, considering the community output will become open source in the end.” We start our investigation by analyzing Kuali. Several derivative research questions to be examined include what are the costs and benefits of becoming a partner of the community source project and under what conditions an institution might be more willing to become a development partner.

The modeling approach adopted in our study is cooperative analysis framework, with an emphasis on interacting behavior among institutions. Decision-making in community source is challenging since the success of community source projects is affected by many factors. For example, successful collaboration among multiple partners plays an important role in the level of accomplishment of community source projects. The Black-Scholes model is used in the payoffs functions for each institution. The Black-Scholes model was developed to capture the randomness in the stock market and the valuation of real options; as such, it is suitable for analyzing investment decisions in community source. To conduct analytical analysis, the values we use for the parameters in the Black-Scholes model is based on observing the Kuali case.

This study makes three important contributions in this context: (1) it formulates the community source partnership decision under a cooperative analysis framework. (2) It presents the first application of the Black-Scholes model that uses a real world community source project as its test bed. (3) It gains insights about various interesting investment behavior observed in emerging community source practice, in turn help the institutions make related decisions.

4.2. Decision Analysis for Investments in Community Source

In this section, we will develop a formal model to analyze the community partnership decision. Since one institution's decision will be affected by other institutions' decisions, we apply a cooperative decision framework to investigate the interactions among institutions.

4.2.1. Model Setup

To understand the rationale for an institution to join in a community source project as a paying developer partner, we first formalize various decisions available to this institute and their impact. For simplicity and ease of exposition, in our first modeling attempt, we assume that there are two players – Institution A and Institution B; both of them are engaged in the community source investment decision process. Each institution needs to decide if it is going to join as a development partner. Each can choose to play one of two strategies: join or not-join.

To develop an analytical model that captures the main elements of the real-world practice, we emphasize the external funding and open source in the community source model. In the traditional system collaboration model, institutions joining together to develop the system will not receive external funding and the resulting software will not open source. Under the community source environment, the situation is quite different. Institutions who join as development partners in community source can receive external funding and the system needs to be open source eventually (Table 4.1). We can see that in a community source project, once the institution joins as a development partner, the project could receive external funding from third-party sources such as a private foundation. Two institutions joining together to develop the project can help reduce the overall project risk and share the development cost. On the other hand, since the project will become open source eventually, the institution which decides not to join as a development partner can wait to adopt the software later on for free. In other words, the open source end result creates disincentive to join. Also the project risk will also make the institution to wait.

We make the following assumptions in our model to reflect these modeling considerations:

1. There are only two institutions, Institution A and Institution B, in the community. The community can be formed with one institution or two institutions.
2. The development partner needs to make an initial investment to the community. We assume a flat fee; the amount of the initial investments is fixed for all development partners.
3. There is an external funding source, a foundation, which will be contributing to the project. Under this assumption, the external funding is a benefit for the institution to join the community.
4. The system developed by the community will be open source eventually. Under this assumption, the institution who decides not to join as a development partner can wait to adopt the software when it becomes open source.
5. Success is very likely in the community source project. We ignore the possibility of failure in the community source development process to simplify the model. Models considering project failures are planned for future research.
6. Institution A knows a prior the possibility for Institution B to join the community, ρ_B. Institution B knows the possibility for Institution A to join the community, ρ_A. Information about ρ_A and ρ_B is common knowledge to both parties. This is a key assumption to simplify the model to make it tractable under the game-against-nature framework.

7. If Institution A (B) joins the community to develop the software and Institution B (A) does not join, Institution B (A) will wait until the software becomes open source. In reality, there might be other alternatives for Institution B (A). For instance, it might purchase a third-party software package or develops it in-house. Our model ignores such alternatives.
8. If both institutions choose not to join, they will develop the system individually.
9. Each of institution is trying to maximize the payoff when they make the decision for joining to be partner or not.
10. If both of the institutions choose to develop, Institution A gets payoff PA_{11}, Institution B gets payoff PB_{11}.
11. If both of the institutions choose to not to develop, there will be no software developed in this community. Institution A and Institution B will develop the system individually. They get payoff PA_{22} and PB_{22} respectively.
12. If Institution A chooses to join and Institution B chooses not to join, Institution A who join to develop gets payoff PA_{12}. Although Institution B does not join to develop at this stage, it can still adopt the software when the software becomes open source. So Institution B payoff is PB_{12}.
13. If Institution B chooses to develop and Institution A chooses not to develop, the payoff for Institution B is PB_{21} and the payoff for Institution A is PA_{21}.

Institution A \ Institution B	Partner	Not partner
Partner	Institution A , Institution B • External Funding • Open source (Both of institutions join the CS) PA_{11}, PB_{11}	Institution A (joining the CS) • External Funding • Open source Institution B (waiting for open source) • No external funding • Open source PA_{12}, PB_{12}
Not partner	Institution A (waiting for open source) • No external funding • Open source Institution B (joining the CS) • External Funding • Open source PA_{21}, PB_{21}	Institution A , Institution B • No External Funding • No Open source (Institution A and B develop the system individually) PA_{22}, PB_{22}

Table 4 1 A Cooperative Analysis Framework.

4.2.2. Payoff Analysis

As we can see above, there are four possible scenarios: Institution A joining and Institution B joining (AJ-BJ), Institution A joining and Institution B not joining (AJ-BNJ), Institution A not joining and Institution B joining (ANJ-BJ), Institution A not joining and Institution B not joining (ANJ-BNJ).

In this study, we apply an option pricing model to decision making in community source investment. Past IS research has recognized that IT investments can embed various types of real options such as defer, stage, explore, alter operating scale, abandon, lease, outsource, and growth (Benaroch 2002). In a community source approach such as Kuali, there are different types of options a participating institution can choose, such as “become a development partner,” “wait and deploy Phase I,” “wait and deploy Phase II,” “expand”, “monitor,” and “abandon.”

Although the traditional NPV model has been extensively used in the past for investment analysis, it fails to take into account of various important real options relevant to IS development (Taudes et al. 2000). As such, option pricing models such as Black-Scholes and binomial have been adopted for IS investment analysis in various contexts (Benaroch et al. 1999).

In our study, the Black-Scholes model is chosen to calculate the institutions' payoffs under different scenarios. This model was chosen largely because of its simplicity and wide application. The Black-Scholes model is shown below:

$$BS = V\mathrm{N}(d_1) - Xe^{-rT}\mathrm{N}(d_2) \qquad (1)$$

where $d_1 = \frac{\ln(V/X) + (r + \sigma^2/2)T}{\sigma\sqrt{T}}$ and $d_2 = d_1 - \sigma\sqrt{T}$

N(.) is the cumulative standard normal distribution function, and d_1 and d_2 are used to determine the cumulative value of the distribution, which in turn affect the value of the asset.

The variables used in the cooperative analysis framework are listed in Table 4.2. The default values are also given in the table based on our observation of the Kuali case. We note that in our subsequent analysis, sensitivity analyses have been conducted to make sure that our main findings are not tied to specific values of these parameters.

Symbol	Notation	Default value
V_0	Base value for the system	\$2m
μ	Savings from the system per student	\$50
S_A S_B	The number of students in Institution A and Institution B	10K, 50K
I_0 I_0^*	Base investment of development in community, flexibility related investment	\$6 m, \$3 m
H_0 H_0^*	Base investment of individual development, flexibility related investment	\$4 m
D_0 D_0^*	Base deployment cost for joining the community or developing individually, flexibility related deployment cost	\$0.1 m, \$0.05 m
D_1 D_1^*	Base deployment cost for taking open source result, flexibility related deployment cost	\$0.3 m, \$0.15 m
E_0 E_0^*	Base maintenance cost for joining the community or developing individually, flexibility related maintenance cost	\$0.1 m, \$0.05 m
E_1 E_1^*	Base maintenance cost for taking open source result, flexibility related maintenance cost	\$0.3 m, \$0.15 m
λ	The proportion of flexibility related investment to base investment	50%
η	The size factor related to maintenance cost	50
r σ f	Risk free rate, System variance , flexibility rate	10%, 50%, 50%
M	Total amount of external funding	\$0.5 m
ρ_A ρ_B	The probability of Institution A joining, the probability of Institution B joining	50%, 80%
T_1	Lifecycle of the resulting software if joining the community	5 yrs
T_2	Lifecycle of the resulting software if taking open source result	4 yrs
T_3	Lifecycle of the resulting software if developing individually	3 yrs

Table 4 2 Variables and Default Values

We make the following assumptions to regarding the parameters in the Black-Scholes model while calculating the payoffs in four possible scenarios.

1. $V = V_0 + S\mu$

   We define that system value (V) is consisting of two parts: base value (V_0) and size related value ($S\mu$). Parameter S means the number of students in the university and μ is the savings from the system per student.

2. $X = I + D + E$

   The cost (X) is consisting of three parts: investment (I), deployment cost (D) and maintenance cost (*E*). External funding (*M*) can be considered as deduction of *I*.

3. $I = I_0 + f\lambda\lambda_0$

We assume that the initial investment increases if the system demands more flexibility. Investment (*I*) is consisting of two parts: initial investment (I_0) and flexibility related investment ($f\ \lambda I_0$). Parameter λ is defined as the percentage of initial investment which is related to flexibility part.

4. $D = D_0 + \frac{\lambda D_0}{f}$

We assume that the deployment cost decreases when the system is more flexible. Deployment cost (*D*) is consisting of two parts: initial deployment cost (D_0) and flexibility-related deployment cost ($\frac{\lambda D_0}{f}$). Parameter λ is defined as the percentage of initial deployment cost which is related to flexibility part. We assume that the initial deployment cost of institutions who participate in developing the system (joining the community or developing the system individually) is less than that of institutions who simply take the open source result.

5. $E = E_0 + \frac{\lambda E_0}{f} + \eta S$

We assume that maintenance cost decreases when the system is more flexible. Maintenance cost (*E*) is consisting of three parts: initial maintenance cost (E_0), flexibility related maintenance cost ($\frac{\lambda E_0}{f}$) and size related maintenance cost (ηS).Parameter η is the size factor related to maintenance cost. The parameter λ is defined as the percentage of initial maintenance cost which is related to flexibility part.

We also assume that the initial maintenance cost of institutions who participate in developing the system (joining the community or developing the system individually) is less than that of institutions who simply take the open source result. *T* is defined as the system usage time in community source. *T* is affected by development time and deployment time. If the institution joins the community to develop the system, development time will be shortened with joint effort in development and the deployment time will be shortened either because of readily-available system customization and existing staff training. As such, the system application time under this situation is the longest. If the institution simply takes the resulting open source package without making any effort in development stage, the deployment time will be longer. If the institution develops the system individually, the development time is expected to be much longer than the community source approach.

6. We define ρ as the probability vector of Institute A and B joining in community source. We assume that ρ is common knowledge. Since the size of Institution A is smaller than that of Institution B, we believe that the probability of joining in community source for institution A is lower. We now present the payoff functions of Institute A and B in these four possible scenarios.

Scenario 1: both Institution A and Institution B join the community – (AJ, BJ)

Under this scenario, Institution A and Institution B receive external funding M, divided equally between them to support development, for developing the system jointly.

The net payoff function for institution A in (AJ, BJ) is as follows:

$$PA_{11} = BS_A = V_A \mathrm{N}(d_{1A}) - X_A e^{-rT_1} \mathrm{N}(d_{2A})$$

where $d_{1A} = \frac{\ln(V_A / X_A) + (r + \sigma^2 / 2)T_1}{\sigma \sqrt{T_1}}$ and $d_{2A} = d_{1A} - \sigma \sqrt{T_1}$

$V_A = V_0 + S_A\mu$, $X_A = \mathrm{I} + D_A + E_A - \mathrm{M}/2$

$\mathrm{I} = I_0 / 2 + f\lambda \mathrm{I}_0 / 2$, $D_A = D_0 + \frac{\lambda D_0}{f}$, $E_A = E_0 + \frac{\lambda E_0}{f} + \eta S_A$

The net payoff function for institution B in (AJ, BJ) is as follows:

$$PB_{11} = BS_B = V_B \mathrm{N}(d_{1B}) - X_B e^{-rT_1} \mathrm{N}(d_{2B})$$

where $d_{1B} = \frac{\ln(V_B / X_B) + (r + \sigma^2 / 2)T_1}{\sigma \sqrt{T_1}}$ and $d_{2B} = d_{1B} - \sigma \sqrt{T_1}$

$V_B = V_0 + S_B\mu$, $X_B = I + D_B + E_B - M/2$

$\mathrm{I} = I_0 / 2 + f\lambda \mathrm{I}_0 / 2$, $D_B = D_0 + \frac{\lambda D_0}{f}$, $E_B = E_0 + \frac{\lambda E_0}{f} + \eta S_B$

Scenario 2: Institution A joins the community but institution B does not – (AJ, BNJ)

Under this scenario, Institution A receives external funding M for developing the system. Institution B does not join and waits for the open source; the cost for institution B only includes the deployment cost and maintenance cost.

The net payoff function for institution A in (AJ, BNJ) is as follows:

$$PA_{12} = BS_A = V_A \mathrm{N}(d_{1A}) - X_A e^{-rT_1} \mathrm{N}(d_{2A})$$

where $d_{1A} = \frac{\ln(V_A / X_A) + (r + \sigma^2 / 2)T_1}{\sigma \sqrt{T_1}}$ and $d_{2A} = d_{1A} - \sigma \sqrt{T_1}$

$V_A = V_0 + S_A\mu$, $X_A = \mathrm{I} + D_A + E_A - M$

$\mathrm{I} = I_0 + f\lambda I_0$, $D_A = D_0 + \frac{\lambda D_0}{f}$, $E_A = E_0 + \frac{\lambda E_0}{f} + \eta S_A$

The net payoff function for institution B in (AJ, BNJ) is as follows:

$$\mathrm{PB}_{12} = BS_B = V_B \mathrm{N}(d_{1B}) - X_B e^{-rT_2} \mathrm{N}(d_{2B})$$

where $d_{1B} = \dfrac{\ln(V_B / X_B) + (r + \sigma^2 / 2) T_2}{\sigma \sqrt{T_2}}$ and $d_{2B} = d_{1B} - \sigma \sqrt{T_2}$

$V_B = V_0 + S_B \mu$, $X_B = D_B + E_B$

$D_B = D_1 + \frac{\lambda D_1}{f}$, $E_B = E_1 + \frac{\lambda E_1}{f} + \eta S_B$

Scenario 3: Institution A does not join the community but institution B joins, simply denoted as (ANJ, BJ):

This scenario is symmetrical to Scenario 2.

