Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2006 Proceedings

Americas Conference on Information Systems (AMCIS)

December 2006

Benevolent Competence and Integrity-Based Trust in Knowledge Transfer: A Look at Software Reuse

John Drake
Auburn University

Nikhil Mehta
Auburn University

Follow this and additional works at: http://aisel.aisnet.org/amcis2006

Benevolent, Competence, and Integrity-Based Trust in Knowledge Transfer: A Look at Software Reuse

John Drake
Auburn University
drakejr@auburn.edu

Nikhil Mehta
Auburn University
mehtan1@auburn.edu

ABSTRACT

The potential of software reuse to reduce costs and time while increasing quality in a new project development, make it an important issue in the management of information systems. The question of how to increase software reuse in new application development has often been asked, but mostly from a managerial perspective rather than the individual perspective. This research effort attempts to address this limitation by conceptualizing software as a form of knowledge, and intending to examine if trustworthiness of various knowledge transfer mechanisms influences receipt of software through those mechanisms. And if so, which knowledge transfer mechanisms do software developers trust most? We propose and intend to test a model of software reuse in which the links between knowledge transfer mechanisms and receipt of software through those mechanisms is mediated by competence-, integrity-, and benevolence-based trust. Hypotheses are developed to test those links, and appropriate methodology is outlined to conduct the research. With a better understanding of why software developers trust certain mechanisms over others, it will be possible to recommend better administrative policies and procedures that work in conjunction with trust to encourage software reuse.

KEYWORDS

Knowledge management, trust, software reuse.

INTRODUCTION

As software development costs have been rising over the years, managers have been searching for means to improve software development efficiency. One of the ways includes reusing software. Given the nature of software development, reusability seems a viable option. As Nazareth and Rothenburger (2004) explain, "75% of program functions are common to multiple programs, 40-60% of all code is reusable from one application to another, 60% of the design of all business applications is reusable, and 15% of program code is unique to a specific application" (p: 1). Potential benefits of reusability are significant. They include reduced software development time and cost, and improved software maintainability, quality, and portability (Apte et al. 1990).

Despite these benefits, reusability does not seem to have developed as a robust practice in organizations (Nazareth & Rothenburger, 2004). Most of the research searching for reasons focuses primarily on technical challenges facing managers trying to develop a reusability strategy (Ali et al. 2004; Owen et al. 2004; Ravichandran 1999; Rothenberger et al. 2003), rather than on relational issues, a point that concerns our study. We identify with the emerging thought that although technical issues have a large impact on the success of software reusability programs, relational issues significantly influence individual willingness to reuse a software component or not (Levin & Cross 2004). In this paper, we focus our attention on one such relational issue – trust.

Trust is defined as the willingness of an individual to be vulnerable to actions of others (Mayer, Davis, & Schoorman 1995). Following Levine & Cross's (2004) conceptualization, we focus on the closely related concept of perceived trustworthiness, which is defined as "that quality of the trusted party that makes the trustor willing to be vulnerable (p. 1478)." The trustworthiness of a knowledge source has been found to be a factor influencing the receipt of useful knowledge in general (Levin et al. 2004). Since software is a type of knowledge object, we suspect that trustworthiness of a source will also be a significant factor in the receipt of useful software. This in turn may help explain why individuals reuse that software in new application development.

Since the source of reusable software components can come from a variety of knowledge transfer mechanisms (Alavi et al. 2001), it would be useful to understand how trustworthiness of different mechanisms affects the receipt of useful software components through those mechanisms, and thus software reuse. The purpose of this study is twofold: to identify whether

increases in the trustworthiness of a source corresponds with increasing usage of software reuse from that source; and secondly, to determine which mechanisms do software developers trust most. We propose and intend to test a model of software reuse in which the links between knowledge transfer mechanisms and receipt of software through those mechanisms are mediated by three factors of trustworthiness – competence, benevolence, and integrity (Mayer et al., 1995). In the next section, hypotheses are developed to test those links, and in the following section, appropriate methodology is outlined to conduct the research.

