UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

REASONED AND INSTITUTIONAL EXPLANATIONS FOR THE USE OF SOFTWARE DEVELOPMENT PROJECT MANAGEMENT PRACTICES

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

DOCTOR OF PHILOSOPHY

By

JEFF CRAWFORD
Norman, Oklahoma
2007

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

REASONED AND INSTITUTIONAL EXPLANATIONS FOR THE USE OF SOFTWARE DEVELOPMENT PROJECT MANAGEMENT PRACTICES

A DISSERTATION APPROVED FOR THE
MICHAEL F. PRICE COLLEGE OF BUSINESS

BY

____
Robert W. Zmud, co-chair

____
Shaila Miranda, co-chair

____
Laku Chidambaram

____
R. Leon Price

____
Mark P. Sharfman

____
Rajeev Sharma

Acknowledgements

I suppose that it’s a natural tendency for a person to become reflective when in the process of moving from one phase of life to another. While walking the last mile of this dissertation, I’ve reflected a lot about the tremendous amount of effort and energy it has taken to get to this point. I am convinced that finishing this dissertation is a function of something much greater than the sum of my efforts. The following words are offered as a tribute to those who have played an essential part in the completion of this dissertation.

First and foremost, I thank my God and Father for the blessings that have been lavished on me over the past five years. The Ph.D. process has provided an opportunity to grow and mature as a person and, more importantly, as a Christian. Jen and I have constantly been reminded during these years that we must rely on the Father in order to survive, and I pray that lesson will never be forgotten. I have also become convicted that with blessings come responsibility and I pray that we will use the gifts we’ve gained during this time to reflect God’s goodness and light in this world. To Him be the glory.

Second, I am indebted to Jen and our kiddos who daily filled me with the strength and encouragement necessary to undertake and complete this work. Jen, my beautiful and amazing wife, has borne a tremendous weight during the past five years so that this dream could be realized. She is my energy and life, and I believe with all my heart that this dissertation is as much hers as mine (Jenny, my love for you is deathless). My three children have also born the brunt of this process and have sacrificed a tremendous amount during this time. Maddie, who was always quick to ask “did you finish your paper today?”, encouraged me so much during the times when I wanted to throw in the towel. Abby made me smile time and time again, and helped me maintain a positive outlook when things seemed bleak. Nathan forced me to get out of the chair when I was mentally exhausted and do something active (usually wrestle), even on days when I felt like I just wanted to go back to bed. My family provided a safe place for me to go every day, a place where I knew I would be loved, and for that I owe the deepest amount of gratitude.

I also have a hard time using words to express the appreciation I have for the role my family played in this process. My dad and mom took great pains to encourage me with words and action. Consistent with how they have always lived, they led by example and gave me the courage and perspective to finish. I am because they are. It was my

brother Chris that introduced me to the field of MIS, and largely because of him that I eventually pursued a Ph.D. in MIS. My brother Paul helped me to find a passion and purpose in life, and has been my primary mentor for the past 10 years. I have always looked to both of my brothers as an example, and they have never disappointed. I also thank their wives, Jennifer and Alissa, who along with my wife Jennifer, have endured our nerdy ways and still love us just the same. I am also eternally grateful to Jen's parents, Ron and Georgia, who encouraged us along the way.

I owe a special debt of gratitude to my friends who helped in this process. The folks at MidFirst Bank, especially Garland Wilkinson and George Records, were exceedingly kind to me during these last few years and played an invaluable part in the realization of this dissertation. The staff at Edmond church of Christ encouraged me constantly over the past five years, and just as importantly provided me a quiet room where I could go at any time to study and write. My fellow Ph.D. students, Yong-Mi Kim, Leslie Albert, Rui Huang, Viet Dao and Victor Law also provided perspective and expertise that has been invaluable during this process. To my friends Mark Hamlin and Chris Carr, I can't express enough my appreciation for including me in their pen-making "guild" (woodworking proved to be an important escape from the stresses of Ph.D. life). Importantly, I would like to thank Professor Larry Toothaker who, during my first semester in the Ph.D. program, showed me that faith and science can coexist. His example will stick with me for the rest of my life.

