Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2003 Proceedings

Americas Conference on Information Systems (AMCIS)

December 2003

Reuse Consumption: A Behavioral Model of Individual Reuse of Pre-Built Software Components

J. Harold Pardue
University of South Alabama

Jeffrey Landry
University of South Alabama

Loveleen Sharma
University of South Alabama

Follow this and additional works at: http://aisel.aisnet.org/amcis2003

REUSE CONSUMPTION: A BEHAVIORAL MODEL OF INDIVIDUAL REUSE OF PRE-BUILT SOFTWARE COMPONENTS

J. Harold Pardue
University of South Alabama
hpardue@jaguar1.usouthal.edu

Jeffrey P. Landry
University of South Alabama
jlandry@usouthal.edu

Loveleen Sharma
University of South Alabama
loveleensharma@hotmail.com

Abstract

As organizations implement software reuse programs to improve productivity and software quality, they face the challenge of how best to encourage reuse. This study describes a behavioral model that attempts to explain the influence of a reuse consumer's perception of the trustworthiness of the producer, the technical characteristics of the component, and the organizational reuse norms on a reuse consumer's intention to reuse a component.

Keywords: Software reuse, TAM, TAM2, trust

Introduction

Reuse of pre-built software components has the potential to improve software quality, process predictability (Nidumolu and Knotts 1998) and programmer productivity. The successful reuse of pre-built software components can reduce the time required to design and implement new software and systems while reducing error rates by relying on the quality assurance activities performed by component producers.

A premise of this paper is that successful intra-organizational reuse requires the consideration of technical and organizational issues, especially social and process issues. Social and process issues related to software development remain at the forefront of issues that affect software quality and productivity. Advances in behavioral and process issues have traditionally lagged behind advances in hardware and software tools. Software development is continuing to change and evolve as organizations are adopting various approaches, including the so-called "agile" practices, such as extreme programming (Beck 1999) and Scrum (Schwaber and Beedle 2002). Given the high costs of developing software, advances in the social and behavioral side of software development, if successful and widely adopted, might provide significant impact on, or even a breakthrough in, IT productivity. By contributing to the understanding of social and process issues as they relate to software reuse, this study aspires to contribute to potential advances in software process improvement and productivity.

Our perspective can be compared and contrasted with popular themes in the study of reuse. The proposed research model incorporates the producer-consumer perspective of Joos (1994), but unlike Joos, we focus on the consumer-side only, and rewards, a key motivational tactic in Motorola's reuse program, play a minor role in our view. The impediments to reuse identified in the software engineering literature (Griss 1993), such as quality concerns and the not-invented-here syndrome, are indirectly address in our model through consideration of trust, risk, and organizational reuse norms. Much attention in software engineering literature is focused on technical issues, such as the use of object-oriented languages for fostering reuse; on the contrary, our study places more importance on social-psychological and organizational reuse issues.

The purpose of this research is to propose a behavioral model that explains individual peer-to-peer reuse consumption in an organizational context. A richer understanding of behavioral factors that influence individual reuse could inform management practices and policies related to encouraging and fostering effective component reuse. Therefore, the research question is: *What is the role of perceived trustworthiness of the component producer and reuse norms in explaining individual, intra-organizational peer-to-peer reuse in an organizational innovation context?*

Behavioral Model of Reuse

Viewed as an innovation adoption decision, individual reuse of software components can be understood in terms of the now familiar diffusion of innovations theory (Rogers 1995, Leonard-Barton 1987). In this theory, the adoption decision is influenced by the adopter's perception of the innovation in terms of five characteristics: relative advantage, compatibility, complexity, trialability, and observability.

Software developers, as potential adopters, perceive that the reusable component, an innovation, has varying degrees of the qualities that make it attractive for adoption. For example, a developer might perceive that using a pre-built component provides a relative advantage over developing new software from scratch. From a cost-benefit perspective, the savings in time from not having to design, code, and test a new component from scratch may be substantial enough to justify learning to use and incorporate a pre-build component on a new project. On the other hand, the decision to reuse rather than rebuild may be incompatible with the norms of a software organization that values the rigor and creativity of writing your own code.