LITERATURE REVIEW

Organizational knowledge management process involves a framework of four steps: (1) creation, (2) storage/retrieval, (3) transfer, and (4) application (Alavi et al. 2001). Implicit in this framework is the view of knowledge as an 'object' that can be stored and transferred within an organization. Knowledge objects are created by individuals, stored in the organizational, and transferred to other individuals who apply it in new contexts. Software reuse assumes that knowledge can be imbedded as an object, specifically as a software object, component, function, or free-standing code. The same framework for knowledge reuse holds true for software reuse.

The best transfer mechanisms (sometimes referred to as channels or communication modalities) for knowledge transfer is context dependent (Alavi et al. 2001). Communication research has identified a four-way classification of transfer mechanisms: formal/informal and personal/impersonal (Mohr et al. 1990). This four-way classification has been extended to the mechanisms in knowledge transfer (Holtham et al. 1998; Willem et al. 2002).

	Personal	**Impersonal**
Formal	• Teams • Mentorship • Personnel transfer	• Knowledge management systems • Training sessions
Informal	• Water cooler conversations • Unscheduled meetings • Friendships	• Open source systems • Internet message boards

Table 1. Knowledge Transfer Mechanisms

Impersonal transfer mechanisms supported by information technologies typically store and distribute explicit knowledge wider as compared to personal transfer mechanisms (Alavi et al. 2001). Software objects are written in an explicit language. Because of the explicit nature of software objects, impersonal transfer mechanisms supported by information technologies are more likely to contain the objects that developers need. It has been observed that libraries of software reuse objects are a necessary component of software reuse initiatives (Apte et al. 1990). The explicit nature of software objects allows them to be shared widely through impersonal channels. Hence, software developers are more likely to use impersonal transfer mechanisms to receive software for reuse purposes.

> *H1: Impersonal knowledge transfer mechanisms will be used more often in software reuse than personal knowledge transfer mechanisms.*

Mayer et. al. (1995) have proposed three dimensions of trustworthiness – benevolence, competence, and integrity. The benevolence dimension is based on the commonality and goodwill of the other person. It is emotional and general in nature, where the expectation is that the other person will want to do "good things" and only has the best intentions toward the recipient. The second dimension is based on competence. It is cognitive and specific in nature, where the recipient judges the expertise of the other person as a sign of trustworthiness, at least trustworthy in that particular area of expertise. The third dimension is based on integrity. Integrity-based trust refers to the whether or not the trustee adheres to a set of acceptable principles.

Trust has a significant impact on knowledge transfer (Lucas 2005). Competence-based trust as well as benevolence-based trust significantly impact the receipt of useful knowledge (Levin et al. 2004). If this were to remain true in software development, competence-based trust and benevolence-based trust should be positively correlated with the receipt of useful software objects. By extension, the more trust, either competence or benevolence, a software developer has in a knowledge source, the higher the percentage of application will come from reused software objects.

Integrity-based trust in software transfer consists of the trustor recognizing that the trustee adheres to a set of principles that the trustor finds acceptable (Mayer et al. 1995). As trust in general significantly impacts knowledge transfer, integrity-based trust will also impact knowledge transfer. In software reuse, integrity-based trust means the trustor believes that the trustee adhered to certain standards and prescribed principles of software development while writing the reusable software. If the trustor does not believe the transfer mechanism contains software built with integrity, they are unlikely to use it.

> *H2a: Increased competence-based trust in knowledge transfer mechanisms will increase the percentage of software reuse.*

> *H2b: Increased benevolence-based trust in knowledge transfer mechanisms will increase the percentage of software reuse.*

> *H2c: Increased integrity-based trust in knowledge transfer mechanism will increase the percentage of software reuse.*

Informal organizations are generally built through organic interactions. Things such as proximity and shared background encourage personal, informal relationships to develop and help organizations run. Benevolence-based trust is a corner stone of these successful relationships (Morton et al. 2004). Benevolent-based trust is also prevalent in less personal yet still informal environments, such as public message boards, (Nelson et al. 2005). The information from these sources may not be any more credible, but because of the benevolent based trust, individuals accept what is said as true. It is thus suspected that software developers will also have higher benevolence-based trust in informal sources. In particular, they will have higher benevolent based trust in informal reuse sources than in formal reuse sources.