The net payoff function for Institution A in (ANJ, BJ) is as follows:

$$PA_{21} = BS_A = V_A \mathrm{N}(d_{1A}) - X_A e^{-rT_2} \mathrm{N}(d_{2A})$$

where $d_{1A} = \dfrac{\ln(V_A / X_A) + (r + \sigma^2 / 2) T_2}{\sigma \sqrt{T_2}}$ and $d_{2A} = d_{1A} - \sigma \sqrt{T_2}$

$V_A = V_0 + S_A \mu$, $X_A = D_A + E_A$

$D_A = D_1 + \frac{\lambda D_1}{f}$, $E_A = E_1 + \frac{\lambda E_1}{f} + \eta S_A$

The net payoff function for Institution B in (ANJ, BJ) is as follows:

$$PB_{21} = BS_B = V_B N(d_{1B}) - X_B e^{-rT_1} N(d_{2B})$$

where $d_{1B} = \dfrac{\ln(V_B / X_B) + (r + \sigma^2 / 2) T_1}{\sigma \sqrt{T_1}}$ and $d_{2B} = d_{1B} - \sigma \sqrt{T_1}$

$V_B = V_0 + S_B \mu$, $X_B = I + D_B + E_B - M$

$I = I_0 + f \lambda I_0$, $D_B = D_0 + \frac{\lambda D_0}{f}$, $E_B = E_0 + \frac{\lambda E_0}{f} + \eta S_B$

Scenario 4: Neither institution A nor institution B joins the community --- (ANJ, BNJ):

Under this scenario, both institutions choose to develop the system individually and the result is not open source.

The net payoff function for Institution A in (ANJ, BNJ) is as follows:

$$PA_{22} = BS_A = V_A \mathrm{N}(d_{1A}) - X_A e^{-rT_3} \mathrm{N}(d_{2A})$$

where $d_{1A} = \dfrac{\ln(V_A / X_A) + (r + \sigma^2/2)T_3}{\sigma \sqrt{T_3}}$ and $d_{2A} = d_{1A} - \sigma \sqrt{T_3}$

$V_A = V_0 + S_A \mu$, $X_A = H_0 + D_0 + E_0 + \eta S_A$

The net payoff function for Institution B in (ANJ, BNJ) is as follows:

$$PB_{22} = BS_B = V_B N(d_{1B}) - X_B e^{-rT_3} \mathrm{N}(d_{2B})$$

where $d_{1B} = \dfrac{\ln(V_B / X_B) + (r + \sigma^2/2)T_3}{\sigma \sqrt{T_3}}$ and $d_{2B} = d_{1B} - \sigma \sqrt{T_3}$

$V_B = V_0 + S_B \mu$, $X_B = H_0 + D_0 + E_0 + \eta S_B$

The expected payoff for Institution A joining is $\mathrm{EPAJ} = \rho_{\mathrm{B}} \mathrm{PA}_{11} + (1 - \rho_{\mathrm{B}}) \mathrm{PA}_{12}$

The expected payoff for Institution A not joining is $\mathrm{EPANJ} = \rho_{\mathrm{B}} \mathrm{PA}_{21} + (1 - \rho_{\mathrm{B}}) \mathrm{PA}_{22}$

The expected payoff for Institution B joining is $\mathrm{EPBJ} = \rho_{\mathrm{A}} \mathrm{PB}_{11} + (1 - \rho_{\mathrm{A}}) \mathrm{PB}_{21}$

The expected payoff for Institution B not joining is $\mathrm{EPBNJ} = \rho_{\mathrm{A}} \mathrm{PB}_{12} + (1 - \rho_{\mathrm{A}}) \mathrm{PB}_{22}$

To gain further understanding of the community source investment decisions and study the relationship among various contributing factors, we have conducted a series of computational analysis through varying the values of some key parameters and observing the expected payoffs as given above. Figures 4.1-4.6 summarize our findings.

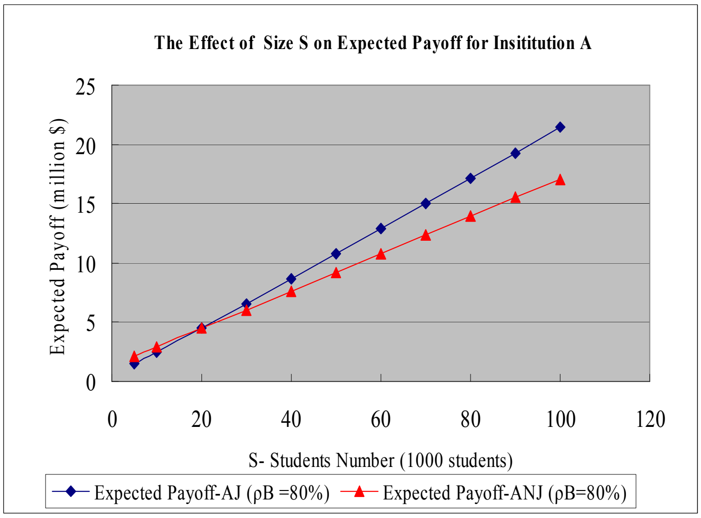

Figure 4 1 Effect of Size on Expected Payoff for Institution A

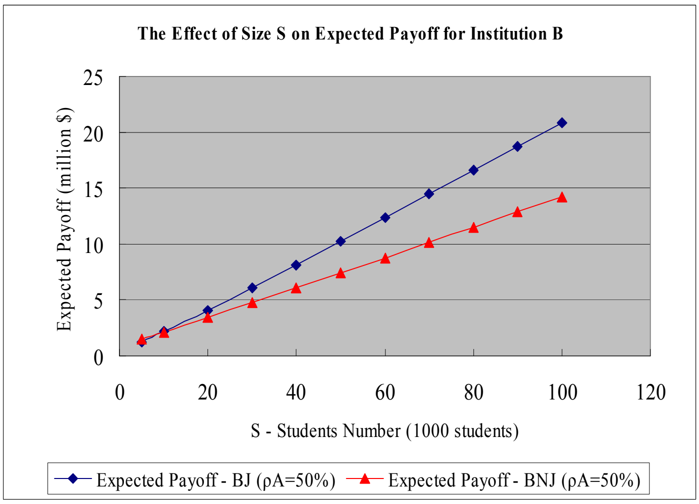

Figure 4 2 Effect of Size on Expected Payoff for Institution B

Figure 4.1 shows the effect of size on the expected payoff for institution A. As shown, there is a crossover point around 20,000 students. Before this point, the expected payoff for not joining is greater than that for joining. After this point, joining for institution A becomes a dominating strategy. More generally, we note that larger institutions are more likely to join the community source than their smaller counterparts. Figure 4.2 shows the effect of S on the expected payoff for Institution B. The trend in Figure 4.2 is similar to Figure 4.1. The difference is that the crossover point has moved up to 10,000 students.

Figure 4.3 and 4.4 show the effect of the deployment cost on the expected payoff for institution A and institution B, respectively. We can see that the expected payoff for not joining is greater than that for joining in Figure 4.3 while the expected payoff for joining is greater than that for not joining in Figure 4.4. This reinforces the observation that small-sized institutions are more likely not joining community source. We believe that an institution with a strong internal technology development team is more likely to have a reduced deployment cost. As this internal technology development capability decreases, the expected payoff for small institutions (Figure 4.3) decreases more rapidly than that for large institutions (Figure 4.4). Large institutions with strong internal capability are more likely to join the community source.

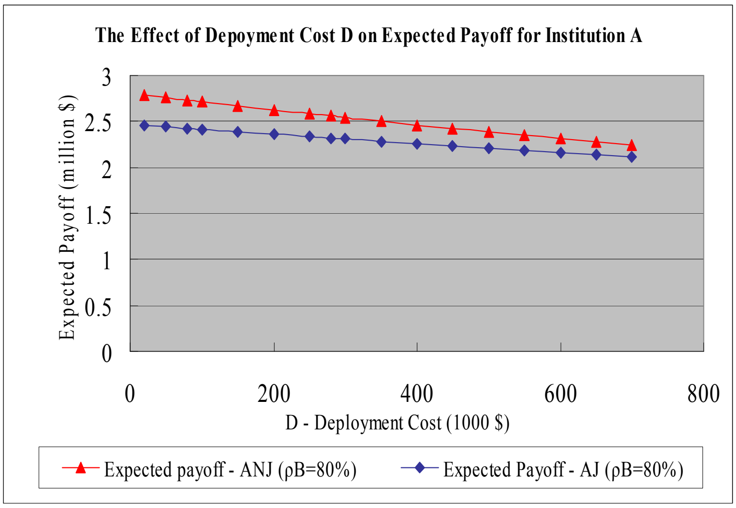

Figure 4 3 Effect of Deployment Cost on Expected Payoff for A

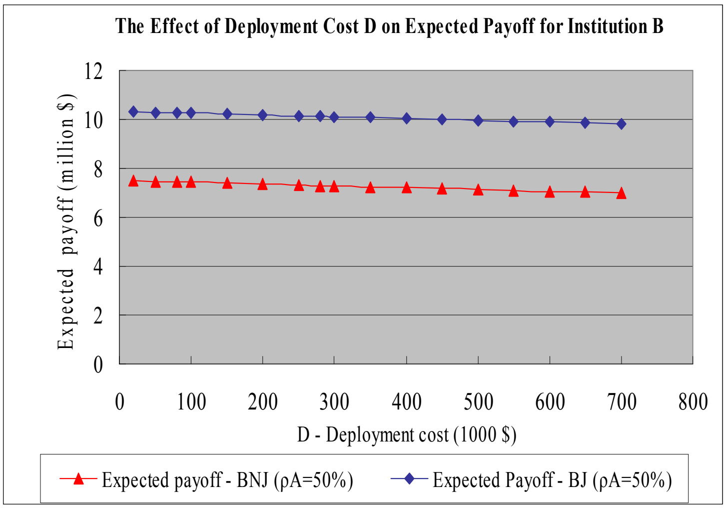

Figure 4 4 Effect of Deployment Cost on Expected Payoff for B

Figure 4.5 shows the effect of system flexibility on the expected payoff for Institution A. On the lower end of the flexibility rate, there is a crossover point. When the flexibility rate is lower than the crossover point, the expected payoff for joining is greater than that for not joining. When the flexibility rate is higher than the crossover point, the expected payoff for not joining becomes dominating. When small size institutions are making the decision to join the community, they are pursuing certain degree of system flexibility which can reduce their deployment cost and maintenance cost without increased investment. Since high flexibility system requires high initial investment, small size institutions may choose to wait for open source to avoid huge initial investment. Figure 4.6 shows the effect of system flexibility on expected payoff for institution B. We can see that expected payoff for joining is always dominant no matter the degree of flexibility rate. Large institutions are more likely to join no matter the degree of flexibility rate.

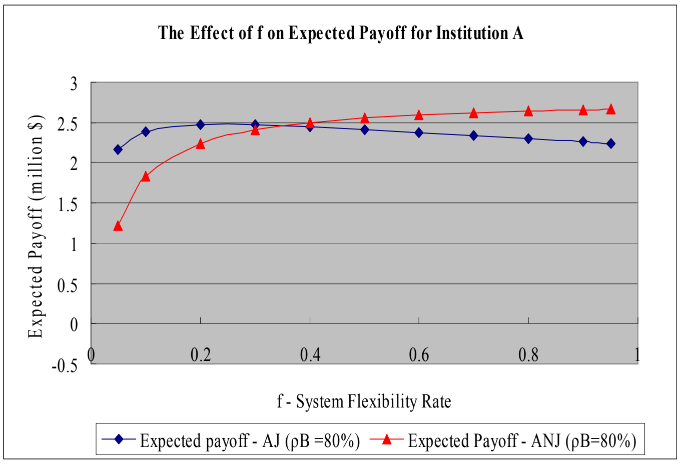

Figure 4 5 Effect of Flexibility Rate on Expected Payoff for A

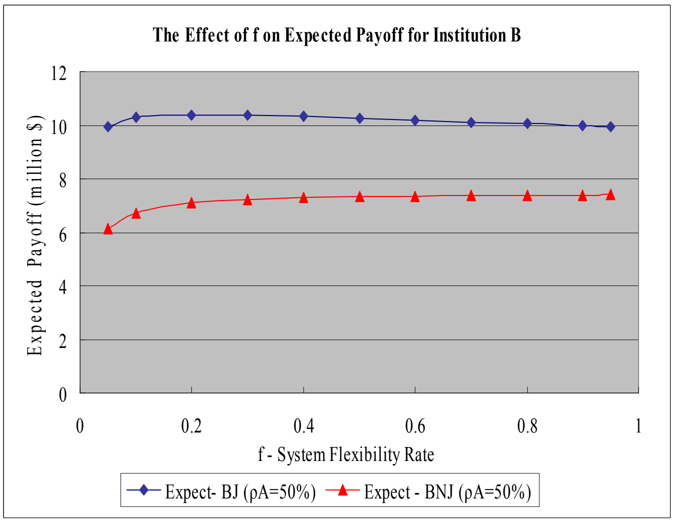

Figure 4 6 Effect of Flexibility Rate on Expected Payoff for B

4.3. Propositions

Next we derive several propositions based on observing cooperative decision analysis with parameters large set based on the Kuali case.

Proposition 1. Larger institutions are more likely to join community source.

Discussion. It is shown in Figure 4.2, Figure 4.4 and Figure 4.6 that the expected payoff for Institution B to join the development partnership is almost always dominant which strongly indicates that larger institutions are more likely to join community source. Since the number of students affects the system value, larger institutions can get more value of the system. The initial investment for a large institution is assumed to be the same as that for a small institution. Larger institutions usually have more financial resource, so the initial investment fee will be more affordable to larger institutions.

Proposition 2. Small institutions will be more likely to join if the system flexibility is low.

Discussion. There are two reasons for this behavior: First, when system requires more flexibility, initial investment will increase and deployment cost and maintenance cost will decrease. Lower system flexibility requires less initial investment, which can be afforded by small institutions. Second, lower system flexibility makes the option of waiting for the open source less attractive, which pushes small institutions to join the development partnership. Figure 4.5 clearly shows that the expected payoff for small institutions joined the development partnership is higher than that for small institutions not joined when the flexibility is at the lower end.

Proposition 3. The more internal development capability the institution has, the more likely it will become a development partner.

Discussion. We assume that institutions with more internal development capability can deploy the system with lower cost. As a result, expected payoff decreases when deployment cost increases (Figures 4.3 and 4.4). This is because higher deployment cost eats into the expected payoff. Note that although the decreasing trend of payoff is the similar, Figure 4.3 shows that smaller institutions turn not to join the development partnership and Figure 4.4 shows that larger institutions turn to join the development partnership.

4.4. Conclusions

4.4.1. Summary

In this study, we investigated the institutional investment decisions in community source. Using cooperative decision analysis framework, we analyze decisions on whether or not an institution should invest in a community source. We derived several propositions based on the Kuali case and the results of a computational study based on both our analytical model and parameter setting derived from the Kuali case.

As we have shown in this study, by becoming a development partner in the Kuali project, the institution can influence the application features, complete the deployment sooner, minimize the total cost of ownership, and reduce the variance of the system value. This incentive is the main driving force for institutions to make community source projects succeed.