Finally, I would like to thank my committee who invested a considerable amount of time in me and showed a great deal of patience as we moved this dissertation to completion. I thank Bob Zmud for teaching me that being a researcher is more than just doing research; that it is a lifestyle and should be enjoyed. I thank Shaila Miranda for acting as my advocate in the Ph.D. program and for demonstrating the process of developing a research idea. Both Bob and Shaila played a critical role in defining and shaping the ideas found in this dissertation, and I am eternally thankful for their efforts. I was also fortunate to have Laku Chidambaram, Rajeev Sharma and Mark Sharfman available to offer insights that helped improve the quality of this dissertation. Leon Price has helped me many times since graduation from the MIS undergraduate program in 1991, and played an essential role in this dissertation by helping me make contacts within organizations. To my committee, I offer my sincerest gratitude.

Table of Contents

List of Tables

List of Figures

Chapter I: Introduction

“No scene from prehistory is quite so vivid as that of the mortal struggles of great beasts in the tar pits. In the mind’s eye one sees dinosaurs, mammoths, and sabertoothed tigers struggling against the grip of the tar. The fiercer the struggle, the more entangling the tar, and no beast is so strong or so skillful but that he ultimately sinks. Large-system programming has over the past decade been such a tar pit, and many great and powerful beasts have thrashed violently in it. Most have emerged with running systems — few have met goals, schedules, and budgets. Large and small, massive or wiry, team after team has become entangled in the tar.” (Brooks 1995, p. 4)

Software development (SD) projects have long been characterized as prone to result in products that are over-budget, over-schedule, feature poor, or bug rich, all of which can be very costly to an organization (Davenport 2005). For example, a 2003 report from the Standish Group suggested that only 34% of SD projects are completed within pre-defined time and budget specifications. Further, a 2002 report from the U.S. National Institute of Standards and Technology (NIST) suggested that bugs from software cost the U.S. economy almost $60 billion annually (2002). While these numbers can certainly be debated, they regardless illustrate that SD success is difficult to achieve in a consistent fashion.

To address the “tar” that can entrap those undertaking software development efforts, authors have suggested the use of formalized project management approaches, such as those found in various software development methodologies, to increase the likelihood of project success (Hartman and Ashrafi 2002; McCarthy 1995; McConnell

1998; Royce 1998). Indeed, literature has suggested that SD failure rates could be reduced if developers were provided a more structured project environment, such as that enabled through the implementation of structured management or development methodologies (Paulk, Weber, Curtis and Chrissis 1995). Methodologies embody philosophy (i.e., addressing the 'what') and technique (i.e., addressing the 'how') (Checkland 1981), both of which are essential in controlling the software development process. Project management (PM) practices are a cogent example of techniques important to the software development process, as discussed in the Project Management Institute's *A Guide to the Project Management Body of Knowledge* (PMBOK), third edition (PMI 2003). Software development project management (SDPM) practices refer to the utilization of PM techniques within a software development context, such as those discussed in waterfall, spiral, agile or extreme development methodologies. For example, development methodologies often utilize structured requirements meetings throughout the development lifecycle as a means of aligning customer requirements and technical design (DeGrace and Stahl 1990).

Research has recently begun to address the factors that influence software developer use of these types of managerial interventions (Hardgrave, Davis and Riemenschneider 2003; Khalifa and Verner 2000), but questions still remain. First, how do an individual's disparate value perceptions interact to influence the use of SDPM practices? Literature has stressed the importance of considering both instrumental and political aspects underlying individual and organizational behavior (Dean and Sharfman 1993; Robey and Markus 1984), but usage models to this point have focused on individual usefulness perceptions that are instrumental in nature. Second, what kind of SDPM practice use can be expected? Adoption literature has traditionally focused on extent of use measures (i.e. number of times used and/or number of features used), but has failed to address the nature underlying that use. Beyond the question of whether a SDPM practice is used is the more important issue of ***how*** the practice is used in terms of (1) faithfulness to organizational expectations and (2) adaptation to the task being completed. Finally, what role does the institutional environment in which a developer works play within the SDPM practice usage decision? Software development environments are dynamic and often subject to malleable constraints, and as such a model of SDPM practice usage must address the role these pressures play in a developer's usage decision.

