Perceived control and the diffusion of software process innovations

Gina C. Green\textsuperscript{a,*}, Rosann Webb Collins\textsuperscript{b,1}, Alan R. Hevner\textsuperscript{b,2}

\textsuperscript{a}Information Systems Department, Hankamer School of Business, Baylor University, P.O. Box 98005, Waco, TX 76798, USA
\textsuperscript{b}Information Systems and Decision Sciences Department, College of Business Administration, University of South Florida, Tampa, FL 33620, USA

Abstract

Emphasis on quality, productivity, and repeatable processes has led to the introduction of innovative software development tools and techniques. Evidence from use of these innovations shows significant gains in developer productivity and software quality. However, many potentially beneficial innovations are not widely diffused. We examine why this is so, focusing on how a software developer’s perceived control over use of a software development innovation impacts the successful diffusion of the innovation. A proposed research model shows that a developer’s perception of freedom in deciding whether to adopt an innovation and in deciding when and how to apply the innovation impacts IT diffusion success. Survey data were collected from practicing developers trained in the Personal Software Process\textsuperscript{SM} (PSP\textsuperscript{SM}) and are analyzed using structural equation modeling. Results suggest that creating a perception that IT adoption and application is an individual decision, while providing moderate controls governing use is important in impacting diffusion success.

© 2003 Elsevier Inc. All rights reserved.

Keywords: Software development innovations; Software process innovations; Diffusion; IT use; Personal software process\textsuperscript{SM}, (PSP\textsuperscript{SM}); Perceived control; Personal control

1. Introduction

It is widely accepted that current software development practices have led to a “software crisis” that threatens the future promise of the information age (Gibbs, 1994). Billions of dollars are spent and a large proportion of those expenditures wasted every year on software systems that are either never...
completed or, if deployed, are of poor quality. In response to this crisis, many software development tools and practices have emerged from the software engineering discipline to improve the effectiveness and efficiency of software development. These tools and practices have included structured programming (Parnas, Clements, & Weiss, 1985; Wirth, 1971), CASE tools, object-oriented development, software inspections (Fagan, 1976), software certification (Poore, Mills, & Mutchler, 1993), personal and team software processes (Humphrey, 1995), and extreme programming (Beck, 2000). Many of these innovations have demonstrated quantifiable improvements in programmer productivity, development costs, and software quality when used (Iivari, 1996; Linger, 1994; Williams, Kessler, Cunningham, & Jeffries, 2000). However, the transition of these research ideas and technologies into routine3 software development practice has proven to be very difficult (Fayad, Tsai, & Fulghum, 1996; Fichman & Kemerer, 1997; Luqi & Goguen, 1997; Pfleeger & Hatton, 1997). When the innovation in question involves a change to software development practices, there are particular difficulties in encouraging individual developers, many of whom feel that software development is an ad hoc combination of technical skills and individualized artistry, to adopt and sustain use of disciplined, repeatable processes.

Recent research suggests that diffusing innovative practices in IT organizations remains a management challenge. In a workshop conducted with 20 senior IT managers, Basadur, Potworowski, Pollice, and Fedorowicz (2000) found that putting new technology to use more quickly in IT organizations was one of the top challenges faced in the management of technology. Latour, Hanna, Miller, and Pitts (2002) confirm the continued importance of understanding the diffusion of innovations, noting that increased understanding of this process “has major implications for our entire society” (Latour et al., 2002). Ravichandran (1999) identifies the understanding of systems and practices that should be utilized to adopt and diffuse innovations as one of the key research questions that needs to be addressed by innovation theory. Further, Cho and Kim (2002) call for more research into understanding individual-level acceptance of process innovations. Our research seeks to address these critical issues by identifying and understanding how management practices regarding software process innovations (SPIs) impact developers’ perceptions of personal control, which in turn affect developer satisfaction with and use of the software development process innovation.

2. Theoretical framework

The context for this research is the diffusion of SPIs in IS development organizations. Fichman and Kemerer (1997) characterize SPIs as technologies that change an IT group’s process for developing software applications. Under this definition, SPIs include technologies such as CASE tools, OO technologies, and software reuse. This research is particularly focused on that subset of SPIs that prescribe engineering-like discipline in software development practices. In particular, we are interested in process innovations that enforce definition, assembly, measurement, and improvement of software development activities. Examples of such SPIs include software reuse, defect prevention/removal, statistical testing and verification, and software inspections.

---

3 “Routine” development practice coincides with what Kwon and Zmud (1987) call “routinization” in their six-stage model of IT implementation. Routinization indicates that use of the technology innovation is part of one’s normal development activity. Routine use of a technology innovation is a desired outcome of technology diffusion (Agarwal & Prasad, 1997).
Previous diffusion research has identified several factors that are important to successful IT diffusion in general. Some of these studies have identified industry characteristics such as the pool of resources available (Cho & Kim, 2002) and organizational characteristics such as development methodologies currently used, training and education, and management support (Davis, 1989; Green & Hevner, 2000; Ivari, 1996; Orlikowski, 1993) as being important to innovation diffusion. Many traditional innovations and IS researchers have focused on the role of perceived innovation characteristics in explaining IT adoption and use (Moore & Benbasat, 1991; Rogers, 1982). Especially prevalent are studies based on the technology acceptance model (TAM) (Davis, 1989; Davis, Bagozzi, & Warshaw, 1989).