> *H3: Software developers will have higher benevolence-based trust in informal mechanisms than formal mechanisms for software reuse.*
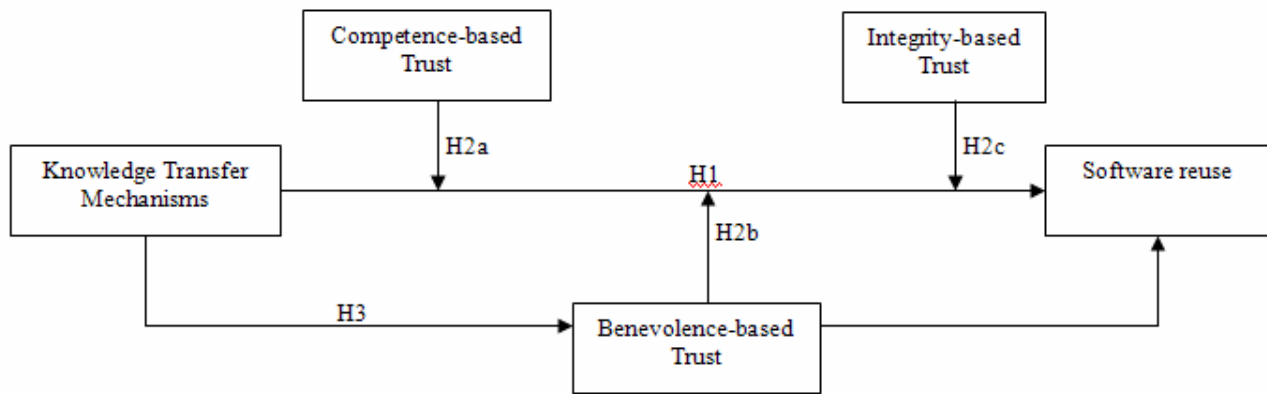

Figure 2. Research Model

REARCH METHODOLOGY

Data Collection

Data will be collected with a self-response, closed-ended survey conducted through mailed questionnaires. The population of interest is software developers in large and medium IT departments. Although self-response surveys do have their limitations, the level of trust can only be measured by directly asking the participants. While trust is usually something that is built up over time, we will only look at a snapshot in time. This is because we are more interested in the current preferences than how they got to those preferences.

There are two phases to this study; a pilot study and the full study. One hundred participants in the pilot study will verify the validity and reliability of scales. Hinkin (1998) suggests using exploratory factor analysis followed by the use of confirmatory factor analysis to assess the goodness of fit of the model and to examine the fit of individual items within that model. A sample size of 700 developers is desired for the full study. We will select a random sample from the members of various computer programmer user groups, and expect a response rate of 25 to 30 percent. Structural equation modeling will be used to test our hypotheses.

Survey Instrument

We adapted the survey instrument from standard measures found in literature. Software reuse will be a self-reported estimate for how much code written during a project was from another source (object, component, or code). For benevolent, integrity, and competence-based trust, we will adapt various measures from Johnson (1996), Mayer and Davis (1999), and McAllister (1995). To assess the use of knowledge transfer mechanisms, we will use 4 archetypes, one for each formal-personal, formal-impersonal, informal-personal, and informal-impersonal. The purpose is to keep the research grounded and to avoid a long questionnaire. Although this may limit the generalizability of our findings between the four types of channels, it sets the groundwork for further study.

EXPECTED IMPLICATIONS FOR THEORY AND PRACTICE

This study is expected to make a theoretical contribution to the knowledge management and organizational learning literatures. Not many studies have investigated a multidimensional conception of trust (competence, benevolence, and integrity) specifically as a moderator between knowledge transfer mechanisms and software reuse. The results of this study are expected to help us understand how trust moderates the influence of knowledge transfer mechanisms on software reuse. In doing so, our study will provide a better understanding of how relational issues make the social milieu of an organization more or less effective for knowledge transfer.