Technology Acceptance Model

A variety of theoretical perspectives have been developed to examine the factors that influence technology usage. One stream of research has focused on intention-based models (Ajzen 1991; Ajzen and Fishbein 1980). The Technology Acceptance Model (TAM) is one such model (Davis 1989, Davis et al. 1989, Taylor and Todd 1995). Based on the diffusion of innovations and social learning theories, TAM (Figure 1) explains how beliefs about a technology innovation affect individual decisions to adopt and use.
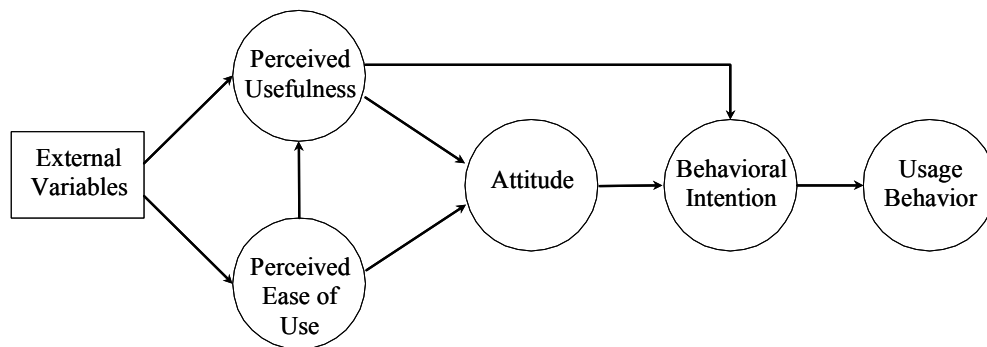

**Figure 1. Technology Acceptance Model (Davis 1989)**

Viewing a reusable component as an innovation, we can employ TAM to explain individual reuse consumption from an intra-organizational, voluntary adoption perspective. Research has found that the key to getting individuals to accept a technological innovation is getting the individuals to perceive the technological innovation as useful and easy to use. Usefulness and ease-of-use are based on Rogers' generic characteristics.

It is hypothesized therefore that individual reuse consumption behavior will evolve along a beliefs-attitudes-intentions pathway. An individual will develop beliefs about a reusable component (that he did not build), and these beliefs will influence that developer's attitude towards reusing that component. We propose that an individual developer's intention to adopt a component developed by a peer is influenced by three factors: the perceived usefulness of the component, the developer's attitude toward reusing the component, and the organization norms concerning reuse. Furthermore, the component's perceived ease of use is hypothesized to influence both perceived usefulness and attitude towards reusing the component.

Organizational Dyadic Trust

In any organizational situation where one party is "vulnerable to the actions of another party based on the expectation that the other will perform a particular action important to the trustor, irrespective of the ability to monitor or control that other party" (Mayer et al. 1995, p.712), trust is an important ingredient for successful interaction. Vulnerability involves risk; the more a trustor trusts a trustee, the more risks the trustor is willing to take in the relationship with the trustee, depending on the perceived risk of the behavior. Recent research on organizational trust in dyads suggests that the trust one individual has for another is influenced by the perceptions of the trustee's ability, benevolence, and integrity, and the trustor's propensity to trust, in general.

We submit that the reuse of a software component produced by another developer within the same organization constitutes an act of trust as defined by Mayer et al. (1995). In other words, when a developer reuses a component, they are exhibiting a willingness to be vulnerable to the actions of the component producer on the expectation that the producer will provide a component that conforms to best practices and quality standards, irrespective of the ability to monitor or control the component producer. The complete research model for this study is depicted in Figure 2 below.
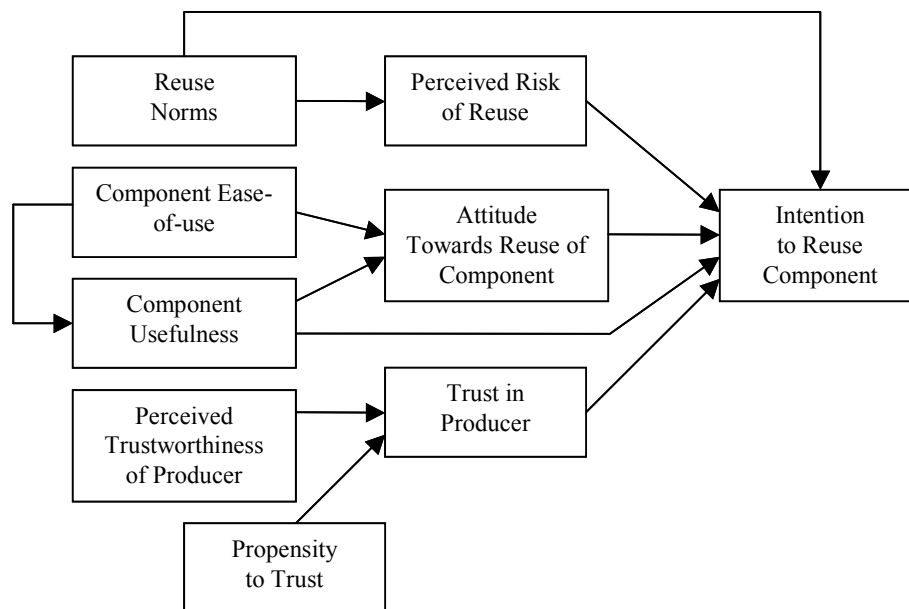
Figure 2. Research Model