The TAM is an IT-based adaptation of the theory of reasoned action (TRA) (Fishbein & Ajzen, 1975), which states that beliefs about an innovation influence attitudes towards using the innovation, attitudes influence one’s intentions to use the innovation, and intentions determine subsequent usage behavior. The TAM, shown in Fig. 1, identifies two beliefs/characteristics about an innovation that impacts one’s attitude towards innovation use, namely, perceived usefulness of the innovation and perceived ease of use of the innovation. TAM has been found to account for approximately 34–47% variance in IT usage behavior (Adams, Nelson, & Todd, 1992; Mathieson, 1991; Taylor & Todd, 1995). The TAM model has been extended on many occasions to include a wide variety of individual factors that impact IT use through their influences on perceived usefulness and perceived ease of use (Agarwal & Prasad, 1999; Jackson, Chow, & Leitch, 1997; Venkatesh, Speier, & Morris, 2002).

The TRA was extended by Ajzen (1991) to account for situations of mandated innovation use, which the TRA lacked; this extension is reflected in the theory of planned behavior (TPB). As Fig. 1 shows, the TPB posits that one’s usage behavior is influenced by a combination of attitude, subjective norms (one’s perception of the influence of others in their social environment), and perceived behavioral control (one’s perception of the freedom they have to perform the behavior of interest).

Fig. 1. TAM and TPB Models (constructs and relationships in dotted lines represent focus of current research).
Several studies have confirmed the importance of the two additional TPB variables in understanding IT diffusion (Liao, Shao, Wang, & Chen, 1999; Mathieson, 1991; Taylor & Todd, 1995). However, while much research has been done on factors that influence attitudes (Agarwal & Prasad, 1999; Chau & Hu, 2001; Hong, Thong, Wong, & Tam, 2001; Venkatesh & Morris, 2000; Venkatesh et al., 2002) and subjective norms (Gallivan, 2001; Venkatesh & Morris, 2000), very little is known about how to influence perceived behavioral control. Further, the impacts of perceived behavioral control have not been examined in the context of SPIs.

Understanding perceived control is particularly important in the SPI diffusion context since many organizations employ formal controls (such as mandated use) to ensure the implementation of adopted innovations (Ravichandran, 1999), yet research has shown that employees who perceive more personal control are more accepting of workplace changes (Kurland & Cooper, 2002; Wanberg & Banas, 2000). Indeed, Collins, Hull, and Hage (1996) find that leaders in adoption of new process innovations emphasize personal control by employees. Our focus on perceived control is consistent with Fichman’s (2000) call for studies that examine the latter stages of technology assimilation: After formal adoption by an organization, what impacts decisions by individuals about whether and how to use an innovation?

2.1. Research model

The research model, shown in Fig. 2, proposes that a developer’s perception of control over innovation usage impacts successful diffusion of the innovation. The model expands on traditional TAM- and TPB-based diffusion models in two key ways: (1) it includes two key measures of diffusion success, and (2) it depicts a multidimensional view of personal control.

![Diffusion of SPIs: research model.](image-url)
2.1.1. Diffusion success

Research in IS implementation and diffusion of innovations finds that successful implementations of innovations are indicated by sustained use or diffusion of the innovation in the organization (Fowler & Levine, 1993; Rogers, 1982). Thus, a key measure of the successful diffusion of an information technology (IT) in an organization is the use of the IT. In their review of research on IS success, DeLone and McLean (1992) observe that IT use is the most frequently reported measure of IT implementation success. Thus, use has been chosen as one of the indicators of IT diffusion success in the proposed research model. Cooper and Zmud (1990) specify several use-related aspects of the implementation process (e.g., an innovation is available for use; use “is encouraged as a normal activity”). In this study, the focus is on acceptance of the innovation; it “is employed in organizational work.” In particular, we study the frequency of SPI use in terms of both time and how often the SPI is used in projects.

When use of an IT is mandated, as is often the case when introducing new software development technologies, satisfaction has been suggested as a more appropriate measure of IT success (Adams et al., 1992; Chau, 1996; DeLone & McLean, 1992; Melone, 1990). Chau (1996) observes that when software development innovations are adopted, the adoption decision is often made at an organizational level and use by developers is mandated. However, he notes that the issue of whether or not the software developer enjoys using the innovation (i.e., his satisfaction with the innovation) is of critical importance to productivity. Highly satisfied users are particularly important in the early stages of innovation adoption since the early adopters can serve as influential opinion leaders in an organization (Leonard-Barton, 1987). Thus, satisfaction with the IT has been chosen as a second IT diffusion success variable in the framework.

The model also posits a relationship between these two IT diffusion variables, consistent with the TAM and TPB frameworks that suggest that attitudes influence behavior. DeLone and McLean (1992) argue that increases in satisfaction with IT use are related to increased usage of the IT; they do not, however, empirically examine this relationship. Rai, Lang, and Welker (2002) explore this relationship in a study of IT usage by university personnel. They find that higher levels of satisfaction with the IT are associated with greater levels of IT use, supporting the relationship between satisfaction and use. However, their study is limited to conditions of voluntary use. Our research examines the relationship between satisfaction and use using actual software developers, as well as conditions ranging from voluntary to mandated use. The following hypothesis is offered:

H1. As software developers’ level of satisfaction with an SPI increases, their level of SPI use will increase.