1.1. Research Questions

The overriding objective of this dissertation can be understood through the following research question: ***What are the primary factors***

that influence a software developer's customized and faithful use of a SDPM practice within a SD project? Before proceeding, it is important to realize the boundary conditions for this question. First, the focus of this study is on software developers that operate in an organizational context. As such, any model used to address the question of SDPM practice use must acknowledge the impact of organizational opportunities and constraints that arise through association with the organization. Second, this dissertation is concerned with SDPM practice usage behavior that is exhibited within a specific development project rather than with regards to general usage behavior for all projects. This distinction is important since it focuses attention on behavioral episodes rather than behavioral tendencies. It is expected that a clearer understanding of SDPM practice usage behavior will be gained by considering software developer perceptions and behaviors within a specific project. Finally, it is important to acknowledge that this dissertation does not propose "best practices" by asserting which SDPM practices are most effective. While this is an important issue, the research at hand is rather focused on the primary factors that influence usage of SDPM practices.

To address the stated research question and in turn resolve these outstanding questions, reasoned action models of individual behavior and institutional theory will be used to generate a model that seeks to explain factors underlying a software developer's faithful and customized use of a SDPM practice.

1.2. Research Contributions

There are at least four contributions that result from this dissertation research. First, the proposed research model utilizes an integrative application of institutional perspectives in order to understand SDPM practice usage behavior. While authors researching individual behavior have addressed institutional pressure separately through concepts such as social norms and management support (Agarwal 2000; Venkatesh, Morris, Davis and Davis 2003), this research evaluates institutional pressure by considering organizational and social sources of pressure in tandem. Specifically, a classification scheme using both sources is proposed to identify the institutional environment within which a software developer works. This framework provides a novel perspective that allows researchers to consider the institutional environment as a whole when investigating individual behavior within an organization.

Second, this research is expected to extend existing adoption studies by considering both instrumental and political purposes underlying SDPM practice usage. Previous developer-focused adoption studies have tended to emphasize instrumental value judgments in determining behavioral choices (Hardgrave et al. 2003;

Khalifa and Verner 2000), but political value has long been understood as important behavioral influencer in a software development context (Robey and Markus 1984). Existing adoption models will be extended by proposing that both political and instrumental relative advantage perceptions play an important role in determining a software developer's use of a SDPM practice.

Third, the proposed model provides a conceptualization of usage behavior which is richer than pure frequency-based measures typical in many adoption studies (Khalifa and Verner 2000). Specifically, this study looks at how use is enacted by the software developer through both faithfulness and customization dimensions. Evaluating how the SDPM practice is appropriated within the project context is of utmost concern since the effectiveness of a practice is a function of how it is used within the development process. It is believed that this conceptualization will also be valuable to other adoption contexts where the nature of usage is important to implementation success.

Finally, this study will provide practitioners with clearer picture of ways in which effective SDPM practice use can be encouraged within a software development environment. Practices are often employed to improve a development group's ability to deliver software projects within time, resource and cost specifications, and as such it benefits the organization when each software developer appropriately utilizes the SDPM practices within their specific work context. The proposed research model provides a framework whereby managers can better understand the factors that can be adjusted to positively influence desired SDPM practice usage behavior.

1.3. Organization of Dissertation

This dissertation has been a living work since its inception in the spring of 2003. During this time, the dissertation has naturally evolved for both theoretical and pragmatic reasons. As such, the dissertation is divided into two main sections; the first which was derived prior to the start of data collection and the second that flowed from constraints encountered during data collection.

1.3.1. Dissertation Section 1: Original Research Model

Chapter II provides a review of literature which informs the discussion of software developer use of a SDPM practice. First, a discussion of contemporary research on software developer behavior is presented and discussed in terms of what it does and does not tell us regarding SDPM practice use. Reasoned action models of individual behavior, decision-making perspectives, and institutional theory are then presented as a means of understanding SDPM practice usage behavior exhibited by a software developer. The resulting discussion provides a framework from which the proposed research model can be understood. ***Chapter III*** draws on this foundation in order to present the research model and hypotheses. Constructs and resulting relationships are presented in relation to the theory detailed in Chapter II. ***Chapter IV*** begins with an overview of the data collection methods, data sources, and data analysis methods expected for this dissertation.

1.3.2. Dissertation Section 2: Revised Research Model

Chapter V begins by discussing a rationale for modifying the research model and follows with a detailed presentation of the research hypotheses. **Chapter VI** then continues by analyzing the data collected and presenting results. Finally, **Chapter VII** concludes by discussing findings, suggesting key implications for both theory and practice, and proposing directions for future research.