We propose to apply Davis' findings about beliefs affecting attitudes towards technology use for explaining individual software reuse. Beliefs about reusable components will influence attitudes toward their use. More specific variables may be found to be related to software components, but we hold that Davis' usefulness and ease-of-use constructs are theoretically adequate for explaining beliefs about software components. We hypothesize that:

   H1: *A software developer's perception of the usefulness of a software component will be positively associated with the attitude toward reusing that component.*

   H2: *A software developer's perception of the ease-of-use of a software component will be positively associated with the attitude toward reusing that component.*

Furthermore, Davis found, and other researches confirmed, that ease-of-use was associated with usefulness. The more an individual believed that a technology was easy to use, the more likely he would also believe that the technology was useful. Therefore,

   H3: *A software developer's perception of the ease-of-use of a software component will be positively associated with the perceived usefulness of that component.*

Not only was usefulness beliefs found to impact attitudes towards a technology's use, but usefulness beliefs were also found to be directly related to intentions to use. Therefore,

H4: *A software developer's perception of the usefulness of a software component will be positively associated with the developer's intention to reuse that component.*

Following the predictions suggested by the social psychology literature, specifically, the theory of reasoned action and theory of planned behavior (Ajzen 1991, Ajzen and Fishbein 1980), Davis found that attitudes influenced intentions. Specifically, Davis hypothesized and found support for the notion that attitudes towards technology use would precede intentions to use technology. We support this assertion with the following hypothesis:

H5: *A software developer's attitude toward the use of a software component will be positively associated with the developer's intention to use that component.*

Social psychology theory suggests that as individuals develop attitudes toward a specific behavior, they will also perceive the opinions of others about their performing the specific behavior. Individuals will perceive and be influenced by the opinions of those whose opinions are important to them. In the workplace, these opinions may be of their superiors, subordinates, or peers. Attitudinal theory predicts that subjective norms influence intentions to behave. Although Davis' initial studies did not find normative influences to be predictive of adoption intentions, subsequent studies of TAM2 (Venkatesh and Davis 2000) did find subjective norms to be significantly related to behavioral intentions in mandatory usage contexts.

When considering whether to reuse a pre-built component, a software developer may be influenced by the opinions of their superiors and peers. If a developer's manager encourages or rewards reuse behaviors, the developer's intentions will be influenced in a positive direction. For example, if a manager perceives that reuse consumption will result in reducing project costs and schedules, and values these benefits of reuse, developer intentions to engage in reuse behavior will be encouraged. On the other hand, if reuse is not rewarded, or if an organization's prevailing beliefs are anti-reuse, for example, if reuse is considered to be a weak or lazy tactic rather than rigorous or creative, the developer's intentions to reuse would be dampened. We are conceiving of subjective norms in the reuse context as "reuse norms." We hypothesize,

H6: *Organizational norms that are supportive (unsupportive) of reuse consumption will be positively (negatively) associated with a developer's intention to reuse a software component.*

We propose that organizational reuse norms directly influence the developer's perception of the risk of reuse. The risk of reuse is defined as the developer's perception of the probability of a negative outcome of reuse, such as unexpected defects or project delays. Besides component-specific and producer-specific factors, the organization and the reuse context which it provides will influence a developer's perception of the riskiness of reuse behaviors. In an environment where reuse is not supported or even discouraged, a developer may view the practice of reuse as relatively risky compared to an environment where reuse is strongly encouraged and supported. We hypothesize,

H7: *Organizational norms that are supportive (unsupportive) of reuse consumption will be inversely (positively) associated with a developer's perception of the risk of reuse.*

The effect of reuse norms on a developer's intention to reuse a specific component will be partially mediated by the developer's perception of the risk of reuse. A developer might have a positive attitude towards reusing a specific component, but nonetheless does not intend to reuse the component because the developer perceives the probability of a negative outcome (e.g., a critical, hard-to-fix fault) to be too. The risks of reuse, which may result from reuse norms, or factors beyond the scope of this study, make developers vulnerable to potential negative outcomes that would discourage reuse. We hypothesize,

H8: *A software developer's perception of the risk of reuse risk will be inversely associated with a developer's intention to reuse a software component.*

Mayer's trust model predicts that trust, or a willingness of one party to be vulnerable to the actions of another, derives from a trustor's perception of the trustee's ability, integrity, and benevolence, and from an individual's propensity to trust people in general. In the reuse context, a potential reuse consumer might develop trust in the reuse producer based on his perceptions that the producer was a competent programmer. The consumer might also develop trust based on the producer's adherence to a set of programming practices that the consumer found acceptable, such as following the coding, documenting and testing policies

defined by the organization. Or, the consumer might develop trust for the producer based on the producer's general helpfulness with problems and questions that arise on projects or specifically with reuse help and support.