2.1.2. Perception of control

Prior research suggests that workers in general and IS workers in particular desire more work-related freedom (Lending & Chervany, 1998; Seeman, 1983). Autonomy in work is considered a key characteristic of professionals (Hall, 1968). Lending and Chervany (1998) attempt to explain the low use of CASE tools by arguing that CASE tools are typically used in conjunction with formal development methods that take some freedom and creativity away from software developers. They note that developers prefer jobs with high autonomy and may thus resist the use of tools that are associated with loss of autonomy. Ross and Wright (1998) associate autonomy of work with workers having freedom to decide what to do and how to do it. Many organizations establish formal managerial
controls in order to promote the use of adopted innovations (Ravichandran, 1999). At the innovation level, many SPIs promote the use of more formal development methods in order to produce higher quality systems more efficiently (Luqi & Goguen, 1997). The present research examines how the potential loss of autonomy or decision-making freedom (i.e., loss of perceived control) in these environments impacts software developers’ ongoing use of SPIs.

According to social psychology research, an individual seeks control because of the need to know causes and consequences of his and others’ behaviors (Baronas & Louis, 1988). Personal control has been defined as “an individual’s beliefs, at a given point in time, in his or her ability to effect a change, in a desired direction” (Greenberger & Strasser, 1986, p. 165). Perceived control has been used in social psychology and organizational sciences as a global concept, but other research has identified important subdimensions of personal control and examined their differential impacts. Table 1 lists the subdimensions that have been used in prior research; these are categorized into three variables and then defined in the study context.

Voluntariness of use is a continuous variable that can range from strictly voluntary to strictly involuntary. The degree to which use of an innovation is perceived as voluntary has been found to impact attitudes toward use of an innovation (Moore, 1989), postadoption intentions to continue use (Karahanna, Straub, & Chervany, 1999), and extent of current IT use (Agarwal & Prasad, 1997; Iivari, 1996). These studies find significant, negative relationships between perceived freedom to adopt an innovation and innovation use. In other words, organizational mandates have been found to increase innovation use. This relationship has been well studied in the context of technology innovations but has not been tested in the context of SPIs. Thus, we hypothesize the following:

H2. As software developers’ perceptions of voluntariness in using an SPI increase (i.e., less perceived organizational mandate), their level of SPI use will decrease.

While voluntariness of use has been found in previous IS research to directly impact innovation use, less explored in IS literature are indirect impacts of voluntariness on IT use via its relationships with other perceived control variables. Evidence in organizational behavior literature suggests a relationship between voluntariness and perceived choice control (i.e., control over when to apply an SPI innovation after an adoption decision has been made). In a study of water saving technology adoption/investment, Lynne, Casey, Hodges, and Rahmani (1995) found that perceived control in the adoption decision is significantly related to perceived control over the decision to invest additional capital in the technology.

H2. As software developers’ perceptions of voluntariness in using an SPI increase (i.e., less perceived organizational mandate), their level of SPI use will decrease.

While voluntariness of use has been found in previous IS research to directly impact innovation use, less explored in IS literature are indirect impacts of voluntariness on IT use via its relationships with other perceived control variables. Evidence in organizational behavior literature suggests a relationship between voluntariness and perceived choice control (i.e., control over when to apply an SPI innovation after an adoption decision has been made). In a study of water saving technology adoption/investment, Lynne, Casey, Hodges, and Rahmani (1995) found that perceived control in the adoption decision is significantly related to perceived control over the decision to invest additional capital in the technology.

IS research has suggested, but not tested, a relationship between perceived voluntariness and perceived choice control. Hebert and Benbasat (1994) examine the impact of voluntariness on the use of IT in a hospital setting. They found that while users, on average, rated perceived voluntariness as neutral on their surveys, comments from them suggest that they did not perceive a wide range of uses and choices available to them during innovation use; they felt their behavior was dictated by organizational policy. The authors suggest that managers should “underscore the element of choice in the range of applications [of the innovation] or the way [the innovation] may be used” to ensure successful diffusion of the innovation (emphasis added). Gallivan (2001) presents a diffusion framework that suggests a secondary/individual adoption phase after organizational-level innovation adoption. This secondary adoption refers to “when and how [individuals] adopt” the innovation and is influenced by an individual’s perception of expectations for use. Gallivan’s suggestion is congruent with the notion that
the amount of control one perceives over when to apply an innovation can be influenced by the degree of perceived voluntariness of use. Therefore, we argue that when developers perceive more freedom in the selection and adoption of the innovation, they will associate that freedom with more flexibility in deciding when to appropriately use the innovation:

**H3.** As software developers’ perceptions of voluntariness in using an SPI increase, their perceived control over choice in when to use the SPI will increase.

In the study context, perceived process control is seen as the degree of developer discretion over how to best appropriate the SPI innovation in development activities. In a study of virtual banking adoption,
Liao et al. (1999) hypothesize a relationship between perceived voluntariness and perceived behavioral control, the latter construct analogous to perceived process control. Due to lack of reliable measures however, this hypothesis is not tested empirically. Benham and Raymond (1996) examine the adoption of voice mail technology and find that beliefs about voluntariness (the degree to which the use of voice mail was perceived to be voluntary) determine a user’s perceived behavioral control in the use of voice mail. Thus,

**H4.** As software developers’ perceptions of voluntariness in using an SPI increase, their perceived process control over how to use the SPI will increase.