Chapter II: Literature Review

In order to generate a model that explains software developer use of a SDPM practice, we must first look to existing literature to provide a proper perspective. This chapter will begin by discussing previous findings that relate to studies of SDPM practice usage. Next, three different theoretical perspectives will be outlined to generate the structural framework for addressing the proposed research question. Finally, the conceptualization of actual use will be discussed, including a presentation of exigencies which might serve to reduce usage behavior.

2.1. Previous Findings

Several authors have addressed the adoption of innovations within a software development context, lending an important perspective to the question of SDPM practice use among software developers. One study in particular considered factors that encouraged COBOL programmers to develop favorable usage intentions regarding the C programming language (Agarwal and Prasad 2000). In the non-mandatory environment (as perceived by the responding software developers) examined by the authors, findings suggested that task-related perceptions of relative advantage were important in determining individual attitudes. Further, the radical shift in development processes dictated that compatibility issues were important when deriving positive attitudes. Intentions to use the C programming environment were then found to be positively influenced by attitudes and ease of use perceptions. This study confirms the importance of reasoned action models of individual action when considering software developer adoption of innovations. However, it is interesting to note that the authors found social factors within the software developers' environment to not be a significant predictor of usage intentions. While a cursory reading of the sample suggests that social factors should be important (i.e., all were members of one development organization), the context suggests why the social environment did not play a more significant role in the adoption decision. As stated by the authors, "the result also suggests that respondents did not believe that the organization in the study was attempting to mandate use of the C language" (Agarwal and Prasad 2000, p. 303). When adoption of an innovation is highly voluntary, social pressure has been shown to have very little effect on the adoption decision (Hartwick and Barki 1994).

A second set of authors (Khalifa and Verner 2000) used Triandis' model of human behavior (Triandis 1980) as a framework from which to understand a software developer's usage behavior for two specific software development methodologies, waterfall and prototyping. Using a cross-sectional, questionnaire-based data collection approach, the authors found that use of either software development methodology was significantly influenced by facilitating conditions (i.e., team size, departmental innovativeness, and organizational support) and developer beliefs regarding the methodology's impact on the quality of the software development process. Use in this study was discussed through two dimensions, where *depth* referred to the extent the methodology was used in each phase of the development process and *breadth* referred to the variety of applications that were developed with each approach. This research also lends credibility to reasoned action models of individual behavior by (1) suggesting that beliefs are an important predictor of software developer usage behavior and (2) acknowledging the importance of facilitating conditions, or "objective factors in the environment that facilitate performance of an act" (Khalifa and Verner 2000, p. 362), when investigating usage behavior within an less-than-voluntary context (Ajzen 1991).

Finally, a third set of authors (Hardgrave et al. 2003; Riemenschneider, Hardgrave and Davis 2002) have recently evaluated several competing models commonly accepted within technology adoption literature to determine their appropriateness in a software developer methodology adoption context. The first study (Riemenschneider et al. 2002) tested the Technology Acceptance model (TAM) (Davis 1989; Davis, Bagozzi and Warshaw 1989), TAM2 (Venkatesh and Davis 2000), Perceived Characteristics of Innovating (PCI) (Moore and Benbasat 1991), the Theory of Planned Behavior (TPB) (Ajzen 1991) and the Model of Personal Computer Utilization (Thompson, Higgins and Howell 1991) and found that reasoned action models of individual behavior could be applied in a methodology adoption context. Specifically, usefulness (a.k.a. relative advantage, attitude, job fit), subjective norm (a.k.a. social factors) and compatibility were shown to positively impact developer intentions to use the methodology while voluntariness was shown to negatively impact intention formation. Their results also demonstrated that ease of use perceptions and perceived behavior control (internal and external) did not significantly impact intention formation. While the authors suggest that "as the behavioral domain changes from tool use to methodology use, there is a reduction in the relevance of how easy or hard the behavior is to perform and whether or not one possesses adequate internal or external resources to perform it" (Riemenschneider et al. 2002, p. 1141), this finding might be an artifact of the data collection approach. Specifically, data collection for the study occurred post-adoption, creating the possibility that perceptions of complexity had been reduced through experience with the development methodology. In a similar way, the importance of perceived behavior control (i.e., the individual’s perceived ability to complete the behavior) by definition will be impacted after a software developer has already used the methodology.