4.4.2. Limitations

There are two main limitations in this study. First, the analytical model can be extended in several important directions. Current analysis involves only two institutions whereas often dozens of partners are present in community source projects. Relevant factors, such as project successful rate, are yet to be considered. Second, there are no empirical data so far to fully validate the propositions.

4.4.3. Future Research

In future research, we will first collect empirical data to validate the propositions. A survey as well as interviews has been planned among partner institutions in Kuali. Second, in the current study, we assess the impact of individual factors. We plan to investigate the joint impact of multiple factors on investment decisions. Lastly, more sophisticated decision models (including game-theoretic ones) will be developed and evaluated.

CHAPTER 5. THE MANAGEMENT PERSPECTIVE - PROJECT SUCCESS

5.1. Introduction

Community source has emerged as a new approach for developing enterprise applications. One of the interesting questions is why institutions are interested in joining community source? Resource dependency theory (RDT) can be applied here to explain this new phenomenon. Assuming the institutions have similar, but not identical, application requirements, there is clearly an opportunity of pooling the resources together towards the common objective – to develop sharable applications with a fraction of the costs by each individual institution. This is really the motivation of community source. Further, the basic requirements of partner institutions include similar application requirements and similar development philosophy in order to unify and share the tasks of system analysis, design, and development. RDT suggests that actors lacking in essential resources will seek to establish relationships with (i.e., be dependent upon) others in order to obtain needed resources (Pfeffer 1981). In community source, institutions participate and invest in community source to maintain needed external resources. Acquiring the external resources (community source) needed by an institution comes by decreasing the institution's dependency on commercial software, which modifies an institution's power with other institutions.

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. As we investigate in this study, 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.

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?

To better understand these issues, a research model is developed and based on the technology – organization – environment (TOE) framework in technology innovation and adoption literature. To make a community source project successful, it requires efforts not only from the community source organization (usually in a format of foundation), but also from the participating partners. Therefore, we examine the factors from the perspectives of a community source foundation and its institutional partners. In this study, we apply the TOE framework to the project success in community source. Further, we use the resource dependency theory (RDT) to explain the research model.

The main contribution of this study is twofold. First, we extend previous studies on the TOE framework and apply this framework to a new system development approach – community source. Second, we make use of a real word community source case to rethink the resource dependency theory.

5.2. The Research Model and Hypotheses

The ubiquitous nature of the "collaborative open source" in community source suggests that more attention should be devoted to understanding resource dependency in constructing a theoretical model of community source development. However, the lack of research on community source development precludes us from building a research model on existing theories directly. We believe the first step to study systematically the innovative system development model of community source is identifying its successful factors. To achieve this research objective, we conducted exploratory research and collected empirical data from the Kuali case, based on the Technology-Organization-Environment (TOE) framework. The TOE framework guided us to explore which technological, organizational or environmental factors are salient for the success of the KFS project.

Although the TOE framework was developed to understand the adoption of general technological innovation, the contexts in this framework can be used to explain the success of community source projects. The history of inter-institutional cooperation to jointly produce software systems has been littered with failures, so starting out to work together is no guarantee of success. It is not easy to successfully develop large, complex financial systems when all the developers and users are in the same organization, and the difficulties are multiplied when both the users and developers come from different institutions with different cultures. Kuali is overcoming these difficulties with innovative organizational and management strategies.

We treated a community source project as an innovation of developing enterprise application systems. The success of a community source project mainly relies on the success of the development and deployment of it final system. In commercial software and traditional open source projects, the developers and the users are different groups. However, in community source projects, the institutions that participated in system development are very likely to adopt the software themselves. Therefore, the development and the deployment are closely related in community source. In community source, some factors affecting the success in community source deployment might also affect the success in community source development. Although we focus on the successful factors in community source development in this study, it is hard to totally ignore the influence from deployment in community source when we study project success.

In the TOE framework, there are three elements that influence the process by which innovations are adopted. They are (1) the external environmental context, (2) the technological context, and (3) the organizational context. The theoretical framework suggested by Tornatzky and Fleischer provides a useful starting point to look into technological innovation process. It allows us to evaluate the importance of different factors which affect the propensity to develop open systems. However, as suggested, IS innovations can be of different types—some may affect the technical IS core only, but some may have influence on the whole organization (Swanson 1994). Community source is unique since it combines service centricity with collaborative open source. Care must be taken when applying this framework to study the links between contexts and project development discussed in the framework.

Based on the TOE framework, a research model tailored for community source is developed and depicted in Figure 5.1. The model is called the "Community Source Success Model" (CSS model). In the model, we posited eight successful factors for community source projects within four contexts. It is worthwhile to mention here that, during the development stage of a community source project, the financial factor may play an important role. Particularly in our case of Kuali, the project was started with an initial fund from the Mellon Foundation and mainly funded later by the development partners. In the existing TOE framework, financial context is not explicitly modeled although financial factors are examined in organizational context. In this study, we extend the TOE framework by adding the financial context to emphasize that multiple institutions share financial resources in community source development. The research framework is a useful analytical tool for identifying and testing successful factors in a community source project.

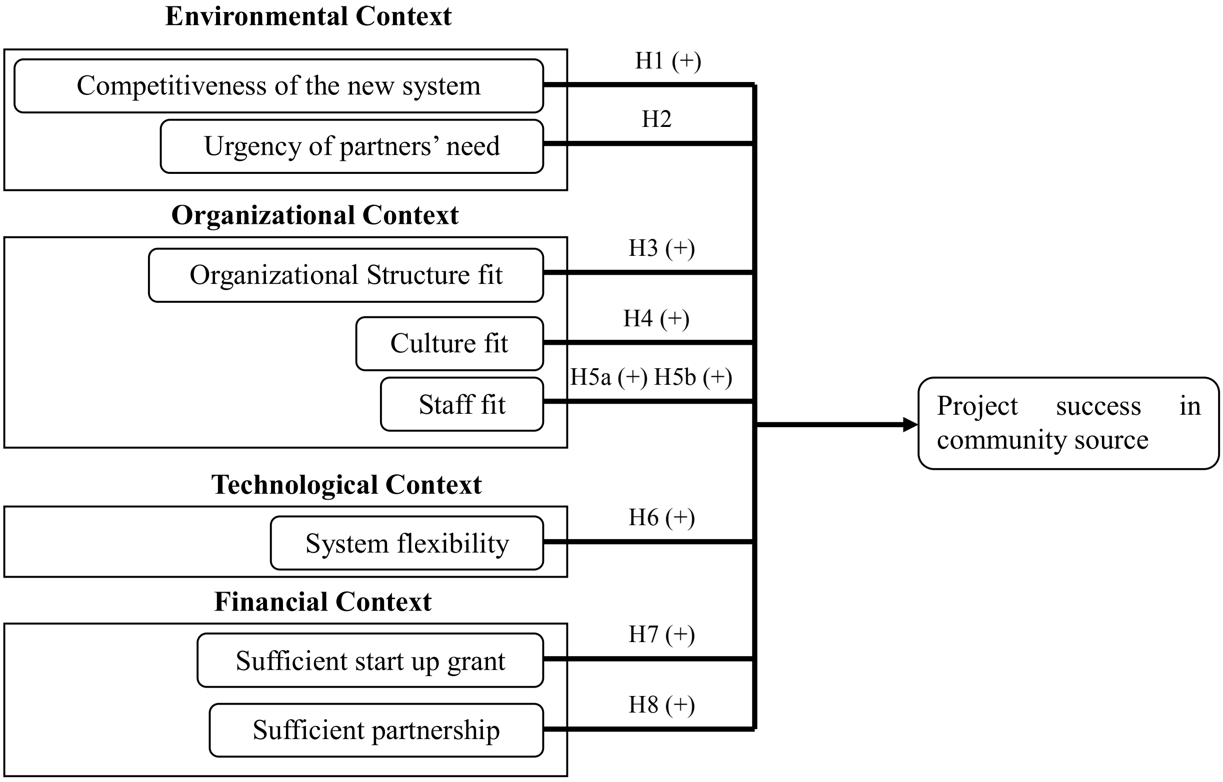

Figure 5 1 A Research Model for Community Source Project Success

5.2.1. Project Success in Community Source

The criteria of the project success for community source can be identified in the following ways: (1) The development cost for individual institution is reduced. (2) The risk of project failure is minimized. (3) The development time is reduced. (4) The system is deployed by certain number of institutions. Based on the above criteria, we argue that the community source project will have high possibility to succeed.

Project management in community source includes two phases, development and deployment. The organizational strategy in achieving success in resource dependency theory (RDT) is defined as organizations maximizing their power in resource control

(Pfeffer 1981). In community source, success can be better achieved when the community source foundation maximizes its power over resources in project development. To make its project successful in the development stage, the community source foundation needs to attract a sufficient number of appropriate institutions to become the development partners. However in the deployment stage, the community source foundation needs to bring as many institutions as possible to become the deployment partners to make the community source success. While we focus on examining project success in community source development in this study, project success in community source deployment can have an impact on project development since successfully deploying KFS can make the Kuali Foundation more trustworthy, and therefore, more universities might want to join the development of more software components in the future.

5.2.2. Environmental Context

In TOE framework, the external environmental context is the arena in which an organization conducts its business. This includes the industry, competitors, regulations, and relationships with the government. These are factors external to an organization that present constraints and opportunities for technological innovations (DePietro et al., 1990). Environment context is described as the arena in which the collaborative open source system is developed in community source. “Competitiveness of the new system” and “urgency of partners’ needs” are major variables in the innovation process of developing community source systems.

Competitiveness of the new system

Current situation is that institutions are not satisfied with the existing systems they acquired from the commercial vendors. However, most of institutions lack of resource to develop the systems by themselves. The motivation for institutions to share resources is to develop a new system which should be much more competitive than the existing system.

Competitive market forces are considered as an environmental factor in the TOE framework (DePietro et al., 1990). Many studies prove competitiveness as an adoption driver (Crook and Kumar, 1998; Grover, 1993). In our research model, we define competitiveness as the advantage of the new system developed in the community source project. For example, Kuali systems appear because universities are not satisfied with their current financial systems. Universities look for a competitive new system which can be acquired with a reasonable cost and in a relative short time cycle. The Kuali systems must be competitive enough to attract universities to join and develop this project. A system that is lack of competitive may end up with no incentive to be developed.

These viewpoints suggest the following hypothesis:

H1: Competitiveness of the community source system in comparison with the old system is positively associated with project success in community source.

Urgency of partners' needs

Institutions that want to deploy the community source software earlier are more likely to become partners because the impact of time is great to the institution (Liu, Zeng and Zhao., 2008). Development partners gain more power by being able to deploy the system sooner. This observation became known to us while we interacted with some Kuali developers. In fact, this advantage makes a big difference when institutions decide whether or not to become a development partner. The above viewpoints lead to the following hypothesis:

H2: Urgency of partners' needs is positively associated with project success in community source.

5.2.3. Organizational Context

In TOE framework, the organizational context describes the characteristics of an organization. Common organization characteristics include firm size, degree of centralization, formalization, complexity of its managerial structure, the quality of its human resources, and the amount of slack resources available internally. It looks at the structure and processes of an organization that constrain or facilitate the adoption and implementation of innovations (DePietro et al., 1990).

In community source project, multiple institutions closely collaborate with each other to develop systems in a virtual organization, which consists of the community source foundation and the development partners. The difference between community source development and traditional IS development is that there are lots of inter-organizational interactions during the community source development stage. How to effectively share resources and work in a team become critical issues to make the project success. The community source project needs to maintain a supportive organizational environment in order to facilitate communications and collaborations among the development partners. “Organizational structure fit”, “culture fit” and “staff fit” are considered as major variables in organizational context.

Organizational structure fit

A good organizational structure will give community source better control of its resources. In community source, multiple institutions form a virtual organization. Even if institutions trust each other, problems will arise in the course of collaboration. Hierarchy is certainly one of the solutions for settling disputes (Williamson, 1975). Ostrom (1990) pointed out the importance of reciprocity norms and Kogut (2000) stressed the importance of rules of behavior. A fit structure can help enforce norms and rules in the organization. As illustrated previously, the Kuali organizational structure is a form of hierarchy which enables the work coordination among development partners in a virtual organization as well as helps monitor and resolve potential organizational conflicts. The partner institutions are more willing to cooperate when the organizational structure fits the needs of the community source development. Thus, the following hypothesis is generated:

H3: Organizational structure fit is positively associated with project success in community source.

Culture fit

In community source, multiple institutions share resources in a virtual organization. Since each institution has unique culture background, culture fit is critical for institutions to collaborate well to develop the system.

Researchers have argued that globally distributed information systems development is situated within a complex and multileveled socio-cultural context, which may range from national (societal), regional, organizational, professional (functional) levels, to team level (Dafoulas and Macanlay, 2001; Karahanna et al., 2005). In general, there are two different conceptualizations of culture: the dimensional view and the emergent view. The dimensional view of culture depicts culture as values, attitudes, and norms shared by a group of people, which are relatively stable and influence how people behave (Avision and Myers, 1995). The emergent view of culture depicts culture historically situated, emergent and contested, which is negotiated and constantly interpreted and re-interpreted in social relations and interactions (Myers and Tan, 2002). In this study, we define culture in the dimensional view. Culture fit in community source means that multiple institutions share the same attitude and norms when they develop a system together.

The selection process of the partners is critical to the success of the community source project since the community source project requires formal collaboration among partners and depends on distributed trust. In the Kuali organization, there is a set of initial investors that come together to support Kuali. The Kuali foundation has to be very careful when selecting additional development partners that have the ability and the right culture fit. An institution can be a negative fit when its values are not aligned with other institutions or its staff do not work well with others. Often this is an issue of "fit" instead of being good or bad. Thus, the Kuali foundation has been very careful to add more development partners since the project was initiated and the team was built.

Based on the above discussion, the following hypothesis is posited:

H4: Culture fit among the development partners is positively associated with project success in community source.

Staff fit

According to RDT, the staffing problem needs to be resolved in order to give community source better control of its human resources. There are two types of staff in community source: functional staff, and technical staff. They are playing different roles to make sure the project work smoothly. To make the project successful, it is important for the community source organization to have the right staff, which we refer to as "staff fit".

Nevertheless, the community source software is built by the functional and technical staff. The right combination of staff is critical to the planning and execution of software development. As such, the following hypothesis is proposed:

H5a: Functional staff fit is positively associated with project success in community source.

H5b: Technical staff fit is positively associated with project success in community source.

5.2.4. Technical Context

In TOE framework, the technological context relates to the technologies available to an organization. Its main focus is on how technology characteristics themselves can influence the adoption process (DePietro et al., 1990). In community source, the technological context mainly focused on how technology characteristics can influence project success in development. "System flexibility" is considered as a major variable in this context.