Our interest in this study is in understanding how management actions may influence perceptions of control and how this perceived control impacts successful SPI diffusion. Baronas and Louis (1988) suggest that enhanced perceptions of choice control are associated with increased levels of developer satisfaction. In an exploratory field experiment, they assign implementation-related tasks to experimental groups and manipulate perceived choice by allowing treatment group subjects to select task deadlines. Although success of the perceived choice manipulation is inconclusive, study results provide partial support for their hypothesis. Psychology research in other contexts has found a significant, positive relationship between choice control and satisfaction (Giacobbe-Miller, 1995; Langer, 1983). Thus,

**H5.** As software developers’ perceptions of choice in when to use an SPI increase, their level of satisfaction with the SPI will increase.

In the job satisfaction literature, control over work methods and task activities is generally found to be motivating (e.g., Aldag & Brief, 1975; Hackman & Oldham, 1976). For example, Tetrick and LaRocco (1987) study the effects of process control on job satisfaction and find that when workers feel they have greater influence over the events in their work environment, they are more satisfied in their jobs. Griffin (1987) discusses the interdependence between specific satisfactions (about definable dimensions of a job) and general satisfactions (overall job satisfaction based on a composite of specific satisfactions) and posits that specific and general satisfactions have some degree of shared variance. Griffin points out that more precise and specific relationships between job characteristics and satisfaction are likely to be stronger and argues that the goal “should be to optimize the degree of specificity incorporated into any one study” (Griffin, 1987, p. 116). The present research is focused on a more specific satisfaction (with an SPI) and the specific job characteristic of process control. It is expected that the relationship between process control and satisfaction, measured at this more specific level, will be congruent with the findings from studies at the general level:

**H6.** As software developers’ perceptions of process control over how to apply an SPI increase, their level of satisfaction with the SPI will increase.

2.2. Software development process innovation: the personal software process SM (PSPSM)

The innovative software technology chosen for this study is the personal software process (PSP). PSP is a process framework and set of software development methods that help individual developers become more disciplined and productive. The focus is to challenge individuals to continuously improve their
The PSP training is presented in an intensive 120-hour course that instructs software developers how to estimate and plan their projects, measure and record their work products, and improve the quality of the resulting software systems (Humphrey, 1997). PSP training moves the software developer through four levels of maturity in the development process. Each level requires the application of task scripts, data forms, and standards.

- **PSP 0**—Developers document their current practices and learn how to record productivity and quality metrics. Initial concepts of process improvement include the identification of process problems to be eliminated.
- **PSP 1**—Project planning and estimating based on historical baseline data are taught. Developers learn to track their work and record metrics on progress.
- **PSP 2**—Design and code reviews are introduced along with quality measurement and evaluation. Effective specification methods and defect prevention are emphasized.
- **PSP 3**—At the highest level of PSP maturity, developers are instructed how to adapt the PSP process and methods to their own working environments.

Results of PSP applications on industrial projects have been positive (Ferguson, Humphrey, Khajenoori, Macke, & Matvya, 1997). Data from three industrial case studies demonstrate that PSP training substantially improves estimating and planning abilities of developers while significantly reducing defects in software products. Other experience reports show that while PSP training provides important long-term benefits to individual developers (e.g., greater skills and discipline), difficulties often arise in the transition of the PSP process and methods into industrial practice (Kamatar & Hayes, 2000; Morisio, 2000).

There are several advantages to the selection of PSP as the innovation through which the research model is tested. The software engineering institute (SEI) plays an important role in creating the organizing vision for a novel SPI technology like PSP so that an organization’s uncertainty about the innovation’s advantages and benefits, required organizational changes, and ways to implement the innovation are significantly reduced (Swanson & Ramiller, 1997). SEI also provides PSP training, which reduces the knowledge barriers to deploying the innovation (Fichman, 2000). In addition, PSP is a software development innovation that is implemented at an individual level. The premise of PSP is that the individual developers must be trained in the most effective software development practices and then be held responsible for their productivity and quality results. Thus, it is interesting to explore if and how individual control in development activities is important when such increased responsibility is given to developers.

### 3. Research methodology

The research is conducted as a field survey. A field survey is appropriate as the use of data from practicing software developers in multiple organizations increases external validity of the results of the study. The results of the study can be more confidently applied across a wide population of software developers and development organizations, an important aspect of applied research. Moreover, there are
calls for more multiorganizational research on single innovations rather than the single-site case studies common in the implementation literature (Klein & Sorra, 1996).

3.1. Data collection procedures

A Cooperative Research and Development Agreement (CRADA) between the SEI at Carnegie Mellon University, Baylor University, and the University of South Florida supported this research. The SEI holds the licensing agreements for the PSP. All industrial PSP training is provided by instructors licensed by the SEI. Organizations and individuals who have received PSP training were contacted by the SEI to evaluate eligibility (e.g., use of the PSP in industrial projects for at least 3 months and less than 2 years) and interest.