The next study by these authors combined TAM (Davis et al. 1989; Venkatesh and Davis 2000) with diffusion of innovations theory (Rogers 1995) to propose a model explaining software developer adoption of a software development methodology. Using a cross-sectional, questionnaire-based data collection approach, Hardgrave et al. (2003) found that perceived usefulness and compatibility all demonstrated a significant relationship with the individual’s behavioral intention to use the software development methodology. When compared to the previously discussed studies, these authors employ a more comprehensive array of usage drivers within reasoned action models and find that social pressure and organizational mandate are indeed important predictors of methodology adoption intentions. Perceptions of complexity were shown to not exert a direct impact on usage intentions; an interesting finding given that the methodology was suggested to be a ‘radical’ change. However, this finding might be explained by the wealth of development experience within the respondent group (average of 10 years development experience), calling into question how radical the new methodology might have truly been (e.g., the methodology did not utilize a CASE tool).

These studies offer several important insights within the current research context. First, they provide validation that understanding factors that influence a software developer's adoption of development techniques and methodologies are an important, yet rarely discussed phenomena within the IT discipline. Second, each illustrates the relevance of utilizing reasoned action models of individual behavior when trying to understand software developer usage of SDPM practices. This is important since it provides a perspective regarding constructs expected to be most applicable within the SDPM practice usage process. Finally, the later studies discussed above acknowledge that the social context in which software developers operate must be considered if we are to understand enacted behavior. Social context is especially important when considering the reasoning behind software developer behavior (Wastell 1999), and a model of SDPM practice usage must account for this influence.

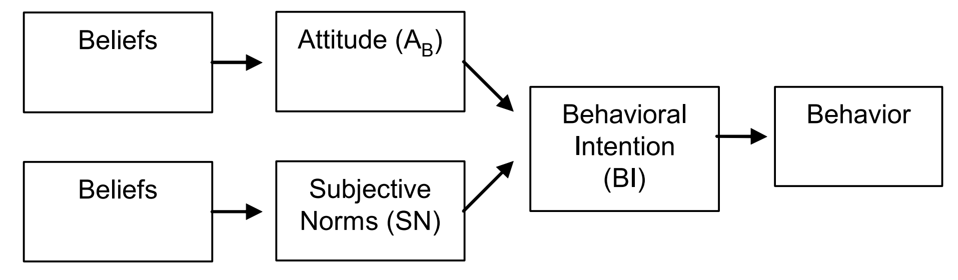
However, there are still several areas which must be addressed if we are to comprehensively understand a software developer's usage of SDPM practices. First, software developers in an organizational context are subject to a complex set of criteria when determining the value of organizational practices (Robey and Markus 1984), but usage models to this point have focused on individual usefulness perceptions that are task-focused only. Second, models to this point have focused on extent-based usage (or intention) measures. Beyond the question of whether a SDPM practice (will be / is being) used is the equally important issue of ***how*** the practice (will be / is being) used by software developers. Finally, the previously discussed models do not address the impact that external pressures might exert on usage behavior. The following sections will discuss three theoretical perspectives that are expected to address these questions in order to provide a robust model of factors that influence software developer SDPM practice usage.

2.2. Reasoned Action Models of Individual Behavior

When deriving the factors that influence a software developer's SDPM practice usage, it is important to begin with existing models that seek to explain individual behavior. Reasoned action models, such as those presented in technology adoption research, provide a theoretical lens from which an individual's usage decision process can be examined (Fishbein and Ajzen 1975; Venkatesh et al. 2003). A number of reasoned action models have received support across a variety of domains (Agarwal 2000; O'Keefe 2002), suggesting that they might provide a powerful lens for this research context. The following paragraphs outline several of the most accepted models of reasoned action which can elucidate factors influencing SDPM practice use.