System flexibility

The modern IT enables more flexible technologies such as XML, SOA and workflow that allow partner institutions have more control about the system, which extends the understanding of RDT in the context of community source.

As a community source organization, Kuali must balance among various requirements from many universities. Institutional partners influence the future of the community and its projects by using their voting rights to determine the development priorities. Therefore, community source has more stringent requirements in system flexibility in order to deal with diverse requirements from its partners. It is imperative to make the Kuali system flexible in terms of both customizability and extensibility. In order to ensure system flexibility, Kuali system utilizes a service-oriented architecture to allow system modification and extension. At the core of the flexibility endeavor are five types of customization including label customization, document type updates, workflow modeling, code modification, and addition of new modules (Liu, Wang and Zhao, 2007).

These viewpoints suggest the following hypothesis:

H6: System flexibility is positively associated with project success in community source.

5.2.5. Financial Context

In community source, institutions get together to share both human resources and financial resources. Financial resources are the key resources in community source project development.

Financial resource available to the organization is included in the organizational context of the TOE framework. This factor has continued to receive attention and empirical support in the IS adoption literature. "Sufficient start up grant" and "sufficient partnership fee" are examined as major variables in this context.

Sufficient start up grant

In a traditional system collaboration model, universities joining together to develop a system will not receive external funding. However, external funding is one of the unique features of community source projects. The Kuali foundation received a grant of $2.5 million from the Andrew W. Mellon Foundation to start up the project. Universities that joined the Kuali foundation can benefit from this external funding. The external funding is the incentive for universities forming the community source organization. On the other hand, larger amount of external funding is not always good for the community source project. If the amount of external funding is too small, there will be not enough incentive for universities forming community source networks. If the amount of external funding is too large, universities can make little initial investment to join community source, which decreases universities' commitment to the project. When the potential loss resulting from the failure of the community source project is small to partner universities, the effort for the partner universities to ensure success of project will decrease. In Kuali, $2.5 million external funding was sufficient to start up the project, which makes universities feel more secure to join Kuali. Therefore, it is reasonable to hypothesize that:

H7: Sufficient start up grant is positively associated with project success in the community source.

Sufficient partnership fee

Community source is primarily funded by the partner institutions. The community source project requires significant investments from those partner institutions. The critical decision for each institution is whether or not it should pay a significant amount of partnership fee in order to join the development community. Since the final product of the development community will become open source, institutions who do not participate in the development stage can still deploy the system for free later. Therefore, the partnership fee needs to be set up in the right level to achieve sufficient fee. It will be difficult to get enough development partners if the partnership fee is too high, while it will be hard to get sufficient funding from the institutions if the partnership fee is too low. With lacking of cash and human recourses, the project cannot survive. Based on the above discussion, the following hypothesis is posited:

H8: Sufficient partnership fee is positively associated with project success in community source.

5.3. The Research Methodologies

5.3.1. Data and Sample

An explorative study was conducted to investigate the variables in the above model. In this explorative study, both a pilot survey and qualitative interviews were conducted.

5.3.2. Survey

The pilot survey was conducted on one of the Kuali Days. The Kuali Days conferences are held twice per year starting from 2005. There were around 450 attendees in the conference while the attendees were around 200 in the Kuali Days 2005. It clearly indicates that Kuali is getting more and more attention from potential development and deployment partners.

We randomly selected one third (150) of attendees to answer our questionnaires and leave the rest 2/3 of attendees for future large scale survey. Of the 150 individuals who received the questionnaires, 50 individuals responded. The return rate is 33%.

The questionnaire consisted of three parts. In the first part, respondents were asked about the background information, such as the university size, the university status with Kuali (development partner, deployment partner or observer), and the respondent's role in the Kuali organization. In the second part, we asked respondents to rank the top three factors in each context (environmental context, organizational context, technological context and financial context). The data generated from the second part helped us check if the proposed model missed any factor that has impact on the success of the community source project. In the third part, respondents were requested to evaluate the importance of each factor in the KFS project. A five-point Likert-scale was used in the choices, where 1 denotes "Strongly disagree" and 5 denotes "Strongly agree".

In the first part of the questionnaire, we ask several questions to understand the background of respondents. The detail questions are listed as follows: "How many students are there at your university?", "What is the plan at your university for replacing the existing financial system with KFS?", "What is the current status of your institution with the Kuali projects", "Are you involved with the development activity of KFS development? If yes, what is your main role in KFS development?" The survey data shows that most respondents are from medium to large universities. This result is not surprising since it is hard for small universities to join the Kuali project due to their limited resources. 46% of respondents indicate that their universities are going to implement the system in 2 to 3 years. 54% of the respondents replied that their universities are development partners of Kuali while 2% of the respondents stated that their universities are deployment partners of Kuali. Since all the development partners will become the deployment partners in Kuali, the deployment partners represent the universities which only join in the deployment of Kuali without joining the development of Kuali. 13% of the respondents indicated that their universities are observers, which shows that there are still a certain amount of universities are in the decision making process. The survey data also shows that half of the respondents were involved in the development activities of the Kuali project. They took the main roles of executive, manager, financial staff, functional staff and technical staff in the development. The data generated from this survey represents the views from various perspectives.

5.3.3. Interviews

Fifteen in-depth interviews were conducted during the conference to get feedback about the research model, the variables affecting project success, and the measurement of these factors. All interviewees were selected from the Kuali partner institutions. Most of them have extensive involvement in development of the Kuali system and play different roles (executive members, financial staff members, functional staff members, technical staff members, etc) in the project. Most of interviews lasted about 30 minutes. Three interviews lasted 1to 2 hours.

5.4. Quantitative Analysis Results

The quantitative analysis is conducted based on the data generated from the survey. The Descriptive Analysis is displayed in Table 5.1 and the inter-construct correlations are shown in Table 5.2.

The coefficient of “competitiveness of new system, structure fit, culture fit. functional staff fit, start up grant and partnership fee” approached significance at the 1% level. The coefficient of “urgency need, technical staff fit, system flexibility” approached significance at the 5% level. Support was found for all hypotheses.

	Number	Minimum	Maximum	Mean	Std. Deviation
KFS success	50	2	5	4.30	0.707
System competitiveness	50	1	5	4.12	0.940
Urgent need	50	3	5	3.92	0.724
Organizational structure fit	50	2	5	4.10	0.743
Culture fit	50	2	5	4.14	0.756
Functional staff fit	50	2	5	3.58	0.731
Technical staff fit	50	2	5	3.54	0.930
System flexibility	50	2	5	3.52	0.614
Start up grant	50	3	5	4.29	0.743
Partnership fee	50	1	5	3.45	0.914

Table 5 1 Descriptive Statistics

	Pearson Correlation
Project Success* Competitiveness of New System	0.682**
Project Success * Urgency Need	0.287*
Project Success *Structure Fit	0.538**
Project Success *Culture Fit	0.569**
Project Success *Functional Staff Fit	0.525**
Project Success *Technical Staff Fit	0.307*
Project Success * System Flexibility	0.291*
Project Success *Startup Grant	0.479**
Project Success *Partnership Fee	0.410**

** Correlation is significant at the 0.01 level (2 – tailed)

* Correlation is significant at the 0.05 level (2 – tailed)

Table 5 2 Inter-construct Correlations

In the survey, we also ask respondents to rank the top three variables in four contexts respectively. For each context, we gave the list of 6 to 10 variables and ask respondents to check the most important three variables in this context which can contribute to the success of KFS project development. The top three variables in each context are described from Table 5.3 to Table 5.6. Frequency in these tables means how many times this variable was checked by respondents. Since the sample size is 50, the sum of frequency for each table should be 150. This ranking can help us to justify variables in current CSS model and test the new variables in the future study.

The top three variables in environmental context are "KFS is competitive in terms of functionality", "KFS is competitive in terms of ease of use" and "KFS is competitive in terms of maintenance"(Table 5.3). All top three variables describe the variable

Variables	Frequency
KFS is competitive in terms of functionality	**43**
KFS is competitive in terms of ease of use	**32**
KFS is competitive in terms of maintenance cost	**29**
Affordable migrating cost	20
Replace in a urgent	15
Others	2

Table 5 3 Top Three Variables in Environmental Context "competitiveness of new system". Variable "Urgency of partners' need" is not ranked highly, which means the respondents do not think "urgency of partners' need" is very important variable affecting project success in community source.

The top three variables in organizational context are "Separate the functional and technical council", "The Kuali Foundation has effective leadership" and "The development partners are sharing the same value and norm" (Table 5.4). Variable "Separate functional and technical councils" is consistent with variable "organizational structure fit" in CSS model. Variable "the development partners are sharing the same value and norm" is consistent with variable "culture fit" in CSS model. Variable "the Kuali Foundation has effective leadership" is not studied in current CSS model.

Variables	Frequency
Separate functional and technical councils	29
The Kuali foundation has effective leadership	27
The development partners are sharing the same value and norm	25
The information is transparent inside the Kuali organization	23
The developers are well trained	18
Appropriate number of development partners	11
Appropriate composition of staff in Kuali organization	10
Appropriate number of staff in the Kuali organization	5
Others	1

Table 5 4 Top Three Variables in Organizational Context

The top three variables in technological context are “the new system is flexible enough for deployment”, “Other” and “The development teams have strong technical support” (Table 5.5). Variable “the new system is flexible enough for deployment” is consistent with variable “system flexibility” in CSS model. Variable “the development team has strong technical support” is not studied in current CSS model. Variable “others” needs to be analyzed for possible inclusion.

Variables	Frequency
The new system is flexible enough for deployment	**42**
Other	**36**
The development teams have strong technical support	**34**
The KFS will provide good system documentation	17
The RICE architecture is based on SOA and workflow	13
The financial documents are XML-based	2

Table 5 5 Top Three Variables in Technological Context

The top three variables in financial context are “the project starts with a proven system such as the IU financial system”, “The Mellon Foundation Grant is important” and “The partnership fee is set at a reasonable level” (Table 5.6). The reason why “the project starts with a proven system” is put in the financial context is that a proven system in community source helps control the development budget. In Kuali, Indiana University contributed their financial system as the base system for development, which largely decreases the development cost.

Variables	Frequency
The project starts with a proven system such as the IU financial system	**34**
The Mellon Foundation Grant is important	**31**
The partnership fee is set at a reasonable level	**25**
A sufficient number of partners are large universities	18
Some partner universities have been contributing generously	18
The partner universities have sufficient IT budgets	15
Others	2

Table 5 6 Top Three Variables in Financial Context

Variable "the project starts with a proven system" was not studied in the current CSS model. Variable "the Mellon Foundation grant is important" is consistent with variable "sufficient start up grant" in CSS model. Variable "the partnership fee is set at a reasonable level" is consistent with variable "appropriate partnership fee" in CSS model.

5.5. Qualitative Analysis Results

The qualitative analysis is developed based on the in-depth interviews we conducted in Kuali Days. We use the results generated from these interviews to better understand the survey results and further validate the finding we received from the survey.

5.5.1. Environment Context

Competitiveness of the new system

Before open source became a viable alternative, there were two primary approaches for organizations to acquire software—self-building and purchasing commercial software. Cost effectiveness is a big concern when organizations consider building software. Availability and flexibility of software meeting unique business need is another issue when organizations turn to the commercial market. People in our interviews indicated their expectations of the new system developed from the community source project, which may have advantages over those systems obtained through traditional ways.

First, several interviewees indicated that "*cost was the main issue*" when their institutions decided to join the KFS project. "*[We expect the KFS project] can [help us] reduce the costs [of system development, implementation and maintenance]*." "*If we re-write the code by ourselves, it is very expensive. [Instead, the KFS project offers us a] cheaper alternative.*"

Second, when considering commercial software, the participating institutions always found that "*there are fewer alternatives of commercial software on the market.*" Even if they did implement commercial software, the institutions still had to deal with several issues. For example, one interviewee complained that "*current vendors are not responsive, [but they still] charge more money for less service.*" Another interviewee said, "*Commercial software [always] needs to be upgraded, and you might not need some features.*"

Third, most interviewees believed the new system developed in the KFS project would fit into the higher education industry better than the "generic" commercial software did, because "*[the system] is built by people [working] in higher education. We understand what we need.*" "*[The system] focuses on higher education … and [we] can share [the system].*" On the other hand, "*there is no motivation for Oracle to design software especially for higher education.*" An interviewee further stated, "*Oracle, PeopleSoft, and SAP did not do a good job [in developing systems for higher education]. "One [piece of] commercial software for everyone does actually not work well in higher education; instead, [the KFS system] works well.*"

Lastly, a developer among the interviewees pointed out the problem of open source. Although he recognized "*open source will be the mainstream in the future,*" the developer argued "*[current] open source [software] cannot be accepted, [because it is] not reliable nor in good quality and its maintenance cost is high.*"

This interview results for variable "competitiveness of new system" is consistent with the survey results and support hypothesis 1, which postulates that competitiveness of new system is positively associated with project success in community source.

Urgency of partners' need

Different development partners had different levels of urgency to have the new system. An interviewee mentioned the People student software that many universities were using. Although the software was not good, the universities still implemented it since "*they cannot wait*". Another interviewee indicated that "*[our university] is not in an urgent to* *replace [our current system], but [our university] needs to do it anyway.*"

From the interviews, we found inconsistent answers for variable "urgency of partners' need". The top three variables ranking in the quantitative analysis also shows that this variable is not ranked highly. We need to further examine this variable in the second survey.

5.5.2 Organization Context

Organizational structure fit

Several members pointed out "*good representation*" and "*right mix of private, public, large, and small*" universities among development partners were important for the KFS project to be success. Furthermore, a balance of large and small universities on their contributions to the project development and making them work together were also important to project success. All development partners should have strong commitment to the project.

"*The number of development partner*" is also a factor affecting project success. Many interviewees indicated that increasingly large number of developers due to a large and growing number of development partners imposed management pressure on the project. The growing number of partners also led to the functional council getting bigger and bigger. "*It is [thus] getting hard to manage.*"

This interview results for "organizational structure fit" is consistent with the survey results and support hypothesis 3, which states that organizational structure fit is positively associated with project success in community source.

Culture fit

An interviewee indicated that "*universities have a collective [organizational] culture … [therefore] a hybrid model*" was necessary for them to work together to develop software. Community source fits well such a hybrid culture environment. Another interviewee further indicated that, even in this community source project, "*it is challengeable to make multiple [development partners] which have different cultures work in a team. How to balance their commitments becomes a key issue.*" Moreover, "*every institution has different need.*" It is thus important for the development partners to work as a team, even when some system functions are not needed by some partners.

The organizational culture inside the community source project itself could also affect project success. People mentioned several cultural factors during the interview. First, "*establishing work relationship is very important"* for project success. Second, "*Passion [of different development partners in the project] is very important*" for project success. One interviewee gave an example of a previous project, which was not successful because the participants' passion could not be sustained. Third, there should be a professional, face-to-face venue for people to discuss the software development, rather than a relatively casual platform in open source projects.