Data to test the research hypotheses were collected via a questionnaire developed for this study. Where possible standard scales for constructs were used (see Table 5 for the source of scales). The questionnaire was pretested with IS professors, graduate students, and practitioners in order to ensure

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Summary of demographic information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable</td>
<td>Value</td>
</tr>
<tr>
<td>Gender</td>
<td></td>
</tr>
<tr>
<td>Male</td>
<td>81%</td>
</tr>
<tr>
<td>Female</td>
<td>19%</td>
</tr>
<tr>
<td>Age</td>
<td>Mean = 32.9</td>
</tr>
<tr>
<td></td>
<td>Range = 21 – 60</td>
</tr>
<tr>
<td>Position</td>
<td></td>
</tr>
<tr>
<td>Analyst</td>
<td>11%</td>
</tr>
<tr>
<td>Designer</td>
<td>21%</td>
</tr>
<tr>
<td>Manager</td>
<td>6%</td>
</tr>
<tr>
<td>Programmer</td>
<td>49%</td>
</tr>
<tr>
<td>Project manager</td>
<td>13%</td>
</tr>
<tr>
<td>Highest degree held</td>
<td></td>
</tr>
<tr>
<td>Bachelors</td>
<td>54%</td>
</tr>
<tr>
<td>Masters</td>
<td>43%</td>
</tr>
<tr>
<td>PhD</td>
<td>3%</td>
</tr>
<tr>
<td>Years of IS experience</td>
<td>Mean = 8.0</td>
</tr>
<tr>
<td></td>
<td>Range = 1 – 25</td>
</tr>
<tr>
<td>Months of PSP experience</td>
<td>Mean = 12.8</td>
</tr>
<tr>
<td></td>
<td>Range = 3 – 24</td>
</tr>
<tr>
<td>Origin of respondent (based on organization)</td>
<td></td>
</tr>
<tr>
<td>U.S.</td>
<td>76%</td>
</tr>
<tr>
<td>Non-U.S.</td>
<td>24%</td>
</tr>
</tbody>
</table>
content validity. With the exception of one item measuring extent of use, which used a five-point Likert scale, all items contained in the questionnaire used seven-point Likert scales. A pilot study with 10 practitioners using PSP on a large industrial project further ensured content validity and provided early data for hypotheses testing.

Surveys were distributed to 154 IS professionals who agreed to participate in the study. Of the 154 surveys distributed, 71 completed surveys were returned, resulting in a response rate of 46%. Of the 71 completed surveys, 8 were unusable, resulting in a study sample size of 63. Table 2 shows a summary of the sample characteristics. The survey sample represents a representative cross section of software developers from 24 U.S. and non-U.S. organizations. Table 3 shows a breakdown of the respondents by organization. Of the 24 organizations represented, 7 organizations had multiple respondents. Table 3 also shows a breakdown of the types of industries represented by the organizations. To control for potential organizational effects, the data analysis model included an organizational variable; no significant organizational effects were found.

4. Results

Descriptive data on model variables and PSP usage are presented in Table 4. The data show that PSP has been used by software developers on an average of two projects (mean = 2.4). Seventy-five percent of these projects were either mission-critical projects or were a mixture of critical and noncritical projects. Approximately one quarter of the software developers surveyed use PSP techniques on all their software development projects. PSP tends to be used across all phases of the software development process but is used most heavily in design and implementation activities.

The approach used to analyze the study data is a two-stage approach using SPSS to analyze reliability and validity of the measurement items and PLS Graph to analyze the structural model.
4.1. Measurement model

The measurement model is assessed in terms of reliability of measures and construct validity. Reliability of scales was assessed using Cronbach’s alpha. Results of the reliability assessment are shown in Table 5. All the original scales exhibit good reliability, ranging from .71 to .90. Scales were further examined to see if improvements could be made to reliability by removing individual scale items. Table 5 shows that three scale items were removed, resulting in a final set of scales with Cronbach’s alpha levels ranging from .71 to .94, comparable to results of previous studies in the research area.
Convergent validity was assessed by a principal axis factor analysis on each scale, extracting factors with eigenvalues greater than one. One factor was extracted in each case, with factor loadings ranging from .75 to .94. Discriminant validity was assessed by completing a principal axis factor analysis on all measurement items. Five factors were extracted in this analysis. Table 6 shows the results of this analysis—all items load strongly on their corresponding factors. There are three items that cross load on different factors. Each case involved items measuring aspects of perceived control. However, good discriminant validity is evidenced by the fact that correlations between factors are significantly less than 1, as shown in Table 7 (Purvis, Sambamurthy, & Zmud, 2001).
4.2. Structural model

Hypothesis testing was conducted using a partial least squares (PLS) analysis on the structural model. The PLS method of analysis is suitable for this study because it allows us to estimate the path model while requiring a smaller sample size than covariance-based techniques such as LISREL (Chin & Newsted, 1999). Because some research has suggested that work experience may influence attitudes towards development, the PLS analysis included IS experience and PSP experience as control variables; however, neither experience variable was found to be significant at \( P=.1 \). Therefore, the results diagram does not include these variables. The results of testing the structural model are shown in Fig. 3.

