1-1-2009

Using Trust-TAM to Explain Software Component Adoption

J. Harold Pardue
University of South Alabama, hpardue@usothal.edu

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

Follow this and additional works at: http://aisel.aisnet.org/amcis2009

Recommended Citation
http://aisel.aisnet.org/amcis2009/177
Using Trust-TAM to Explain Software Component Adoption

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

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

ABSTRACT

This study uses Trust-TAM to describe and empirically test a behavioral model of software component adoption. Three sets of predictive factors were tested and found to be statistically significant in their relationship with intention to adopt: the consumer’s perception of the trustworthiness of the component producer, the adoption-related characteristics of the component, and the organizational reuse norms. The results further validate the use of Trust-TAM as a predictor of technology adoption, adding to the explanatory power of the traditional TAM to include interpersonal and normative elements. Practical implications include the importance of creating a professional environment that fosters trust and norms that support technological innovation.

Keywords

IT adoption, software component reuse, TAM, TAM2, trust, trust-TAM.

INTRODUCTION

Adoption of pre-built software components has the potential to improve software quality, process predictability (Nidumolu and Knotts 1998) and software developer 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 component adoption, 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 addressed 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 component adoption in an organizational context. By peer-to-peer, it is meant the adoption of software components written by an individual’s colleagues. 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, TAM2) is one such model (Davis 1989, Davis et al. 1989, Taylor and Todd 1995b, Venkatesh and Davis 2000). Based on the diffusion of innovations and social learning theories, TAM and TAM2 (Figure 1) explains how beliefs about a technology innovation affect individual decisions to adopt and use.

![Figure 1 – TAM (Davis 1989) with subjective norm from TAM2 (Venkatesh and Davis 2000)](image)

Davis (1989) found empirical support for hypotheses that technology usage follows a beliefs-attitudes-intentions pathway. 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. 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.

Using TAM requires defining a technology and an adopter of that technology. We now consider the choice of each as we defend the application of TAM. Typically, the technologies studied have been information technologies, such as email, word processors, spreadsheets, voice mail, graphics packages, microcomputers Web sites, and newsgroups. See Wang and Benbaset (2005) for a non-exhaustive list. Venkatesh et al (2003) studied “sophisticated organizational technologies,” such as a portfolio analyzer. Taylor and Todd (1995a) studied the case of adopting the services of a computer resource center. A CRC is a bit different in that it provides a range of information services, including access to computers and a collection of
information technologies, and assistance by staff personnel. Arguably, it is also a sophisticated form of organizational technology as well.

In the current study the technology in question is a reusable software component. It can be considered as a sophisticated organizational technology in that it possesses attributes of both process and product (Damanpour 1991) types of innovations. Adopting a reusable software component requires acceptance of both a software process—intra-organizational reuse in general—and a software product—the particular reusable component produced by another developer in that organization.

In most TAM studies, the target adopter is an IT end-user. In our case, the target adopter is a software developer, who is adopting a component for use in developing an information system to be used by end-users. We are not concerned with end-use acceptance of an information system, but in the developer’s acceptance of reusable software component in the process of building the system. In this sense, our choice of technology adopter is a bit different from prior studies, but the correct one in the case of the technology in question. Our study tests TAM in a slightly different way.

It is hypothesized therefore that individual component adoption behavior evolves along a beliefs-attitudes-intentions pathway. An individual develops beliefs about a reusable component (that he did not build), and these beliefs 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 adoption 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.

TRUST-TAM

TAM is a widely researched and well-accepted model explaining individual IT adoption. Perceived usefulness and ease of use are powerful and parsimonious predictors of intentions to adopt IT innovations. Both variables are expressions of the characteristics of the object of adoption—the innovation itself. But innovations do not spring spontaneously out of nowhere; they are produced, packaged, and/or championed by individuals and organizations. Models of interpersonal and organization trust have been used to explain the risk-taking behaviors of trustors with respect to trustees. When individuals and organizations are associated with and inseparable from innovations they propagate, adoption is arguably a function not only of perceptions about the innovation, but also reflective of the trust that potential adopters have in the inventors, producers, or champions of innovation. Adoption does not occur in a vacuum; adopters are influenced by the social context of adoption, and as TAM2 has demonstrated, subjective norms influence adopter intentions. Thus, innovation-centered models (TAM) and social context models (TAM2) have been augmented by trust models to more fully explain the individual adoption intentions and behaviors. Trust-TAM models (Wang and Benbasat 2005; Geffen et al 2003a; Geffen et al 2003b), as they are called, have been used in an online shopping context, for example.

Trust-TAM is used here to explain software component adoption. In the software component adoption context, a component is an innovation object about which a potential adopter, the reuse consumer, develops usage related beliefs. Subjective norms expressing positive or negative expectations concerning reuse, emanating from peer and supervisor influences, are relevant. Reusable components also have a producer, and when the producer is known to consumers, perceptions of producer trustworthiness are formed in the minds of consumers. Therefore, it is hypothesized that component adoption, i.e. reuse, is
influenced by both ease of use and usefulness perceptions, as well as trust in the producer. The complete research model for this study, combining constructs from TAM, TAM2 and trust, is depicted in Figure 2 below.