This interview results for "culture fit" is consistent with the survey results and support hypothesis 4, which postulates that culture fit is positively associated with project success in community source.

Staff fit

The interviewees talked about a number of staff issues which related to project success. First, several interviewees consistently pointed out that finding sufficient software developers with good quality "*is a big issue*" for project success. The fact is all developers were contributed by the development partners. Therefore, the quantity and quality of the developers working on the system were limited by the development partners' own human resource. What the community source project can do is making sure all developers have a commitment to the project and collaborate with each other as a team. In order to achieve this goal, the community source project "*needs to better educate developers and programmers. The training program needs to be more organized [and] we need to assign somebody [specifically] responsible for training program.*"

Many interviewees also mentioned the problem of "*high turn-over rate of developers*". They believed it had a negative impact on project success. In order to solve this problem, some interviewees suggested outsourcing the software development function. Meanwhile, the community source project needs to help new developers joining the project get "*into the team and get involved in the development quickly.*"

This interview results for "staff fit" is consistent with the survey results and support hypothesis 5a and hypothesis 5b, which indicates that staff fit is positively associated with project success in community source.

Leadership

Even though it was not included in our original survey, several interviewees considered the leadership, especially the support from top management, as an important project successful factor. "*Board members need to engage and contribute more in the project [in order to ensure the success]. Right now, the project relies too heavily on the functional council.*" When mentioning the functional council, most people recognized it played a very important role in the project. "*Our functional council [and its leadership] is very effective*." "*[It] makes the difference.*"

In addition, one interviewee suggested the leadership of the project should have a "*very clear development scope for [the project] ... and [be] very careful not to develop the system out of scope ... Fixed scope is critical.*" "*Sometimes the functional council is not sure what they want,*" another interviewee added.

Variable "Kuali foundation leadership" was mentioned as an important variable to project success by many interviewees. From the quantitative analysis, Variable "Kuali foundation leadership" is also ranked as one of the top three variables in organizational context. We will examine "Kuali foundation leadership" as a new variable in organizational context and test it in the second survey.

5.5.2. Technological Context

System flexibility

When being asked about the technical factor in project success, many interviewees responded with "system flexibility". "*System flexibility [is an important successful* *factor]*". "*Actually, with flexible system, universities can work on different modules [and] build common system together … [The KFS system] is very configurable.*" When the system lacked flexibility, it was hard for participating universities to implement it later. Other system-related factors for project success mentioned by the interviewees included "*[system] integration ability*", "*[system] accessibility*" and "*[easiness of system] implementation*".

This interview results for "system flexibility" is consistent with the survey results and support hypothesis 6, which indicates that system flexibility is positively associated with project success in community source.

Technical capability

When turn-over rate of the developers was high and the quality was uneven among the developers, several interviewees pointed out it was very important to document software development, conduct code review, and provide technical support to new developers.

Variable "technical capability" was mentioned as an important variable to project success by several interviewees. From the quantitative analysis, Variable "technical capability" is also ranked as one of the top three variables in technological context. We will examine "technical capability" as a new variable in technological context and test it in the second survey.

5.5.3. Financial Context

Sufficient start-up grant

One interviewee indicated "*part of the reason [why the previous project was not successful] is because of lacking funding.*" The start-up grant offered by the external foundation "*is very important*" for the project. The external foundation "*brings [not only] cash [but also] creditability support [to the project]. [They make] it easier to attract universities to be partners.*"

This interview results for "sufficient start up grant" is consistent with the survey results and support hypothesis 7, which states that sufficient start up grant is positively associated with project success in community source.

Sufficient partnership fee

One interviewee admitted their financial resources were very limited. However, the KFS project allowed all development partners to pool the resources together by asking each one to contribute $24,000 cash per year.

This interview results for "sufficient partnership fee" is consistent with the survey results and support hypothesis 8, which postulates that sufficient partnership fee is positively associated with project success in community source.

A proven base system

Many interviewees pointed out that the KFS project was successful partially due to an existed base system contributed by one of the development partners. This base system saved a lot of financial resource and efforts when they developed the new system.

Variable "the project starts with a proven system" was mentioned as an important variable to project success by many interviewees. From the quantitative analysis, Variable "the project starts with a proven system" is also ranked as one of the top three variables in financial context. We will examine "the project starts with a proven system" as a new variable in financial context and test it

5.6. Findings

The survey data collected from 50 respondents support our research model and thereby validate our theoretical development for the study. There are three important findings from the survey study. First, eight factors in four contexts are significantly related to the project success in community source. Second, Four additional factors are found as important factors affecting the project success in community source. The four factors include "the Kuali Foundation leadership" in organizational context, "technical capability" and "technical support" in technological context, "the project starts with a proven system" in financial context.

The data collected from the 15 interviews generated the following findings. First, "Urgency of partners' need" is not described as an important factor affecting the project success in community source. Second, three additional factors we generated from the survey result "the Kuali Foundation leadership", "technical capability" and "the project starts with a proven system" are proved as important factors for community source project success.

The results of the survey and interviews based on the research model help us validate the variables that affect the success of the Kuali project in development. This is a timely study since the Kuali financial system was just released, and therefore it is important to know how people think about the most important factors and their relevancy to its success. The institutions need to make decisions for further actions. This provides an ideal opportunity for us to survey Kuali constituents about how to make the Kuali project more successful.

Based on this exploratory study, we will modify the variables in four contexts and justify community source success model. Four additional variables "the Kuali Foundation leadership", "the project starts with a proven system", "technical capability" and "technical support" will be added into current CSS model and tested in the second survey. "Urgency of partners' need" will be further observed in the second survey. Measurement items will be developed for identified factors and the adjusted CSS model will be empirically tested in community source.

5.7. Conclusions

5.7.1. Summary

Achieving project success is critical to community source since it typically consists of a sequence of projects as in the case of Kuali. In this study, we examined the TOE framework in light of the resource dependency theory. The Kuali case helps us better understand how RDT can be used to understand many issues related to community source. Basically, individual institutions lacks of resources to acquire the customized system alone, and therefore community source provides the platform for multiple institutions to share resources and develop systems that can meet their needs. We apply the TOE framework in assessing project success in community source and justify variables based on RDT. Survey and interviews are conducted to test the research model and hypotheses.

5.7.2. Limitations and Future Research

This research has several limitations. First, our study only investigates the factors affecting the likelihood of project success in development and deployment. To gain a better picture of the community source project success, technology diffusion should also be examined as a next step. Second, the dataset will be generated from the higher education sector. However, universities are quite different from commercial companies, and therefore, the research result may not be directly applicable to other industries. Accordingly, one future research direction is to design longitudinal study examining the diffusion of community source approach, and another direction is to study community source projects outside the higher education sector as they become available.

5.7.3. Implications for Research and Practice

This study has potential implications for research and practice. First, better understanding community source provides another opportunity to investigate about software development approaches including commercial software and open source development. Community source is a new approach to system development. However, community source combines features from both commercial software and open source. This study provides useful insights for further research in system development approaches among commercial software, community source, and open source. Second, it will lead to guidelines for strategic planning and decision making in institutional software acquisition. Institutions face lots of uncertainty when they make the decision to join the community source. This study provides the baseline for institutions to judge if community source is the right choice for them. Third, our research will also help managers of community source better plan and manage their software projects. Knowing the right project success model can guide the management to make project management more successful.

CHAPTER 6. THE MANAGEMENT PERSPECTIVE - OUTSOURCING

6.1. Introduction

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, 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. In order to understand these challenges better, we studied the Kuali case, which offered us a great opportunity to study the project management issues of community source in a higher education setting. Based on this study, we suggest that global outsourcing of software development is a viable solution to those issues we identified in the community source project. The primary research questions of this study are: "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?"

6.2. Economic Effect of Community Source Approach

In this section we discuss the economic effects of the community source approach on the Kuali project.

6.2.1. Unique Costs/Benefits of In-house Development by Individual Institutions

As illustrated in Figure 6.1, when institutions (***I****i*) develop their own applications, they each spend certain amount of costs (***C****i*) in terms of human resources and capital spending in order to derive the needed benefits (***B****i*). In the case of Kuali, the number of universities (n) is a reasonably number of institutions to coordinate and a reasonably large number of institutions to reap the collaboration benefits. Since each institution has its unique cost structure and economy of scale, each institution has its own unique application requirements, spends different costs, and derives particular benefits.

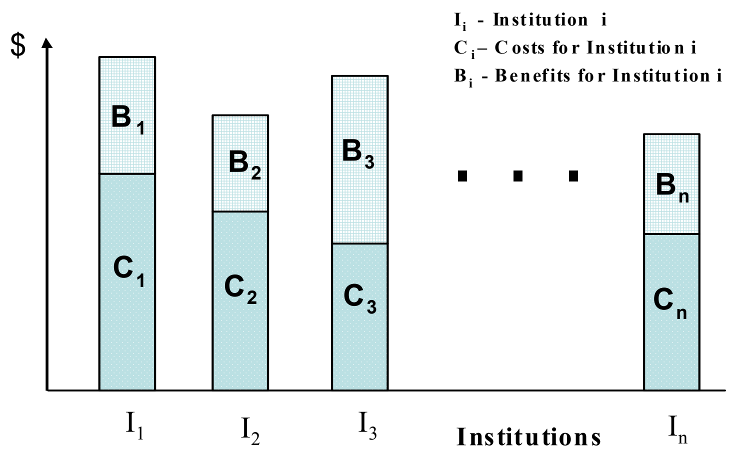

Figure 6 1 Costs and Benefits of In-house Development of Individual Institutions

6.2.2. Opportunity of Sharing Costs and Reducing Risks

Assuming the institutions have similar, but not identical, application requirements, there is clearly an opportunity of pooling the resources together towards the common objective – to develop sharable applications at a fraction of the costs that each individual institution would have to bear. This is exactly the motivation of community source, which was demonstrated in the Kuali case. Furthermore, in order to unify and share the tasks of system analysis, design, and development, the development partners should have common application requirements and development philosophies.

6.2.3. Technology Flexibility is the Enabling Engine for Community Source

The Internet, workflow, and service-oriented architectures together provide the much needed technology infrastructure that meet the application requirements of partner institutions. No matter how similar the applications are among partner institutions, individual variations are inevitable. As such, it is not possible to develop a single ten-key solution to all partner institutions. That is, the underlying technology and system infrastructure must be flexible and customizable. In the case of Kuali, a service-oriented architecture with embedded workflow modeling capability was used to allow five-levels of customization. This technology flexibility is an important requirement.

6.2.4. Cost Reduction by the Community Source Approach

Let the institutions shown in Figure 6.1 form a community source; we assume that each institution pays the same partnership fees and that the partnership cost is the only cost for all institutions; we then have a new cost and benefit chart as shown in Figure 6.2. The benefits remain the same as in Figure 6.1, but the costs have been reduced to an equal amount among all partner institutions. Clearly, the institution I_3 gains more than other institutions since it maintains the same benefits but gains a larger reduction in costs.

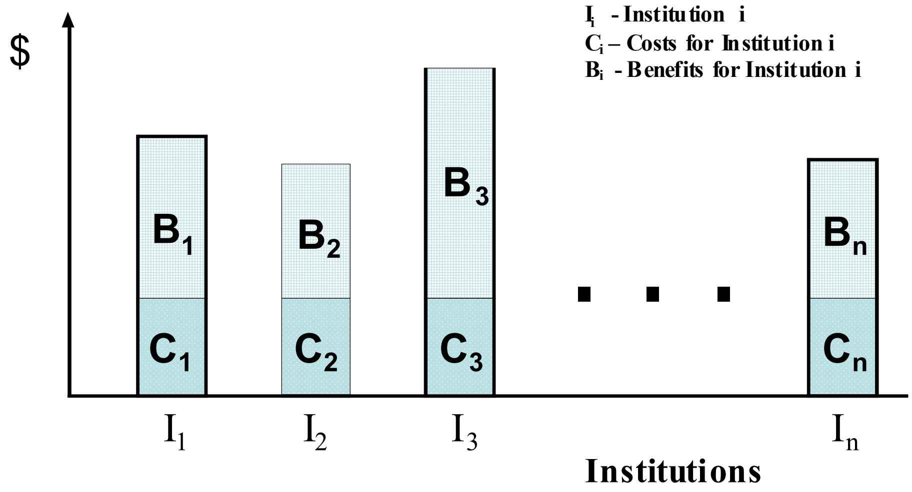

Figure 6 2 Costs and Benefits of Community Source Development of Partner Institutions

6.2.5. Economic Effect of the Project Triangle

The comparison between Figure 6.1 and Figure 6.2 shows the obvious economic benefit of community source, i.e., a significant reduction of overall costs due to cost sharing among partner institutions. However, a separate economic effect of community source that is not shown in Figures 6.1 and 6.2 has to do with the project triangle in software development, which consists of scope, cost, and schedule. For a given project, once the scope and the costs are fixed, the schedule must also be fixed, and vice versa. Since the community source approach allows each individual institution to afford much large cost, the scope can therefore be increased and the schedule be reduced significantly. That is, the community source approach dramatically improves the development capability of the individual institution.

6.2.6. Enhancement of the Open Source Approach via Community-based Development

The Kuali case shows that the Andrew Mellon Foundation was able to promote open source by providing a seed money for the Kuali project, which in turn attracted significant amount of funding from the development partners. This is an interesting economic phenomenon since the impact of the initial funding magnified the economic effect of community source in the Kuali case.

6.3. Proprietary Software, Community Source and Open Source

Table 6.1 shows the similarities and differences between proprietary software, open source and community source, three approaches for software development. In proprietary software development, a software company builds the software and sells it to organizations and individuals who use the software. In open source development, people who build the software are volunteer developers and the software is free to use. In community source development, software developers are employees from partner organizations. The partner institutions invest significantly and coordinate with each other in the software development project. The final product of the project is open source and available to each partner.

While proprietary software was described as a cathedrals and open source as a bazaar by Raymond (Raymond 1999), community source can be described as a shopping mall with a virtual organization consisting of partner institutions, each of which commands its own employees. In community source, how to manage the developer team in a virtual environment becomes the key issue to the project success.

	Developer	**End User**	**Characteristics**
Proprietary Software	Employees in third-party software companies	Institutions or individuals who pay for the software	The end users buy the software. The initial investment required for the software development is made by the software company.
Open Source	Volunteer software developers	Institutions or individuals who use the software for free	The software is free for end users. The developers are volunteers.
Community Source	Employees in Development partners	Development partners and other participating institutions	The initial investment required for the software development is shared by the partners. The final product is open source.