Hypothesized relationships are tested by examining the direction and significance (estimated by a bootstrapping procedure) of the standardized path coefficients in the research model. We find support for

Table 6
Factor analysis with all items

<table>
<thead>
<tr>
<th>Scale item</th>
<th>Factor 1</th>
<th>Factor 2</th>
<th>Factor 3</th>
<th>Factor 4</th>
<th>Factor 5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Perceived voluntariness</strong></td>
<td>.92</td>
<td>.43</td>
<td>.51</td>
<td>-.06</td>
<td>.13</td>
</tr>
<tr>
<td>...superiors expect me to use PSP</td>
<td>.83</td>
<td>.60</td>
<td>.40</td>
<td>.08</td>
<td>.09</td>
</tr>
<tr>
<td>...use of PSP is voluntary</td>
<td>.87</td>
<td>.47</td>
<td>.45</td>
<td>-.03</td>
<td>.36</td>
</tr>
<tr>
<td>...supervisor does not require me to use PSP</td>
<td>.86</td>
<td>.50</td>
<td>.45</td>
<td>-.04</td>
<td>.44</td>
</tr>
<tr>
<td>...using PSP is not compulsory</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Perceived choice control</strong></td>
<td>.49</td>
<td>.97</td>
<td>.14</td>
<td>.40</td>
<td>-.07</td>
</tr>
<tr>
<td>...can you decide what parts of PSP you will use</td>
<td>.54</td>
<td>.92</td>
<td>.08</td>
<td>.45</td>
<td>-.00</td>
</tr>
<tr>
<td><strong>Perceived process control</strong></td>
<td>.71</td>
<td>.28</td>
<td>-.81</td>
<td>-.09</td>
<td>.10</td>
</tr>
<tr>
<td>...prespecified sequence of steps that I was required to follow</td>
<td>.36</td>
<td>.07</td>
<td>-.78</td>
<td>-.25</td>
<td>.23</td>
</tr>
<tr>
<td><strong>Satisfaction with PSP</strong></td>
<td>.02</td>
<td>.33</td>
<td>-.24</td>
<td>.91</td>
<td>-.21</td>
</tr>
<tr>
<td>...PSP is exactly what I need</td>
<td>.09</td>
<td>.42</td>
<td>-.23</td>
<td>.86</td>
<td>-.21</td>
</tr>
<tr>
<td><strong>Frequency of PSP use</strong></td>
<td>-.30</td>
<td>-.02</td>
<td>-.25</td>
<td>.22</td>
<td>-.92</td>
</tr>
<tr>
<td>...proportion of projects I use PSP with</td>
<td>-.17</td>
<td>.10</td>
<td>-.19</td>
<td>.26</td>
<td>-.91</td>
</tr>
<tr>
<td><strong>Percentage of variation explained</strong></td>
<td>41.05</td>
<td>6.87</td>
<td>5.56</td>
<td>23.47</td>
<td>11.97</td>
</tr>
</tbody>
</table>

4.2. Structural model

Hypothesis testing was conducted using a partial least squares (PLS) analysis on the structural model. The PLS method of analysis is suitable for this study because it allows us to estimate the path model while requiring a smaller sample size than covariance-based techniques such as LISREL (Chin & Newsted, 1999). Because some research has suggested that work experience may influence attitudes towards development, the PLS analysis included IS experience and PSP experience as control variables; however, neither experience variable was found to be significant at \( P=.1 \). Therefore, the results diagram does not include these variables. The results of testing the structural model are shown in Fig. 3.

Hypothesized relationships are tested by examining the direction and significance (estimated by a bootstrapping procedure) of the standardized path coefficients in the research model. We find support for

Table 7
Factor correlations

<table>
<thead>
<tr>
<th>Factor</th>
<th>Factor 1</th>
<th>Factor 2</th>
<th>Factor 3</th>
<th>Factor 4</th>
<th>Factor 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Perceived voluntariness</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Perceived choice control</td>
<td>.53</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Perceived process control</td>
<td>.51</td>
<td>.11</td>
<td>1.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Satisfaction with PSP</td>
<td>.06</td>
<td>.41</td>
<td>-.26</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>5. Frequency of PSP use</td>
<td>.23</td>
<td>-.02</td>
<td>.20</td>
<td>-.23</td>
<td>1.00</td>
</tr>
</tbody>
</table>
all hypothesized relationships, though not necessarily in the hypothesized directions. Increases in developer satisfaction with using PSP are found to be associated with increased frequency of PSP use ($\beta = .33, P < .1$), lending support for H1. Results also support a significant, negative relationship between perceived voluntariness and PSP use as hypothesized in H2 ($\beta = -.33, P < .05$). Finally, results support H3 and H4 relating the perceived control variables. In particular, perceived voluntariness is positively and significantly associated with both perceived choice ($\beta = .56, P < .01$) and perceived process control ($\beta = .63, P < .01$).

Perceived choice in when to use PSP is positively and significantly associated with developer satisfaction with using PSP ($\beta = .48, P < .05$), supporting H5. However, while perceived process control (control over how to appropriate PSP) is found to be significantly related to satisfaction with using PSP, the relationship is in the opposite direction of that stated in H6 ($\beta = -.35, P < .05$). These results are discussed in the next section.

5. Discussion

The results of this study increase our understanding of (1) the relationship between IS diffusion success variables and (2) the relationship between software developer discretion and the diffusion of SPIs.