![Figure 2 - Research model](image)

We propose to apply Davis’ findings about beliefs affecting attitudes towards technology use for explaining individual software reuse. Beliefs about reusable components 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 ease-of-use and usefulness of a software component will be positively associated with the attitude toward reusing 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:

**H2:** 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, and Venkatesh and Davis (2000) 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 component adoption 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,
H3: Organizational norms that are supportive (unsupportive) of component adoption will be positively (negatively) associated with a developer’s intention to adopt 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 consumer might develop trust in the producer based on his perceptions that the producer was a competent developer. The consumer might also develop trust based on the producer’s adherence to a set of software development 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 help and support during the adoption process.

H4: 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.

H5: A potential 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 adopt 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. Component adoption 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,

H6: A software developer’s trust in the component producer will be positively associated with the software developer’s intention to adopt that component.

See Table 1 below for a list of constructs and definitions.

<table>
<thead>
<tr>
<th>Construct</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Propensity to trust</td>
<td>general willingness to trust others</td>
</tr>
<tr>
<td>Reuse norms</td>
<td>perception that managers, peer developers, and client representatives support or discourage reuse behavior</td>
</tr>
<tr>
<td>Component characteristics (U + EOU)</td>
<td>beliefs that using the component will make a significant, positive impact on software project outcomes, such as quality and efficiency (U), and that that using the component will be relatively free of effort (EOU)</td>
</tr>
<tr>
<td>Attitude toward component reuse</td>
<td>the overall favorableness or unfavorableness of an individual’s feelings toward using the component on a software project</td>
</tr>
<tr>
<td>Perceived trustworthiness of producer</td>
<td>the software developer’s perceptions of another developer’s ability (trustee’s competence in the area related to the task of the transaction), benevolence (extent to which a trustee is believed to want to do good to the trustor), and integrity (trustor’s perception that the trustee adheres to a set of principles that the trustor finds acceptable)</td>
</tr>
<tr>
<td>Trust in producer</td>
<td>the willingness of a software developer to be vulnerable to the actions of another developer based on the expectation that the other developer will perform a particular action important to the trustor, irrespective of the ability to monitor or control that other developer</td>
</tr>
<tr>
<td>Intention to reuse</td>
<td>individual’s intention to use component on a future project</td>
</tr>
</tbody>
</table>

Table 1. Construct Definitions
METHODOLOGY

A controlled experiment was conducted to test the research model. Undergraduate CIS students enrolled in a CS1 and CS2 programming sequence were enlisted as participants. Students were freshman with minimal prior programming experience. Component reuse was cast in the context of reusing components produced in previous semesters by other students. There were three experimental manipulations: reuse norms, ease of use and usefulness, and trustworthiness of the producer. Two scenarios were written for each manipulation, one high and one low. That is, one manipulation described a scenario with a high level of the antecedent and the other a low level. As an example, the manipulation for reuse norms is provided below.

Please read this passage before responding to the following items. (High subjective norm)

Your instructor favors the idea of software reuse from the student reuse library. He considers it an efficient approach to software development. He likes to see students incorporate existing, working, and well-tested components in their programs. “Good programmers don’t reinvent the wheel,” he has been known to say. He even offers to assist students that try to reuse code from the student reuse library on their assignments. He goes out of his way to help students understand code obtained from the reuse library, and make it work in their programs. He maintains an FAQ and discussion list on his web page to assist students in integrating the reusable components from the library.

Please read this passage before responding to the following items. (Low subjective norm)

Your instructor disfavors the idea of software reuse from the student reuse library. He views reuse as a sign of laziness. He would rather see students write new code from scratch. “Good programmers take pride in their own work; they do it themselves,” he has been known to say. Your instructor does not prohibit reuse, but he does not offer to help students with problems trying to reuse components from the reuse library, either. His policy is that students must reuse at their own risk. Unlike other instructors, he doesn’t post helpful instructions on reusable components on his web page, or maintain an FAQ list about reuse. The instructor was heard telling a student, “I’ll help you debug your own code, but I won’t help you debug code you are reusing. If you can’t understand the other person’s code, you shouldn’t reuse it. Figure it out, or write it yourself.”

The manipulations were part of an 11-page survey containing prose passages and 51 questionnaire items. Some of the passages were common to all participants. However, participants were randomly assigned to one of eight possible combinations of the manipulation. The questionnaire items all used 7-point Likert scales with bipolar adjectives. The passages and questions were given in a deliberate ordering, with student participants being provided the salient information just prior to needing the information in a given set of passages for responding to items on relevant variables. First, participants were introduced to the “reusability experiment,” and given the items on propensity to trust. The next passage explained the existence of a reuse library, and its intended purpose was given. After reading one of two passages on reuse norms, focusing on what their instructor thinks about reuse, they were asked about reuse norms as a manipulation check and about perceived reuse risk, which should be related to perceived norms. The next passage was one of two manipulations of combined ease-of-use/usefulness of a particular reusable component that may be helpful on a fictional programming assignment. The subjects were then surveyed on component quality (ease-of-use and usefulness) and on their attitude toward using the component. The next passage manipulated their perceptions about the trustworthiness of the producer of the component. They were surveyed on trustworthiness and asked about their trust in the producer. Finally they were given the items on intention to reuse.