With an increased interest among practitioners in the role of trust in organizational settings, our study may convince practitioners, especially in software firms, to invest in ways to improve organizational trust as an effective means to improve software reuse in their organizations. Results may help executives identify and invest in appropriate knowledge transfer mechanisms to improve software reuse.

REFERENCES

1. Alavi, M. and Leidner, D. E. (2001) Knowledge management and knowledge management systems: Conceptual foundations and research issues, *MIS Quartlery*, 25, 1, 107 - 136.

2. Ali, F. M. and Du, W. (2004) Toward Reuse of object-oriented software design models, *Information and Software Technology,* 46, 8, 499-517.

3. Apte, U., Sankar, C. S., Thakur, M., and Turner, J. E. (1990) Reusability-based strategy for development of information systems: Implementation experience of a bank, *MIS Quarterly*, 14, 4, 421 - 433.

4. Hinkin, T. R. A (1998) Brief tutorial on the development of measures for use in survey questionnaires, *Organizational Research Methods*, 1, 1, 104 - 121.

5. Holtham, C. and Courtney, N. (1998) The executive learning ladder: A knowledge creation process grounded in the information systems domain, Fourth Americas Conference on Information Systems, Baltimore, MD, 594 -597.

6. Johnson, J. L., Cullen, J. B., Sakano, T., and Takenouchi, H. (19960 Setting the stage for trust and strategic integration in Japanese-U. S. cooperative alliances, *Journal of International Business Studies*, 27, 5, 981 - 1004.

7. Levin, D. Z., and Cross, R. (2004) The strength of weak ties you can trust: The mediating role of trust in effective knowledge transfer, *Management Science*, 50, 11, 1477 - 1490.

8. Lucas, L.M. (2005) The impact of trust and reputation on the transfer of best practices, *Journal of Knowledge Management*, 9, 4, 87 - 101.

9. Mayer, R. C. and Davis, J. H. (1999) The effect of the performance appraisal system on trust for management: A field quasi-experiment, *Journal of Applied Psychology*, 84, 1, 123 - 136.

10. Mayer, R. C., Davis, J. H., and Schoorman, F. D. (1995) An integrative model of organizational trust, *Academy of Management Review*, 20, 709 - 734.

11. McAllister, D. J. (1995) Affect- and cognition-based trust as foundations for interpersonal cooperation in organizations, *Academy of Management Journal*, 38, 1, 24 - 59.

12. Mohr, J. and Nevin, J. R. (1990) Communication strategies in marketing channels: A theoretical perspective, *Journal of Marketing*, 54, 4, 36 - 51.

13. Morton, S. C., Brooks, N. J., Smart, P. K., Backhouse, C. J., and Burns, N. D. (2004) Managing the informal organization: Conceptual model, *International Journal of Productivity and Performance Management*, 53, 3/4, 214 - 232.

14. Nazareth, D. L. and Rothenberger, M. A. (2004) Assessing the cost effectiveness of software reuse: A model for planned reuse, *Journal of Systems and Software*, 73, 245-255.

15. Nelson, M. R. and Otnes, C. C. (2005) Exploring cross-cultural ambivalence: a netnography of intercultural wedding message boards, *Journal of Business Research*, 58, 89 - 95.

16. Owen, J., Burstein, F., and Mitchell, S. (2004) Knowledge reuse and transfer in a project management environment, *Journal of Information Technology Cases and Applications*, 6, 4, 21 - 35.

17. Ravichandran, T. Software reusability as synchronous innovation: A test of four theoretical models, *European Journal of Information Systems,* 8, 183 - 199.

18. Rothenberger, M. A., Dooley, K. J., Kulkarni, U. R., and Nada, N. (2003) Strategies for software reuse: A principal component analysis of reuse, *IEEE Transactions on Software Engineering*, 29, 9, 825 - 837.

19. Willem, A. and Scarbrough, H. (2002) Structural effects on inter-unit knowledge sharing: The role of coordination under different knowledge sharing needs, Third European Conference on Organizational Knowledge, Learning, and Capabilities, Athens.