> H9: *Perceptions of a component producer's ability, integrity, and benevolence in the software development and reuse context will be positively associated with the developer's trust in the component producer.*

> H10:   *A potential reuse consumer's propensity to trust will be positively associated with the developer's trust in the component producer.*

According to Mayer's model, the manifestation of trust is risk-taking in relationship. In a software reuse context, risk-taking in relationship is conceived of as the intention to reuse a software component and the actual use of that component. Reuse is about relying on someone else's work, trusting that the producer created a quality component. Reuse consumption involves risk to the developer. How will that component perform? Will it work? How does it work? How well was it tested? Will it fail? If it fails, will I be able to find the fault and fix it? As the trust model suggests, the trust in a producer will influence the likelihood that a consumer will decide to risk using that producer's component. We hypothesize,

> H11:   *A software developer's trust in the component producer will be positively associated with the software developer's intention to reuse that component.*

Methodology

A pilot study, designed as a controlled experiment, is proposed to test the research model. Undergraduate CIS students enrolled in a CS1 and CS2 programming sequence will be enlisted as participants. Reuse norms will be defined in terms of the classroom/programming assignment context. The experimental manipulation will be affected by constructing scenarios regarding the producer of the component (a student from a previous semester) and the instructor's position on reuse.

The use of student subjects may be considered problematic in this study, due to the need for an organizational context and dyadic trust that does not exist in the classroom in the same way that it does in industry. We believe, however, that scenarios can be constructed that provide a theoretically valid operationalization of the research constructs while communicating a meaningfully realistic context for student subjects. While the classroom reuse environment may not be equivalent to industrial environments, it is nevertheless analogous enough to at least provide a useful pilot study, even if generalizability remains a concern. If a good start is obtained from the pilot study, the design may be refitted for a sample of experienced developers in realistic organizational situations.

Implications

Our theoretical framework will provide a richer understanding of the interpersonal and organizational dynamics of individual, peer-to-peer reuse. Advances in the understanding of social and process issues in an organizational software development context can lead to similar advances in software productivity and quality. At a minimum, practical implications include suggested practices for managers and developers for fostering reuse in a software organization. If trust plays a significant role in explaining reuse, then managers of software organizations should consider creating work environments that encourage and enable component producers to build relationships with potential adopters that promote trustworthiness.

References

Ajzen, I., "The Theory of Planned Behavior," Organizational Behavior and Human Decision Processes, vol. 50, 1991, pp. 179-211.

Ajzen, I. and Fishbein, M., Understanding Attitudes and Predicting Social Behavior, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1980.

Beck, K. "Embracing Change with Extreme Programming," IEEE Computer, October 1999, pp. 70-77.

Davis, F. D. "Perceived Usefulness, Perceived Ease of Use, and End User Acceptance of Information Technology," MIS Quarterly (13), September 1989, pp. 318-339.

Davis, F. D., Bagozzi, R. P. and Warshaw, P. R., "User Acceptance of Computer Technology: A Comparison of Two Theoretical Models," Management Science, vol. 35, no. 8, 1989, pp. 982-1003.

Griss, M. L. "Software Reuse: From Library to Factory," IBM Systems Journal (32:4), 1993, pp. 548-566.

Joos, R. "Software Reuse at Motorola," IEEE Software (11:5), September 1994, pp. 42-47.

Leonard-Barton, D., "Implementing Structured Software Methodologies: A Case of Innovation in Process Technology," Interfaces, vol. 17, no. 3, 1987, pp. 6-17.

Mayer, R., Davis, J, and Schoorman, F. "An Integrative Model of Organizational Trust", Academy of Management Review, 20:3, 1995, pp. 709-734.

Nidumolu, S. R. and Knotts, G. W. "The Effects of Customizability and Reusability on Perceived Process and Competitive Performance of Software Firms," MIS Quarterly, June 1998, pp. 105-137.

Rogers, E. M., Diffusion of Innovations, The Free Press, New York, 1995.

Schwaber, K. and Beele, M. Agile Software Development with Scrum, Prentice Hall, New Jersey, 2002.

Taylor, S. and Todd, P. A. "Understanding Information Technology Usage: A Test of Competing Models," Information Systems Research (6:2), June 1995, pp. 144-176.

Venkatesh, V., Davis, Fred D.. "Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies," Management Science (46:2), February 2000, pp. 186-204.