5.1. SPI diffusion success

Consistent with the relationship posited in Delone and McLean (1992) and with results found under voluntary conditions in Rai et al. (2002), our data analysis results show a positive relationship between
developer satisfaction and use of software development innovations. In particular, increases in developer satisfaction with using PSP are shown to be associated with increases in frequency of PSP use (H1). Since the data collected in this study are cross sectional, it is not possible to test the direction of this relationship. It may be that increasing developer satisfaction with using an SPI can significantly improve the frequency with which the developer uses the innovation in software development activities. It is also possible that use of an innovation, whether mandated or not, combined with dissatisfaction with an innovation would create a cognitive dissonance (Festinger, 1957) that the user would seek to reduce by increasing their perceptions of satisfaction.

5.2. Effect of perceived control on SPI diffusion success

Data analysis results support the importance of perceived control by showing significant relationships between perceived control variables and IT diffusion success. However, the nature of the impact of perceived control on diffusion success is complex. Consistent with previous psychology research (Giacobbe-Miller, 1995; Langer, 1983), increases in developer choice over when to use PSP are found to be associated with increases in developer satisfaction with using PSP (H5). However, increases in developer perceptions of process/behavioral control over how to use PSP are found to be associated with decreases in the developer satisfaction with PSP use (H6). In other words, the more that standards and structure are emphasized in use of PSP, the more satisfied the developer is in using the PSP. These results suggest that while developer discretion in when to use an innovation is related to higher levels of satisfaction with the innovation, too few constraints on how to use the innovation are associated with developer dissatisfaction. Thus, allowing freedom to choose when to use a development innovation while imposing some managerial and/or innovation-related constraints may actually enhance software developer satisfaction with using the innovation.

While the resulting relationship between perceived process control and satisfaction with PSP use contradicts the direction of the relationship in the stated hypothesis (H6) as well as some previous research, it is similar to the findings of Henderson and Lee (1992) who found that decreases in one’s personal process control result in higher levels of IS team performance. While the outcomes of interest in the present study differ from Henderson and Lee, the studies have in common the fact that they examine process control from the standpoint of IS developers. This similarity suggests that there may be differences between IS personnel and/or tasks and personnel/tasks examined in previous studies that may account for the differences in the effect of process control on satisfaction. IS tasks have been noted to represent complex, unstructured tasks (Khalil & Clark, 1989; Vessey & Glass, 1998). By enforcing standards and structure, a software development technique can bring enhanced control to a previously unstructured task environment, thereby reducing overall task complexity and increasing satisfaction.

Results also support posited relationships between the perceived control dimensions. Hypotheses relating increases in software developer perceptions of voluntariness to increases in perceived choice and process control are supported by study data (H3 and H4). Voluntary use of PSP is related to greater perceptions of control over software development tasks. Results suggest that the ability for software developers to decide whether or not a software development innovation should be adopted in their development environment is strongly related to enhancing their overall perception of control over how to accomplish their development tasks with the innovation. While
these results are consistent with results in previous research (e.g., Benham & Raymond, 1996; Lynne et al., 1995), none of the previous studies identify, relate, and test all three perceived control variables together.

The degree to which the adoption of PSP is of free will (voluntariness) is found to be directly and inversely related to frequency of PSP use (H2). This finding is consistent with Karahanna et al. (1999) and confirms that low levels of developer discretion in adopting a software development innovation (e.g., organizational mandates) can serve to increase the frequency with which developers use the innovation in development activities. However, it also suggests that developers, when given a choice of whether or not to adopt a process innovation (in this case, the PSP), generally chose to use it less frequently. An evaluation of comments supplied by survey respondents gives a possible explanation for this relationship. Several respondents cite organizational factors such as lack of time allowance in project schedules and pressures from customers to deliver software systems faster as inhibitors to frequent use of PSP. Thus, when organizational mandates are in place, developers may find it easier to justify the additional time often required to implement process innovations. In the absence of these mandates, developers may find it difficult to justify longer project schedules to customers or other managers.

To summarize, these findings suggest that relationships between perceived control and the diffusion of an innovation are not simple. Developers’ beliefs that they have choice over when to use the software development innovation are associated with greater satisfaction with the innovation. But developers’ perceptions of increased control over how the innovation is to be used in their tasks are associated with less satisfaction. Lower perceived control over whether or not to adopt an innovation (low voluntariness) was found to be directly (positively) and indirectly (positively and negatively) related to use of an innovation.

5.3. Influencing perceived control

In light of this study, managers are encouraged to be attentive to issues of perceived control, and its influence in determining the extent to which SPIs will be used by software developers. The present research can be used to provide guidance in influencing perceived control. Our research suggests that in the context of diffusing process innovations, software developers desire standards and controls that govern how an SPI is to be used. Thus, if the SPI itself does not contain standards that govern the use of the innovation, management should undertake efforts within the organization to develop standards and guidelines to ensure consistency in SPI use.