Table 6 1 Proprietary Software, Open Source and Community Source

6.4. IS Staffing Factor as an Outsourcing Driver

With regard to IS development, there has been an industry-wide concern about the insufficiency and inferior quality of internal developers and their lagging performance. Many organizations confront a wide array of disparities between the required capabilities and skills to deliver the software and the actual technology capabilities and skills of in-house IS staff (DiRomualdo et al. 1998). Subsequently, academic research has found a positive relationship between IS outsourcing and the discrepancies between the desired and actual levels of in-house IS staff support and quality (Dibbern et al. 2004). When an organization recognizes that it does not have the skills and resources necessary to achieve its goal in software development, outsourcing then becomes a viable solution. Through outsourcing, the organization can close the gap strategically and obtain cost-effective access to specialized software development skills and capabilities (DiRomualdo et al. 1998; Finlay et al. 1999). In fact, this was one of the primary reasons at the beginning of IS outsourcing (McFarlan et al. 1995). In a review of the outsourcing history, Davis et al. (2006) explained that the outsourcing of IT began with the hiring of external professionals to work in areas where organizations did not have sufficient skills and/or enough people to accomplish the software development jobs.

Even if the insufficiency of in-house software development skill is not a primary concern, some companies still outsource their IS development in order to do better those things that they are already doing, utilizing state-of-the-art skills, tools and competencies from external sources and improving the quality of software development (DiRomualdo et al. 1998). As they enter into outsourcing arrangements, organizations discover that external vendors could provide high-quality and skilled technical developers and more flexible and responsive design and development of IT systems (Khan et al. 2004). In general, the technical expertise and specialization of outsourcers result in better and more effective systems at a significantly lower cost than those developed by the internal IS department (Wang et al. 1997).

In addition, some organizations may even have trouble attracting, managing and retaining their in-house IS developers. Outsourcing offers a way to gain those software development skills without getting involved in the complex staff management issues those organizations are not skilled in and even do not want to manage (McFarlan et al. 1995). Outsourcing vendors have the specialty and proficiency at recruiting and managing software developers, which makes them better than internal IS functions/groups (DiRomualdo et al. 1998). Furthermore, leveraging external developers provides organizations the flexibility to hire IS staff only when necessary and for only as long as needed (Jiang et al. 2006). In this way, outsourcing simplifies IS staff management and saves related human resource costs.

6.5. Core Consideration in Outsourcing

After delegating software development management to external vendors, an organization, especially when under financial and human resource pressures, may find itself be able to direct its key resources to and focus its management time and energy on its core activities (DiRomualdo et al. 1998; McFarlan et al. 1995). Actually, the motivation for outsourcing has evolved from a primary focus on cost reduction to an emphasis on improving overall organization performance. This latter objective can be realized by off-loading responsibilities to an experienced external partner and gaining better internal resource deployment and management (Davis et al. 2006; DiRomualdo et al. 1998).

According to the survey and analysis conducted by Dibbern et al. (2004), consideration of IS outsourcing is also a consequence of a shift in business strategy. Many companies have abandoned their diversification strategies to focus on core competencies. Senior executives have come to believe that in order to sustain their companies' competitive advantages they should concentrate on what they do best and outsource the rest (Dibbern et al. 2004). Similarly, after reviewing the history and prospects of IT outsourcing, Davis et al. (2006) argue that an organization should not engage in activities that are outside its core competencies, because others can generally do such functions better, less expensively, and/or faster. Therefore, many organizations have "upgraded" their in-house jobs to focus on "core" activities, with "commodity" activities being outsourced to specialists (Davis et al. 2006; King 2008). For example, while many programming and software development jobs are being outsourcing, new in-house jobs are being created that focus on management and customizing of externally-developed software (Davis et al. 2006).

The "job upgrading" concept is also supported by King's (2008) observations of a number of post-outsourcing organizations. The author argues that the need for some traditional capabilities in an organization will virtually disappear. Among the capabilities that will clearly need to be downgraded are those in systems development, since much of an organization's software will be developed on an outsourced basis and customized to fit the organization's unique needs. On the other hand, a new internal capability, which reflects a deep understanding of its business, operations, goals, priorities, and strategies, is needed (King 2008).

6.6. Global Outsourcing

In today's global economy, offshore outsourcing IT services is simply too compelling to be ignored. Globalization has resulted in billions of people joining the free-market world, and millions of companies involving in world trade of products and services. This trend has produced a world where not only goods are globally tradable, but so is labor and service, which can be sent over a wire rather than physically relocated (Aspray et al. 2006). Offshore outsourcing software development is a well paradigm for globalization of IT service. The main incentive for offshore outsourcing is quality criteria. The ideal outsourcing partner assures high-quality work at low prices and a modern IT infrastructure, and guarantees international quality standards. Another incentive for offshore outsourcing is core competence focus. By offshore outsourcing processes outside its core business, an enterprise can devote itself entirely into value-added activities within its core competencies. This can help to unlock internal resources (Erber et al. 2005). Gupta et al. (2007) pointed out that the key drivers for offshoring will be strategic, not economic. They argued that to reap the full benefits from offshoring and to develop sustainable models, one need to treat offshore vendors as strategic partners rather than as mere low cost service providers. One can employ professionals in multiple parts of the world, perform tasks at all times of the day, and bring new products and services quicker to the market by offshoring. The creation of new globally distributed workforces and global partnerships can lead to major strategic advantages for companies and countries alike.

Outsourcing and offshoring are two independent options, which, if they occur simultaneously, will lead to offshore outsourcing as a specific form of outsourcing (Erber et al. 2005). In this study, we unify the concepts of outsourcing and offshoring into one as “global outsourcing”, which means outsource software development to either domestic or foreign third-party vendors. As we discussed in the previous sub-section of outsourcing, many in-house IS departments lacked the necessary technical talent, management skills, and financial resources for software development projects (DiRomualdo et al. 1998). On the other hand, there was no shortage of computer skills and competencies either from domestic outsourcing or international offshoring (Davis et al. 2006). Furthermore, most third-party software development firms in the U.S. as well as other countries have adopted Six Sigma or other approaches to quality management (Davis et al. 2006). As a result, global outsourcing of software development to supplement or even replace permanent in-house IS staff has increased dramatically in the real world (Ang et al. 2001). Organizations outsource their software development to the external vendors which can provide the lowest-cost but also the best-qualified software developers worldwide (Kakabadse et al. 2005). The direct benefits expected from such practices are improvement of software development and acquisition of better and cheaper technical skills and competencies at a global scale (DiRomualdo et al. 1998). Furthermore, companies will benefit from global outsourcing not just by accessing to better technical skills and expertise to software development, but also by freeing up limited financial and human resources and easing internal management attention on developers (DiRomualdo et al. 1998). Specifically speaking, global outsourcing allows the organization to rely on management in the external vendor to oversee the software development tasks at which it might be at a relative disadvantage, so that the organization can increase managerial attention and resource allocation to those tasks that it does best (Jiang et al. 2006).

In summary, we find from our literature review that 1) many organizations regard outsourcing software development as a solution for the problems of poor performance of their in-house IS staff and lack of skilled professionals and a solution for the issue of in-house IS staff management; and 2) released financial, human and management resources by outsourcing can in turn give organizations an opportunity to concentrate on business analysis, performance improvement, new initiatives, and strategic planning. These two motivations of outsourcing in turn will help organizations gain sustainable strategic advantage. We further find that the main incentive and driver for offshore outsourcing software development are consistent with the ones of general outsourcing. Accordingly, Figure 6.3 illustrates the motivations and expected benefits of global outsourcing of software development.

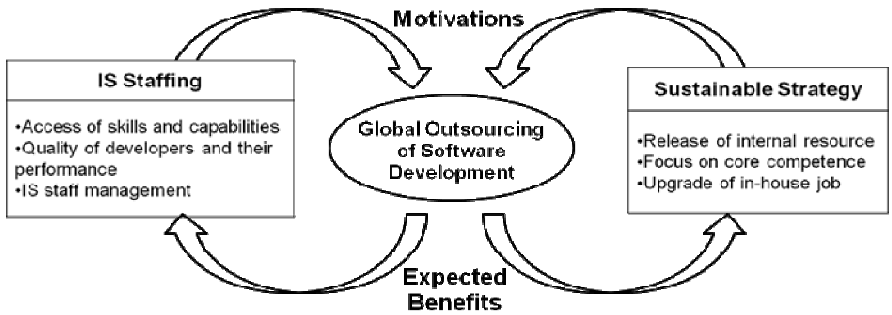

Figure 6 3 Motivations and Expected Benefits of Outsourcing

6.7. Challenges in Community Source Development

6.7.1. Research Methodology

In order to achieve an in-depth understanding of the motivations and expected benefits of outsourcing in community source project, we adopted the qualitative approach of interviewing Kuali development partners on KFS development. Thirteen interviews were conducted during a biannual conference of Kuali Days.

All interview respondents were senior executives, project managers, developers or other

IS staff members from the Kuali's partner institutions. Most of them had extensive involvement in the development of the Kuali project. Since Andrew Mellon Foundation, rSmart Group and NACUBO played important roles as did other partner universities in the Kuali project, we also interviewed managers from these three institutions. Most interviews ran around 30 minutes. Three of them lasted between one to two hours.

The interviews were "structured open-response interviews" (King 1994), in which the formats of interview guideline and schedule were similar to a structured survey. However, most of the questions were open-ended and non-directive. There was flexibility during the interview to focus on particular questions, factual information and evaluative comments, and to skip others, based on the interviewee's personal experience and expertise. For example, we focused more on asking technical issues when we interviewed developers. As a result, the qualitative interview provided a complete descriptive report of interviewees' personal perceptions, beliefs, views and feelings about the KFS development project.

The interview procedure worked as follows. First, we explained to the participants that the purpose of this study. Second, we asked the participants to provide their general impressions about the KFS development. Third, we asked about the problems during the KFS development. Finally, we asked for suggestions to improve the later development of KRA and KS.

6.7.2. Interview Findings

Table 6.2 presents the categories and codes that emerged from the content analysis of all the interview notes. The detailed explanations and findings are reported as follows in a descriptive, narrative and succinct form, categorized into five sub-sections and supported by interview participants' quotations (in *italics* in the following texts). In order to ensure anonymity all respondents' last names were removed in the report.

<table>
<tr><th>Category</th><th>Codes</th><th>Ranking by quote counts</th></tr>
<tr><td rowspan="3">Staffing problem of the community source project</td><td>Managing / management</td><td>1</td></tr>
<tr><td>Quality / performance</td><td>2</td></tr>
<tr><td>Turn-over rate / recruiting</td><td>4</td></tr>
<tr><td>Sustainability problem of the community source project</td><td>Sustainability / focus shift</td><td>4</td></tr>
<tr><td>Outsourcing as a solution for the problems</td><td>Outsourcing</td><td>3</td></tr>
</table>

Table 6 2 Summary of the Interview Analysis Results

Managing Developers

There are nine counts in the interview transcripts that interviewees mentioned the challenge of managing software developers in the community source project. This was the most-mentioned problem during the interview. At the beginning of the project, "*the developers are actually not there; they need to be recruited later on.*" (Steve, project manager) Later on, partner universities contributed their own employees as the software developers for the project. However, there was an "*unbalance problem*" of such human resource contributions between different partner institutions. Tim, a member of the Kuali organization, pointed out the problem that "*some large universities over-contribute to project development, while some small universities cannot contribute enough.*" Some participating colleges even had a "*difficult time to get enough developers [to work on Kuali's projects].*" (Tim, member of the Kuali organization)

Every participating university was required to contribute software developers to the project. As the number of development partners grew, the number of developers also grew "*The increasing number of development partners put pressure to the Kuali organization for project management.*" (Mike, member of the Kuali organization) At the same time, "*the number of developers is [also growing] too large [for such a small organization] to manage effectively.*" (Steve, project manager) Moreover, all the developers worked in different locations at different time, and reported to both their home institutions and the Kuali organization. This was essentially a virtual team that the Kuali organization had to coordinate and manage, but managing such "*a virtual team is not easy.*" (Steve, project manager) It was even more "*challengeable to make [developers from] multiple universities which have different culture to work in a team. How to balance their commitment becomes a key issue.*" (Alex, member of the Kuali organization)

Finally, every time "*when the new development partner joins Kuali, new developers will join the development team… [I]t is not easy [for those new developers] to come into the team quickly and it takes lots of time and effort [for both the developers and the development team].*" (Bob, developer) This point is supported by a statement made by a developer from a new participating university: "*We join Kuali in the late stage. We have difficult time to involve in the project development.*" (John, developer)

The interviews at this point led us to find that while software developers were the most basic human resource for a community source project to perform its job, the creation, the growth and the change of this developer team were out of the control of the project. Managing such a virtual team therefore became an indefinite and intractable problem.

Quality Issue of Developers

The second most-mentioned challenge was the quality of developers. It "*is a big issue. In the future project, I wish we could select high-quality developers.*" (Steve, project manager) George, a senior executive in the Kuali organization, suggested that "*we should hire good quality developers to work for us.*" The quality of the developers working for the project depended largely on the contributing institutions and thus there was a large variance on the developers' quality. One interviewee gave an example, "*it is hard for X University to get good developers since the living expense there is very high.*" (Alex, member of the Kuali organization) Another example was that "*[the developers contributed by] X College ... cannot do the work they promised.*" (Tim, members of the Kuali organization)

Another quality-related issue identified in the interviews was the training program for the developers. "We need to better educate [our] developers… Our training program in the future needs to be more organized." (Alex, member of the Kuali organization) Moreover, "we need to assign somebody responsible for training program [for our developers]. We do not give this specific role in KFS right now." (Steve, project manager)

High Turn-over Rate of Developers

The high turn-over rate of the developers was also identified as a challenge for the project. According to Alex and Mike, members of the Kuali organization, "the current problem for KFS is the high turn-over rate of developers"; "*we have high turn-over rate of developers in KFS. In some universities, the turn-over rate of developers is very high.*" As a sequence, "*the high turn-over rate of developers delays our progress.*" (Steve, project manager) Steve further stated that "*we could be much more effective if we don't need to deal with this issue.*"

Sustainability of the Community Source Project

When we conducted the interviews, KFS was almost completed and KRA and KS were being launched. After learning lessons from developing KFS, and in looking ahead to the future of the Kuali organization, several interviewees identified the sustainability requirement for the organization. At the strategic level, George, a senior executive of the Kuali organization, indicated that "*sustainability for Kuali is our behavior of volition… Right now, the Kuali organization should focus on sustainability.*" At the project level, Steve, a project manager, also identified sustainability as "*a big issue*." Kate, a developer, suggested that "*right now how to sustain the system is very important for us.*" However, "*there is not enough resource for sustaining the system.*" (Steve, project manager)

Outsourcing as a Solution for the Problems

As shown above, we revealed several problems of software development in the community source project. As the Kuali organization moved on to its next software project, Eric, a member of the Kuali organization from its external sponsor, indicated that "*Kuali needs to spend more time to design in KS than that in KFS and KRA. The development problems now will become more serious in KS if the problems are not solved properly.*" Since in-house software development had been shown to have a number of issues with this community source project, the interviewees suggested that outsourcing be a potential solution. "*I think outsourcing developers might be an option for future projects… Outsourcing developers will help the Kuali organization [concentrate its energy to] manage more development partner [institutions].*" (Mike, member of the Kuali organization)

Tim, another member of the Kuali organization, disclosed another motivation of outsourcing—focus shift. "*[From now on] the Kuali organization should focus less on technical issues and more on community issues.*" George, the senior executive of the Kuali organization, also indicated the shift of the organization's strategic focus, which could lead to outsourcing as well. "*During KFS development, the Kuali organization focused on managing the development project. Right now, the Kuali organization should focus on sustainability.*" (George, the senior executive)

6.8. Discussion: Global Outsourcing of Community Source

At the beginning of his interview, George, the senior executive of the Kuali organization, explained that software development experienced three eras: "*[During] the first era, each of us built it; [in] the second era, we bought it…then hammered it…sent more money; [in] the third era, we now built it together [e.g. community source].*" He then posed the question of "*what next*". Based on our literature review on the motivations and expected benefits of outsourcing and our interview analysis on the challenges of the community source project, we proposed that global outsourcing of community source development be an answer to George's question. Furthermore, the results in the Kuali interview analysis specifically lead us to construct a research framework for global outsourcing of community source development, as shown in Figure 6.4. The framework described the relationship between in-house IS staffing problems and global outsourcing and the relationship between global outsourcing and the sustainable strategy of the organization. Based on the research framework, mapping it with the concepts in Figure 6.3, and evidenced by the interview results, we put forward the following seven propositions.