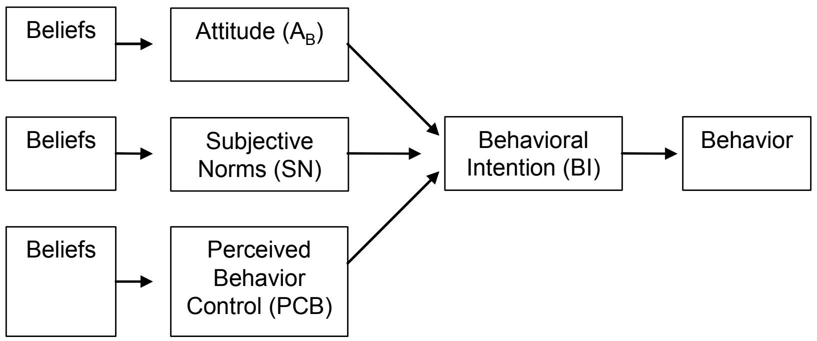
Reasoned action models of individual behavior find their roots in TRA, or the Theory of Reasoned Action, as proposed by Fishbein & Ajzen (1975). TRA, in considering volitional behavior, suggested that an individual's disposition towards a behavior (attitude) combined with their perception that referent others think they should exhibit the behavior (subjective norms) influence the likelihood of developing an intention to exhibit the behavior. Expressed mathematically, TRA proposes that $\mathbf{BI = A_B(w_1) + SN(w_2)}$, where BI is an individual's behavioral intention that is influenced by both an individual's attitude towards the behavior (A_B) and subjective norms with regard to the behavior (SN). Both A_B and SN are suggested to contribute to BI differentially, as represented by the different weightings (w_1 & w_2) assigned to the factors (O'Keefe 2002). The theory also suggests that attitude and subjective norm generation are derived from individual beliefs (Mathieson 1991). Research from multiple domains such as family planning, paper recycling, exercise, consumer purchases, and technology use have confirmed that TRA provides a powerful perspective for understanding individual behavior in voluntary environments (Agarwal 2000; O'Keefe 2002). The primary relationships proposed in TRA are graphically illustrated in Figure 2.2-1.

Figure 2.2-1: Theory of Reasoned Action



Building on TRA, authors subsequently proposed that an additional construct, perceived behavior control, was important in situations where the behavior in question was considered less than voluntary. The Theory of Planned Behavior (TPB) specifically suggested that “the presence or absence of requisite resources and opportunities” (Ajzen and Madden 1986, p. 457) played an important role in determining whether an individual would exhibit a given behavior. Thus, TPB built on TRA by proposing $\mathbf{BI = A_B(w_1) + SN(w_2) + PCB(w_3)}$, where PCB represents perceived behavior control and w_3 represents its weighting. Similar to attitudes and subjective norms, perceived behavior control was suggested to be largely shaped by individual beliefs. According to O’Keefe, “the number and diversity of supportive findings suggest that the TPB will often provide a superior model (when compared to TRA)” (2002, p. 116). The relationships proposed in TPB are pictured in Figure 2.2-2.

Figure 2.2-2: Theory of Planned Behavior



Another reasoned action perspective for individual adoption behaviors can be found in Diffusions of Innovations (DOI) literature (Rogers 1995). This perspective suggests that adoption of an innovation is strongly influenced by how an individual answers the following questions (Moore and Benbasat 1991): (a) how valuable is the innovation compared to what it will replace (relative advantage), (b) how well does the innovation fit with existing norms or past experiences (compatibility), (c) how difficult is the innovation to use (complexity), (d) how difficult is it to observe the outcomes of using the innovation (observability), and (e) to what degree can the individual experiment with the innovation before adoption (trialability). Expressed mathematically, a DOI perspective proposes that **ADOPT = RA(w_1) +**

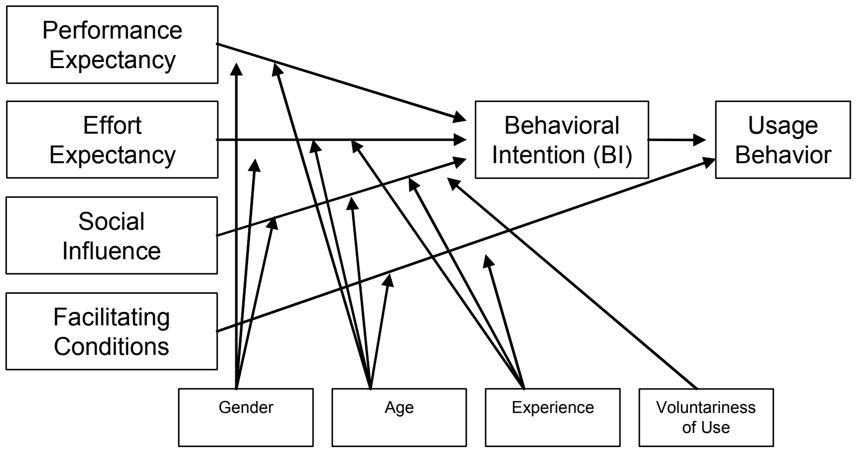
COMPAT$(\mathbf{w}_2)$ + **COMPLX**$(\mathbf{w}_3)$ + **OBSR**$(\mathbf{w}_4)$ + **TRIAL**$(\mathbf{w}_5)$ [1], a model which has been derived and validated through the comprehensive evaluation of hundreds of diverse innovation studies (Rogers 1995).