Lewis, Ross, and Mirowsky (1999) found that opportunities to acquire and apply knowledge increase one’s sense of personal control. Thus, managers may further enhance the potential for SPI diffusion success by making opportunities for training and education on the SPI readily available to software developers. In particular, since increases in choice or decisional control have been found to positively influence SPI diffusion, training and education opportunities should include guidance from experienced individuals inside and outside the organization as to what development situations are good candidates for the particular SPI use. Many developers report low usage of an SPI because they learn (through personal experience, trial, and error) that the SPI is not well suited to their particular development tasks despite strong management encouragement to use the SPI. Training and education on these issues up-front can result in developers having a greater sense of control over when to best apply a given SPI, and thus more satisfaction in using the SPI.
6. Limitations

There are three primary limitations to this cross-sectional study using a single survey instrument. First, causal relationships between constructs cannot be tested using this research method. Second, there is potential for common method bias since data on all study variables were collected using the same instrument. The potential bias in this case is that the relationships between variables result from the method and not relationships between the underlying constructs. In particular, there is concern that respondents, who see the items for all variables, will “guess” the anticipated relationships between those variables. Since PSP is an individual-level SPI, the individual developer was deemed the single best informant for the study and follow-up interviews were not feasible. Future survey research on personal control and SPI diffusion should select an SPI for which multiple informants are appropriate and/or plan for follow-on interviews with respondents. Third, the relatively small number of respondents limit the generalizability of the study findings.

7. Conclusions

This research examines the impact of perceived control on the successful diffusion of innovative software development processes. We integrate and extend existing theory from IT implementation, diffusion of innovations, and social psychology research to advance a research model that enables us to better understand how issues of control impact the diffusion process.

One of the key findings of this research is that software developers’ perceptions of control over their work when using an innovative software development technique are significantly related to the satisfaction of that developer with use of the technique and therefore the ongoing use of the innovation by the developer. This finding is an important contribution to IS research as no existing IS research links perceived control to satisfaction with innovation use, links satisfaction with innovation use to innovation usage, and does so using actual software developers in conditions ranging from voluntary to mandated use.

An additional contribution to IS research is the explication of the general perceived control construct into three related dimensions: perceived voluntariness of innovation use, perceived choice in when to use the innovation, and perceived control over how to use the innovation (process control). Separating this construct into different dimensions allowed us to address the call made by IS researchers to better understand how organizational practices and compliance processes impact the spread of technology (Karahanna et al., 1999; Sharma & Rai, 2000).

Our study has implications for management practice. We found a negative relationship between voluntariness and use, which is consistent with previous research that demonstrates that organizational mandates increase innovation use (Agarwal & Prasad, 1997; Iivari, 1996; Karahanna et al., 1999; Moore, 1989). However, our study goes further to find two indirect relationships between voluntariness and use through its impacts on other perceived control variables.

Our study finds that giving freedom to developers in adopting innovative SPIs (i.e., increased voluntariness) is positively associated with increased feelings of perceived choice control, which is in turn associated with increased satisfaction with and use of the SPI. Thus, managers are encouraged to provide more freedom for individual developers to decide when to apply innovative software development techniques during the development process. Managers are also encouraged to provide
educational opportunities that enhance developer’s abilities to determine what conditions are best suited to applying a particular SPI in a development situation.

Additionally, our study finds that giving too much freedom in how to use an innovation can serve to decrease developers’ satisfaction with using the innovation. This finding suggests to managers that the use of more formal, structured engineering practices in software development should continue to be investigated as a way to provide more controls and more predictability in software development in order to provide greater satisfaction with and sustained use of SPIs. In the absence of formal methods, managers should adopt standards and guidelines on SPI use prior to attempting diffusion within the organization.

The above findings demonstrate to IS managers that while organizational mandates may serve to increase short-term SPI use, these mandates can have corresponding negative impacts on sustained use (Agarwal & Prasad, 1997). These negative impacts can be mitigated by ensuring (1) that developers have a choice in when to use the innovation, and (2) that the innovation (or management) provides more structure for their work.

While this study has established a developer’s perception of discretion or control over development activities as important to the diffusion of SPIs, the study also presents opportunities for future research into perceived control. One such opportunity is the study of additional variables that may provide more explanatory power in understanding the diffusion process. One such variable is one’s desire for personal control, which has been identified in previous research as a potential moderator of the effects of perceived control on satisfaction (Greenberger & Strasser, 1986). Other potential factors include individual characteristics such as personality types (e.g., methodic/rational vs. random/creative), level of motivation, innovation characteristics such as perceived usefulness, as well as organizational factors such as project schedule pressures and level of organization process maturity.

The measure of innovation use in this study focuses on the frequency of use. However, software development innovations such as the PSP actually include several development practices that together improve the discipline and effectiveness of software developers. Thus, a developer could conceivably use a small subset of PSP frequently and not see as significant an increase in effectiveness as a developer who uses a larger subset of PSP. Thus, future research should address the use of a more multifaceted measure of use that captures both frequency and depth of innovation use. Depth could be measured (following Cooper & Zmud, 1990) by asking developers to rate their level of maturity in using PSP on the four levels described earlier.

While this study focuses on the diffusion of SPIs, the theory-based model may offer insights for technology-based innovations as well. Future research can validate this by repeating the study using technology-based IT innovations.

As IS managers introduce new technologies, processes, and procedures to IS developers, they can use the results of this research and future research to design an environment that facilitates the success of SPIs in their organization.

Acknowledgements

This research was supported by the Software Engineering Institute at Carnegie Mellon University through a Cooperative Research and Development Agreement with Baylor University and the University of South Florida.
References