Two pilot tests were conducted and the instrument refined. A factor analysis was not conducted because the group sizes were too small, but reliability was assessed using the Cronbach’s Alpha. As a result of the pilot tests, the experimental manipulations were strengthened, the description of the reuse library and its intended purpose was improved and expanded, the prose stating that component reuse was not considered cheating was clarified, and the ordering of sections on the survey were modified.

RESULTS

A total of 109 undergraduate students were surveyed. Approximately equal numbers of students were given each one of eight combinations of the manipulation. The strength of the experimental manipulations was assessed using ANOVA. The F-statistic for all manipulations was significant at 0.01. For each of the three manipulations, participants responded to items
designed to measure their perception of reuse norms, EOU/U, and trustworthiness. The manipulation check suggests that the participants discerned the direction of the manipulation (high or low). That is, where the scenario described the instructor as having a low level of support for component reuse, participants responded on the low end of the Likert scale for reuse survey items. Construct attributes are listed below in Table 2.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Cronbach Alpha</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. EOU/U</td>
<td>.88</td>
<td>0.76</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Trustworthiness</td>
<td>.85</td>
<td>.04</td>
<td>.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Propensity</td>
<td>.86</td>
<td>.07</td>
<td>.15</td>
<td>.59</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Reuse norms</td>
<td>.87</td>
<td>.16</td>
<td>.04</td>
<td>.02</td>
<td>.95</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Attitude</td>
<td>.92</td>
<td>.89</td>
<td>.10</td>
<td>.01</td>
<td>.30</td>
<td>.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. Trust</td>
<td>.79</td>
<td>.04</td>
<td>.92</td>
<td>.20</td>
<td>.05</td>
<td>.12</td>
<td>.87</td>
<td></td>
</tr>
<tr>
<td>7. Intention</td>
<td>.91</td>
<td>.47</td>
<td>.42</td>
<td>.28</td>
<td>.28</td>
<td>.52</td>
<td>.47</td>
<td>.92</td>
</tr>
</tbody>
</table>

Table 2. Construct Attributes

Similar to Wang and Benbasat (2005), the three antecedents of trust (integrity, competence, and benevolence) were modeled as a single reflective second order factor. The results in Table 2 support the reliability and validity of our instrument. Cronbach alpha values all exceed 0.7 and the average variance extracted (AVE) or variance captured by the indicators relative to measurement error for each construct is greater than 0.5.

Data analysis was conducted using structural equation modeling (LISREL). The structural model, path coefficients, explained variance, and significant levels are captured in Figure 3. The analysis shows that all six hypotheses were supported by the experimental data. All path coefficients were significant at the .05 level.

Figure 3 – LISREL results
DISCUSSION

As the results suggest, intention to adopt a software component can be explained by three statistically significant pathways: component perceptions, reuse norms, and producer trust. The premise of this study was that successful intra-organizational reuse is more than just a technical issue but includes organizational issues, especially trust and social norms. The answer to our research question is that trust and social norms appear to play a significant role in peer-to-peer reuse of software components in an organizational setting.

Although we cannot draw any definite conclusions due to the relatively low level of experience of the component adopters, based on the content of the scenarios used in this study, the results suggest that managers who positively promote software component reuse as efficient, smart, and an expression of quality and provide real support for reuse can have a significant impact on component adoption by developers. Likewise, a manager who strives to create a culture of quality and invests and develops her programmers will foster employees who engender and project trustworthiness to other developers. This in turn, will make developers more likely to adopt and reuse a software component written by another developer.

LIMITATIONS

The use of student subjects presents a limitation in this study. The theoretical context of the model calls for an organizational setting and the development of dyadic trust among professionals. The professional and organizational setting does not exist in the classroom in the same way that it does in industry. We believe, however, that the scenarios constructed provided a theoretically valid operationalization of the research constructs while communicating a meaningfully realistic context for student subjects. The experimental controls and data collection instructions used in the classroom design can be easily refitted for a sample of experienced developers in realistic organizational situations.

IMPLICATIONS

Our theoretical framework provides 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.

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 that promote trustworthiness with potential adopters, and use language, symbols, and rewards that create cultural norms that encourage reuse.

FUTURE DIRECTIONS

Future directions include replicating the current study to an organizational setting using professional software developers. Another direction would be to extend the Trust-TAM model to other innovation contexts. An extension to the model include the feedback loops predicted by trust models, going from outcomes of risk-taking, i.e. technology adoption, that would affect trust. Web 2.0 technologies promote collaborative environments using innovative technologies. TAM-Trust would be useful in explaining the impacts of both the innovation and the social context on technology adoption and use.

REFERENCES