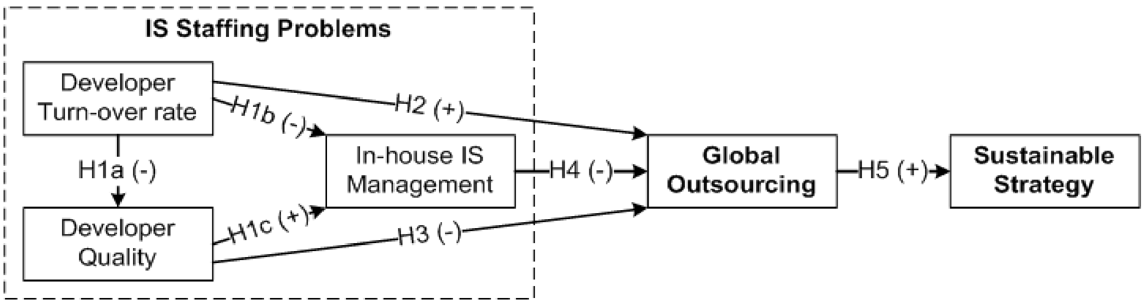

Figure 6 4 A Research Framework for Global Outsourcing of Community Source

6.8.1. The Developer Staffing Problem

Proposition H1a: The higher the turn-over rate of developers, the lower the quality of developers working for a community source project.

Proposition H1b: The higher the turn-over rate of developers, the more difficult in managing the in-house developer team in a community source project.

Unlike the traditional open source project, in which most developers participate voluntarily and are motivated by their own interests, in the community source project, the developers are employees assigned by the development partners. The problem of developers' turn-over rate in community source is therefore contributed by three facts. First, the turn-over rate of IS staff in the home institution itself is high. Due to the high cost of living and/or relatively low salary, some developer might leave the home institution and thus not work for the community source project any more. Second, the development partners might re-assign some developers to other projects and replace them with new developers. Lastly, when the community source organization grows, new partners join the project and a few old partners drop out. Subsequently, over time developers come and go.

The high turn-over rate makes it impossible for the community source project to maintain a consistent quality within the developer team and to manage a relatively stable developer team. However, there is no such a reliable IS human resource in hand for the community source organization to access. The organization has to devote an enormous amount of attention, time, effort, and resources to deal with this developer issue and, consequently the development progress is slowed down.

Proposition H1c: The lower the quality of the developers, the more difficult in managing the in-house developer team in a community source project.

The quality of the developers had been a serious management problem in our community source case. The quality of the developers is not within the control of the community source organization. It depends on the IS human resource that individual partner can access. As the development partners “contribute” their own developers to the project, the quality of the developers varies greatly. In the Kuali case, some universities, especially those small colleges, were even unable to provide competent developers to perform their jobs for the community source project. Moreover, the overall quality of software developers working in educational institutions generally cannot compare with the quality of developers at professional software companies.

The poor quality of the developers not only negatively affects the quality of the software development but also makes the management of such a diverse team very complicated and time- and effort-consuming. In the Kuali case, in order to deal with the quality issue, the community source organization had to allocate a great deal of human and financial resources, especially for a developer training program.

6.8.2. Global Outsourcing as a Solution to the Developer Staffing Problem

Proposition H2: The high turn-over rate of the developers is a driver to outsource software development of a community source project to either domestic or international third-party vendors.

Proposition H3: The low quality of the in-house developers is a driver to outsource software development of a community source project to either domestic or international third-party vendors.

The developers designated by participating institutions might have private interests or incentives from the home institution that conflict with those of the community source project as a whole. These developers differ greatly in experience, skills, and business and technical knowledge. The turn-over rate and quality problems of developers are essentially the issue of accessibility of IS skills and capabilities in the community source project.

According to the motivations of global outsourcing identified in Figure 4, the needs to access sufficient and reliable IS skills and competencies and to ensure the performance of software development drive the consideration of global outsourcing. Senior executives in many organizations (including the community source organization in our case) believe that external IS vendors possess better technical and business expertise in providing IS development and provide the development more efficiently and effectively than internal IS departments. Globalization further makes it possible to access the best external vendors worldwide. DiRomualdo et al. (1998) described two company cases in the early era of global outsourcing. When Dow Chemical realized it was losing IS staff with critical technical skills, it created unique outsourcing joint ventures both domestically and internationally, to gain access to a broader talent pool worldwide. In a similar case, Philips Electronics restructured its IS organization and outsourced system delivery to both domestic and foreign joint ventures with professional software and system integration companies. Therefore, global outsourcing could be a viable solution for the problems of lack of IS skills and capacities and poor performance of in-house IS staff.

Proposition H4: The difficulty of managing the in-house developers is a driver to outsource software development of a community source project to either domestic or international third-party vendors.

The costs associated with communicating among and training of developers, identification of system requirements, and project management can be quite significant for internal software development. In the case of Kuali, for a small community source organization, managing such a virtual team in dispersed locations, with a large and ever-growing number, high turn-over rate, and varied skills of developers, was particularly difficult, intractable, and thus resource-consuming.

Global outsourcing makes it possible not to get involved in the complex IS staff management issues that the community source organization is unable to handle or not skilled in. Moreover, by outsourcing the software development, the community source organization can accommodate more partner institutions without worrying about managing and coordinating the developers "contributed" by them. Global outsourcing will fundamentally change the way how a community source system is developed since it eliminates the need of programmer contribution and simplifies in-house IS staff management.

It is worthwhile to point out that, a primary reason some organizations retain their software development in-house is trade secrets and/or critical key processes embedded in their software and systems that they would not wish to be made available to outsiders (King 2008). This is, however, not a big concern for a community source project to consider global outsourcing. When development is completed, the community source software is fully accessible to all development partners and becomes open source.

Community source partners are able to customize the software and integrate it into their organizations.

6.8.3. Sustainable Strategy

Proposition H5: Global outsourcing of software development will help community source management focus more on system sustainability.

Global outsourcing of software development can ease internal IS staff management and thus release the human, financial as well as other resources needed for the community source project. According to Davis et al. (2006), software outsourcing means that a much smaller proportion of the effort involves in-house IS staff, while a higher proportion lies in the software developers of the external vendor. The organization that outsources software development will not need large numbers of developers and programmers and its in-house IS team will shrink significant in size. The smaller function of the in-house team will involve project management, technology architects, and system strategies. Thanks to the easy and low-cost access to developers worldwide, the organization will thus save the investment in manpower, facilities and equipment for software development.

The community source organization in our study faced the challenges of sustaining long-term management and developing various community source systems to strength the strategic alliance among its participating institutions. During our interviews a number of executive members of the organization indicated that global outsourcing could help them shift the strategic focus of the project and achieve the goal of sustainability.

By global outsourcing of software development, the community source organization, under the financial and human resource pressures, is able to allocate and concentrate its resource, time and effort more efficiently on overall project management and performance, on system deployment, customization, integration and sustainability, and on organizational planning and strategies. In this way, in-house jobs will be “upgraded” to focus on core activities and priorities, and thereby able to help the organization gain sustainable advantages, survive, and achieve healthy growth.

6.9. Conclusions

6.9.1. Implications for Practice

Community source has just emerged recently as an innovative and economic way to develop enterprise information systems, not only in educational settings like the case in our study, but potentially in commercial settings as well. By pooling the resources of partner institutions, this new development model allows the sharing of costs, risks, and responsibilities and the sharing and control of the application/system which the partner institutions develop together. More importantly, most development partners of a community source project are themselves the users of the final product. Community source model offers the development partners an open development platform to share their common values, strategic goals, concepts, ideas, and knowledge with each other. The application/system developed out of this community source model could, therefore, be better shaped by the strategic requirements of the organizations and better tailored to different organizational environments and business needs than those generic commercial software/systems can.

Despite the advantages promised by community source, our study of the Kuali case shows that managing system developers in such a virtual environment and under limited resources is quite challenging, due to the facts of unbalanced development partners' contributions, decentralized locations, high turn-over rate, and poor quality and/or performance of the developers. The experience and issues of the Kuali project depicted in this paper provide other organizations insights into community source development.

The findings in our study further suggest that organizations making a similar decision to join or create a community source project and expecting a similar challenge of IS staff management consider global outsourcing as a practical alternative for internal software development. Global outsourcing will allow the project management of community source to deal with a single software contractor/vendor rather than a large number of individual developers from multiple partners, while still keep control of software design and quality of the final delivery which is guaranteed by international industry standards. Subsequently, global outsourcing can help community source management focus on its core activities—sustaining the system and strategic planning.

6.9.2. Research Contributions

To the best of our knowledge, our work is the first academic research focusing on the community source phenomenon and its interrelationship with global outsourcing. In recent years, both open source software development and outsourcing have gained momentum in IS research. This study takes the related research one step forward to incorporate the novel community source model in today's global economy. The rich information and in-depth understanding gained from the extensive survey of previous literature and a qualitative analysis of a real-world case of the Kuali community source project allow us to propose a research framework to study the drivers of global outsourcing in community source and expected values of global outsourcing for community source.

6.9.3. Future Research

In a future study, we plan to conduct a larger scale and more quantitatively-oriented survey, based on the research framework we propose here. We expect to validate the findings in the current qualitative study by a quantitative study and test the seven propositions in this paper rigorously. That is, our research has the potential to develop a general and solid theory for global outsourcing of community source development. Furthermore, we hope this study will inspire other researchers to use the research framework we suggest in this study to investigate further the interrelationships among community source, open source, and global outsourcing.

CHAPTER 7. CONCLUSIONS

7.1. Discussion of Community Source

As we can see, community source is getting more popular in higher education. There are three community source projects that also attract broad interest in higher education besides the Kuali project: the uPortal Project (www.uportal.org), the Sakai Project, and the Open Source Portfolio Initiative (OSPI). The uPortal project began in 2000 as a $3.2 million open source project of JA-SIC to create portal software for education. The Sakai Project (www.sakaiproject.org) became a $6.8 million open source project in January 2004 and promises to develop a next generation course management system (CMS). The Open Source Portfolio Initiative (OSPI) (www.theospi.org) was formed in January 2003 to address the software needed for student electronic portfolios (ePortfolios).

The greatest value for 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 the cross-language middleware code, Indiana University did not benefit; When Cambridge University developed a teaching tool, the U.S. and other institutions benefited 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 the organizations in the community source are mutually using other people's money to get and sustain the systems they need. Community source is foremost a coordination mechanism for institutional, corporate, and personal investments of resources, ideas, and talent.

Community source faces lots of uncertainty and might not become popular in the long term for certain reasons. For example, the commercial software might decide to lower the price due to the competition from the community source. As long as community source makes an impact to society, my research will have its unique value. My research contributes to the application software development literature and will have important an impact on 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 the open source.

7.2. Reflections and Practical Guidelines

This research generated several reflections and practical guidelines listed as below:

1. Developing and sustaining software in community source context is a good fit to the culture and values of higher education.

   The core values of higher education exist in discovery, knowledge sharing, and scholarly communities. In open-source software, communities draw on the leverage of a shared core system without constraining the option for local add-ons to meet specific needs.

2. The unbundling of software and support increase its efficiency.

First, the community source brings new efficiencies to the marked-based pricing of competitive support options. Second, the efficiency comes from the remarkable ability of the community to support itself.

3. Community source approach is not for all institutions.

   As we mentioned before, the success of community source project is affected by various factors: the size of the institution, internal technical capability, culture fit among partner institutions, system flexibility etc. When the decision makers in an institution decide if they are going to joining as a partner in community source or not, they have to conduct their investment analysis very carefully. Community source is not for everyone. The collaboration in community source will work effectively only if the partner institutions have a culture and philosophy fit.

4. CIO should fully notice the community source approach

   CIO should fully notice this new approach and become very active observers even if they decide not to make their institutions join community source at this time. The community source approach represents the new trend of enterprise application development. It is no doubt that the key concepts such as sharing resources and cost existing in community source will bring a unique value to system development.

5. Community Source Foundation should focus on system sustainability

   Once the system is developed, system sustainability becomes the main issue for Community Source Foundation to consider. Community Source Foundation will face the challenge of sustaining long-term management and developing various community source systems to strengthen the strategic alliance among its participating institutions. Senior managers in Community Source Foundation must know how to transfer the focus from project development to system sustainability. This is essence for maintaining community source successful in the long term.

7.3. Long Term Objectives

As the community-based open source becomes a new approach for enterprise application development, there is a clear need for a theoretical rigorous and practically relevant framework for understanding the value that is created and recreated through enterprise application development under community-based open source environment. In the long term, I would like to propose 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. My dissertation research for community source in the technology perspective, the economic perspective and the management perspective sets a strong foundation for theory building on software adoption. Eventually, a generic framework will be developed for describing the process of acquiring application software via either commercial, open source, or home grown approaches.

REFERENCES

Aspray, W., Mayadas, F., and Vardi, M. Y. (2006) "*Globalization and offshoring of software: A report of the ACM Job Migration Task Fource,*" Association for Computing Machinery.

Agerfalk, P.J., and Fitzgerald, B. (2008) "Outsourcing to an unknown workforce: Exploring opensourcing as a global sourcing strategy," *MIS Quarterly* (32:2), June, pp 385-409.

Ang, S., and Slaughter, S.A. (2001) "Work outcomes and job design for contract versus permanent information systems professionals on software development teams," *MIS Quarterly* (25:3), September, pp 321-350.