Researchers of technology adoption have extensively drawn on the reasoned action models discussed to this point. For instance, the widely utilized Technology Acceptance Model (TAM) and its predecessor TAM2 draw on these models to propose that salient beliefs regarding the technology (perceived ease of use and perceived usefulness) influence attitudes which work together with social norms to impact an individuals intention to adopt and eventual adoption of the technology (Davis 1989; Davis et al. 1989; Venkatesh and Davis 2000). Further, authors have also drawn on DOI literature to propose the Perceived Characteristics of Innovating (PCI) model to explain technology adoption behaviors (Moore and Benbasat 1991). Understanding that each of these technology adoption models possesses areas of overlap, authors have recently attempted to combine them into one unified model that can comprehensively explain technology adoption behavior. The resulting model is called the Unified Theory of Acceptance and Use of a Technology (UTAUT) and is displayed in Figure 2.2-3 (Venkatesh et al. 2003). UTAUT proposes that behavioral intentions are influenced by performance expectancy

[1] **ADOPT** = individual adoption of the innovation, **RA** = relative advantage, **COMPAT** = compatibility, **COMPLX** = complexity, **OBSR** = observability, **TRIAL** = trialability, and $\mathbf{w}_{\mathbf{1\text{-}5}}$ = the weighting of each antecedent condition

(i.e., relative advantage), effort expectancy (i.e., perceived complexity), and social influence (i.e., social norms) expressed mathematically as **BI = PE(w_1) + EE(w_2) + SI(w_3)** [2]. Usage behavior is then suggested to be influenced by behavioral intentions and facilitating conditions (i.e., perceived behavior control and compatibility), expressed mathematically as **USE = BI(w_4) + FC(w_5)** [3]. Moderators such as gender, age, experience and voluntariness of use are also proposed to impact the influence of constructs on usage intentions and actual use. Validation of UTAUT in a technology adoption context is still in the early stages, but current findings are encouraging (Venkatesh et al. 2003).

Figure 2.2-3: Unified Theory of Acceptance and Use of Technology



[2] **PE** = Performance Expectancy, **EE** = Effort Expectancy, **SI** = Social Influence, $\mathbf{w_{1-3}}$ = weightings assigned to each antecedent condition
[3] **USE** = Use Behavior, **FC** = Facilitating Conditions, $\mathbf{w_{4-5}}$ = weightings assigned to each antecedent condition

An important question that must be addressed is how applicable these different models are when trying to understand a software developer's use of a SDPM practice. First, the appropriateness of reasoned action models in understanding software developer adoption behavior must be confirmed. From a technology adoption perspective, adoption research has often focused on end users and not "experts" within the information technology domain. Several studies have successfully applied reasoned action models to understand software developer behavior, such as studies looking at the adoption of the C language (Agarwal and Prasad 2000) and software development methodologies (Hardgrave et al. 2003; Khalifa and Verner 2000). Thus, reasoned action models of individual behavior appear to be relevant perspectives from which to understand software developer behavior.

Second, the question of applicability must be evaluated in terms of the target innovation being considered. While many studies have focused on the adoption of tools, or an artifact that can be used for accomplishing some ends, this study focuses on the adoption of practices. In their consideration of the applicability of five different reasoned action models to a software development methodology adoption context (Riemenschneider et al. 2002), authors found that exhibited relationships tended to be consistent with those found when the adoption artifact of interest is a tool. This suggests that reasoned action models are an appropriate perspective for considering factors that influence a software developer's use of SDPM practices.

Thus, findings from the wide array of reasoned action studies offer several important insights into the factors that might influence a software developer's use of a SDPM practice. Specifically, the models discussed above suggest that the following perceptual factors should be considered when investigating software developer use of SDPM practices:

- SDPM practice complexity
- SDPM practice relative advantage
- SDPM practice social norms
- Facilitating conditions
- SDPM practice usage intentions
- SDPM practice voluntariness