Avison, D. E.,and M. D. Myers (1995) "Information systems and anthropology: an anthropological perspective on IT and organizational culture," *Information Technology & People* (8:3), pp. 43-56.

Bagozzi, R.P. and U.M. Dholakia (2006) "Open source software user communities: a study of participation in Linux user groups". *Management Science* 52(7):1099–1115.

Baldwin C.Y., C. Dholakia, and l E. von Hippe (2006) "How user innovations become commercial products: a theoretical investigation and case study". *Research Policy* 35(9):1291–1313.

Benaroch, M. (2002) Managing Information Technology Investment Risks: A Real Options Perspectives, *Journal of Management Information Systems*, Fall, 19(2), pp.43-84.

Benaroch, M., Kauffman, R. (1999) A Case for Using Real Options Pricing Analysis to Evaluate Information Technology Project Investments, *Information Systems Research*, Mar 1999, 10, 1.

Beverakis, G., Dick, G.N., and Cecez-Kecmanovic, D. (2009) "Taking information systems business process outsourcing offshore: The conflict of competition and risk," *Journal of Global Information Management* (17:1), January-March, pp 32-48.

Black, F., Scholes M. (1973) The Pricing of Options and Corporate Liabilities, *The Journal of Political Economy*, Vol. 81, No.3., pp.637-654.

Byrd, T., A. Byrd and D. E. Turner, (2000) "Measuring the flexibility of information technology infrastructure: exploratory analysis of a construct", *Journal of Management Information Systems*/Summer, Vol. 17, No. 1. pp. 167-208.

Casati, F., Castano, S., Fugini, M., Mirbel, I., and Pernici, B. (2000) "Using Patterns to Design Rules in Workflows," *IEEE Transactions on Software Engineering* (26:8), pp 760 - 785.

Chau, P. Y.K., K. Y. Tam, (1997) "Factors affecting the adoption of open systems: an exploratory study", *MIS Quarterly*, March, pp. 1-24.

Cherbakov, L., G. Galambos, R. Harishankar, S. Kalyana, G. Rackham, (2005) "Impact of Service Orientation at the Business Level", *IBM Systems Journal*, Vol. 44, No. 4.

Crook, C. W. and R. L. Kumar (1998) "Electronic Data Interchange: A Multi-Industry Investigation Using Grounded Theory," *Information & Management* (34), pp. 219-232.

Dafoulas, G. and L. Macaulay (2001) "Investigating Cultural Differences in Virtual Software Teams," *The Electronic Journal on Information Systems in Developing Countries* (7:4), 2001, pp. 1-14.

Davis, G.B., Ein-Dor, P., King, W.R., and Torkzadeh, R. (2006) "IT offshoring: History, prospects and challenges," *Journal of the Association for Information Systems* (7:11), November, pp 770-795.

Deiters W., Goesmann T., Loffeler T., (2000) "Flexibility in workflow management: dimensions and solutions", *Computer Systems Science and Engineering*, Vol 15; Part 5, pages 303-314.

Delone W.H. and E.R. Mclean (1992) "Information systems success: the quest for the dependent variable". *Information Systems Research* 3(1), 60–95.

Delone W.H. and E.R. Mclean (2002) "Information Systems Success Revisited," in Proceedings of the 35th Hawaii International Conference on System Sciences, IEEE Computer Society Press, Los Alamitos, CA.

DePietro, R., E. Wiarde, and M. Fleischer (1990) "The Context of Change: Organization, Technology, and Environment in the Processes of Technological Innovation," Tornatzky, L. G., and Fleischer, M. (eds.), Lexington, MA, Lexington Book.

DeSanctis, G. and M.S. Poole (1994) "Capturing the complexity in advanced technology use: adaptive structuration theory," Organization Science, 5, No. 2, p. 121-145.

Dibbern, J., Goles, T., Hirschheim, R., and Jayatilaka, B. (2004) "Information systems outsourcing: A survey and analysis of the literature," *The DATA BASE for Advances in Information Systems* (35:4), Fall, pp 6-97.

DiBona, C., Ockman, S., and Stone, M. (1999) *Open Sources: Voices from the Open Source Revolution* O'Reilly & Aoosciates, Inc., Sebastopol, CA.

DiRomualdo, A., and Gurbaxani, V. (1998) "Strategic intent for IT outsourcing," *Sloan Management Review* (39:4), Summer, pp 67-80.

Ekstrom, M. A. and Bjornsson, H. C., (2005) "Valuing Flexibility in Architecture, Engineering and Construction Information Technology Investments," *Journal of Construction Engineering and Management*.

Erber, G., and Sayed-Ahmed, A. (2005) "Offshore outsourcing: A global shift in the present IT industry," *Intereconomics* (40:2), pp 100-112.

Feller, J., and Fitzgerald, B. (2000) "A Framework Analysis of the Open Source Software Development Paradigm," *Proceedings of the Twenty First International Conference on Information Systems.*

Finlay, P.N., and King, R.M. (1999) "IT sourcing: A research framework," *International Journal of Technology Management* (17:1-2), pp 109-128.

Fulk, J., J. Schmitz, and C. W. Steinfield (1990) "A Social Influence Model of Technology Use," in Organizations and Communication Technology, J. Fulk and C. W. Steinfield (eds.), Newbury Park, CA: Sage Publications, pp. 117-140.

Gordijn, J., Akkermans, H., Van Vliet, H. (2000) "Business modeling is not process modeling," *Proceedings of the Conference on Conceptual Modeling for E-Business and the Web,* pp 40-51.

Grover, V. (1993) "An Empirically Derived Model for the Adoption of Customer-Based Interorganizational Systems," *Decision Science* (24:3), pp. 603-640.

Gulla, J. A., T. Brasethvik, (2000) "On the Challenge of Business Modeling in Large-scale Reengineering Projects". *Fourth International Conference on Requirements Engineering*, 17 p.

Gupta, A., Seshasai, S., and Mukherji, S. (2007) "Offshoring: The transition from economic drivers toward strategic global partnership and 24-Hour knowledge factory," *Journal of Electronic Commerce in Organizations* (5).

Hann, I., J. Roberts, and S. Slaughter (2004) "Why developers participate in open source software projects: An empirical investigation," *International Conference on Information*

Systems, pp. 821-830.

Huhns, M. N. and M. P. Singh, (2005) “Service-oriented computing: key concepts and principles”, *IEEE Internet Computing*, January – February, pp. 75-81.

Hull, J.C. (1993) Options, Futures, and Other Derivative Securities (2nd ed.), *Prentice Hall*, Englewood Cliffs, NJ, 1.

Iacovou, A. L., I. Benbasat, and A. Dexter (1995) “Electronic data interchange and Small Organizations: Adoption and Impact of Technology”. *MIS Quarterly*:465-485.

Jiang, B., Frazier, G.V., and Prater, E.L. (2006) "Outsourcing effects on firms' operational performance: An empirical study," *International Journal of Operations & Production Management* (26:12), pp 1280-1300.

Kakabadse, A., and Kakabadse, N. (2005) "Outsourcing: Current and future trends," *Thunderbird International Business Review* (47:2), March-April, pp 183-204.

Karahanna, E., J. R. Evaristo, and M. Srite (2005) “Levels of culture and individual behavior: an integrative perspective,” *Journal of Global Information Management* (13:2), pp. 1-20.

Khan, N., and Fitzgerald, G. (2004) "Dimensions of offshore outsourcing business models," *Journal of Information Technology Cases and Applications* (6:3), pp 35-50.

Kimmle, M. (1972) *Design of Management Information Systems: An Approach and Case Study, PhD Dissertation*, Harvard University Press.

King, N. (1994) "The qualitative research interview," in: *Qualitative methods in organizational research: A practical guide,* C. Cassell and G. Symon (eds.), Sage, London, pp. 14-36.

King, W. and Cleland, D. I. (1975) The Design of Management Information Systems: An Information Analysis Approach, *Management Science*, Vol. 22, No. 3, 286-297.

King, W.R. (2008) "The post-offshoring IS organization," *Information Resources Management Journal* (21:1), January-March, pp 77-87.

Kogut, B. (2000) “The Network as Knowledge: Generative Rules and the Emergence of Structure,” *Strategic Management Journal* (21), pp. 405-425.

Kuan, K. and P. Chau (2001) “A perception-based model of EDI adoption in small businesses using technology–organization–environment framework”. *Information and Management* 38 , pp. 507–521.

Liu, M., and Zhao, L.J. (2007) "Real options analysis of the community source approach: Why should institutions pay for open source?," *The First China Summer Workshop on Information Management*.

Liu, M., D. D. Zeng, and J. L. Zhao (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, 2007, Keystone, CO, USA.

Liu, M., Wheeler, B., Zhao, J.L. (2008) "On Assessment of Project Success in Community Source Development," *Proceeding of International Conference on Information Systems.*

McFarlan, F.W., and Nolan, R.L. (1995) "How to manage an IT outsourcing alliance," *Sloan Management Review* (36:2), Winter, pp 9-23.

Mockus, A., R. Fielding, and J. Herbsleb (2002) "Two case studies of open source software development: Apache and Mozilla". *ACM Transactions on Software Engineering and Methodology* 11 3, pp. 1–3

Mukhopadhyay, T., S. Kekre, and S. Kalathur (1995) "Business value of information technology: a study of electronic data interchange," *MIS Quarterly* 19 2, pp. 137–156.

Myers, M.D. and F. Tan (2002) "Beyond models of national culture in information systems research," *Journal of Global Information Management* (10), pp. 24–32.

Nelson K., Nelson H., Ghods M. (1997) "Technology flexibility: Conceptualization, validation, and measurement", *Proceedings of the 30th Hawaii International Conference on System Sciences*, Volume 3, 10 p.

Ostrom, E. (1990) Governing the Commons: The Evolution of Institutions for Collective Action. New York; *Cambridge University Press*.

Papazoglou, M. P., D. Georgakopoulos, (2003) "Service-oriented computing: introduction", *Communications of the ACM*, 46 (10), October, 24 – 28.

Peltz, C. (2003) *Web Services Orchestration: A Review of Emerging Technologies, Tools, and Standards*, Hewlett Packard, Co.

Perry, R., Quirk K. (2007) "An Evaluation of Build Versus Buy for Portal Solutions" IDC White paper, sponsered by IBM, December.

Petrovic, O., Kittl, C., Teksten, R.D., (2001) “Developing Business Models for eBusiness”, *the Proceedings of the International Conference on Electronic Commerce 2001*, Vienna, Austria, October 31 - November 4.

Pfeffer, J. (1981) *Power in Organizations*, Pitman, Marshfield, MA.

Raymond, E. （1999） *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, revised ed. O’Reilly & Associates, Boston, MA.

Raymond, E. (2001) *The Cathedral and the Bazaar.* Sebastopol, CA: O.Reilly Media., Inc..

Robert, M. H. (2003) “Cooperative Game Theoretic Models for Decision-Making in Contexts of Library Cooperation,” *Library Trends*, Vol. 51 No.3, p441-61 winter.

Roberts, J., I. Hann, and S. Slaughter (2006) “Understanding the motivations, participation and performance of open source software developers: A longitudinal study of the Apache Projects,” *Management Science* 52(7) 984-999.

Schulte, W and Yefim V Natis (1996) *Service-Oriented Architecture*, Part 2, Gartner report SPA-401-069, 12 April.

Selznick, P. (1949) *TVA and the Grass Roots.* A Study of Politics and Organization. Berkeley, University of California Press.

Stewart, K. J., T. Ammeter, and L. Maruping (2005) “A preliminary analysis of the influences of licensing and organizational sponsorship on success in open source projects,” *Proceedings of the 38th Hawaii International Conference on System Sciences*.

Stohr, E.A., Zhao, J.L. (2001) "Workflow automation: overview and research issues," *Info. Systems Frontiers: Special Issue on Workflow Automation* (3:3), pp. 281-296.

Taudes, A., Feurstein, M., Mild, A. (2000) “Options Analysis of Software Platform Decisions: A Case Study,” *MIS Quarterly*, Vol. 24 No 2, pp. 227-243/June.

Thomaidis, N. S., Nikitakos, N. and Dounias, G. D. (2006) “The Evaluation of Information Technology Projects: A Fuzzy Multicriteria Decision-making Approach,” *International Journal of Information Technology & Decision Making*, Vol. 5, No. 1, 89-122.

Tillquist, J., J. L. King,and C. Woo (2002) “A representational scheme for analyzing information technology and organizational dependency,” *MIS Quarterly*, 26(2): 91.

Van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., and Barros, A.P. (2003) "Workflow patterns," *Distributed and Parallel Databases* (14:3), July, pp. 5-51.

Vixie, P. (1999) "Software engineering," in: *Open Sources: Voices from the Open Source Revolution,* C. Dibona, S. Ockman and M. Stone (eds.), O'Reilly & Associates, Inc., Sebastopol, CA, pp. 91-100.

Wang, E.T.G., Barron, T., and Seidmann, A. (1997) "Contracting structures for custom software development: The impacts of informational rents and uncertainty on internal development and outsourcing," *Management Science* (43:12), December, pp 1726-1744.

Wang, J., and Kumar, A. (2005) "A Framework for Document-Driven Workflow Systems," *International Conference Business Process Management*, Nancy, France, pp. 285-301.

Weber, M. （1947）*The Theory of Social and Economic Organization*. Translated by A. M. Henderson & Talcott Parsons, The Free Press.

Weerawarana, S. (2005) *Web services platform architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and more*, Prentice Hall PTR, Upper Saddle River, NJ, 2005.

Wheeler, B. (2007) "Open source 2010: Reflections on 2007," in: *EDUCAUSE Review*, pp. 49-52.

Wheeler, B. (2004) "The open source parade," in: *EDUCAUSE Review*, pp. 68-69.

Williamson, O. (1975) Markets and Hierarchies: Analysis and Antitrust Implications, New York, Free Press.

Yoshino, M.Y., and Rangan, U.S. (1995) *Strategic Alliances – An Entrepreneurial Approach to Globalization,* First Edition. Harvard Business School Press.

Zhao, J. L. (1998) "Intelligent Agents for Flexible Workflow Systems", *Proceedings of the AIS Americas Conference on Information Systems,* Baltimore, Maryland, August 14-16.

Zhao, J. L., H. K. Cheng, (2005) "Web Services and Process Management: a Union of Convenience or a New Area of Research?" *Decision Support Systems,* 40, pp.1-8.

Zhao, J. L., Tanniru M., Zhang L.-J., (2007) "Services computing as the foundation of enterprise agility: overview of recent advances and introduction to the special issue", *Information Systems Frontier,* Volume 9, Number 1 / March, Pages 1-8.

Zhu, K., K.L. Kraemer, and S. Xu (2002) "A Cross-country Study of Electronic Business Adoption Using the Technology-Organization-Environment Framework," *Proceedings of the Twenty-third International Conference on Information Systems*.