While these constructs provide important guidance when examining individual behavior, several limitations must be addressed. The first limitation lies in an undivided focus on instrumental value judgments as a precursor to usage behavior. While instrumental relative advantage has been demonstrated to be important in the previously discussed models, software developer perceptions regarding value of the SDPM practice will be formed in an organizational setting where developer worth is evaluated in more ways than just in relation to task performance. As such, the developer will be subject to value judgments that are more complex than those solely focused on the task-related benefits of using the SDPM practice. Another key limitation of reasoned action models can be found in a fragmented treatment of institutional pressures on individual behavior. Institutional pressures are often discussed through social norms or facilitating condition constructs, but rarely presented in a comprehensive manner with regards to the pressured inherent in the institutional environment. Finally, reasoned action models tend to focus on extent-based measures of use / adoption and do not get into the critical issue relating to how usage behavior is exhibited. The following sections will apply a decision-making perspective and institutional theory to build on the value offered through reasoned action models of individual behavior.

2.3. Decision-Making Perspectives

Relative advantage, or perceived usefulness, is most often addressed in adoption research with regards to the innovation's instrumental value, defined by its ability to improve task-related job performance (see the Performance Expectancy construct discussion in Venkatesh et al. 2003 for a detailed treatment). However, the value an organizational worker assigns to a behavior has often been suggested to result from more than purely instrumental criteria. For example, researchers focusing on the adoption of IT have suggested that value judgments are realized in both short- and long-term manifestations, where short-term represents the value of a behavior with respect to the task at hand while long-term represents the value of a behavior with regard to professional-related benefits such as increased compensation and improved image (Compeau and Higgins 1995; Thompson et al. 1991). Further, non-instrumental relative advantage perceptions have been addressed in adoption literature through the image construct proposed in the PCI model (Moore and Benbasat 1991) where individuals are suggested to be more prone to adopt an innovation when they perceive it to help them improve their social standing. The context of an individual's environment often dictates the types of usefulness perceptions that are most salient when considering adoption behaviors (Dennis and Reinicke 2004), and evaluating software developer usage of a SDPM practice within an organizational context requires us to consider which sources of relative advantage will be most pertinent for this study.

Researchers looking into the decision-making process of management executives offer an insightful perspective when considering decision-making in an organizational context. Of particular interest is the suggestion that managers are subject to rational (Fredrickson and Iaquinto 1989) and political (Eisenhardt and Bourgeois 1988) considerations when deriving solutions to organizational problems. Procedural rationality, or “the extent to which the decision process involves the collection of information relevant to the decision and reliance upon analysis of this information in making the choice” (Dean and Sharfman 1993, p. 1071), is suggested to be an important aspect of the decision-making process when objectives are known and managers are able to cognitively evaluate the value of derived alternatives (Eisenhardt and Zbaracki 1992). Political behavior, or “the observable, but often covert, actions by which executives enhance their power to influence a decision” (Eisenhardt and Bourgeois 1988, p. 738), is also proposed to be important within the decision-making process when the manager is operating in a context where coalitions of people and competing interests exist (Eisenhardt and Zbaracki 1992). Dean & Sharfman (1993; 1996) suggest that procedural rationality and political behavior are distinct, yet interrelated factors that serve to influence strategic decision-making. Software developers are often situated in development projects where (a) an overall objective is understood such that alternative choices of action are able to be evaluated and (b) competing interests exist between members of the development organization such that choices must be evaluated in light of their ability to protect the self-interests of the software developer. This suggests that political and rational dimensions will also be important in a software developer context, a suggestion which is validated in several software developer-focused studies.

For example, Robey & Markus, in their discussion of organizational decisions regarding IS design, suggest that “IS design can fruitfully be explained as both a rational and a political process” (Robey and Markus 1984, p. 12). The authors explain that systems development can be viewed as a rational process in that it strives to create systems that (a) increase organizational effectiveness and (b) are adopted and used as prescribed. In addition, systems development can also be viewed as a political process where potential exists for the various participants, with multiple competing objectives, to gain or lose power within the design process. In other words, systems development choices within an organization are a function of the instrumental and political utility attributed to each individual decision. While these two dimensions are divergent, it is important to recognize that they are also interdependent as evidenced in the following quote (Robey and Markus 1984, p. 12):

> *“Rituals in systems development function to maintain the appearance of rationality in systems development and in organizational decision making. Regardless of whether it actually produces rational outcomes or not, systems development must* ***symbolize*** *rationality and* ***signify*** *that the actions taken are not arbitrary, but rather acceptable within the organization’s ideology. As such, rituals help provide meaning to the actions taken within an organization.”*

A more recent treatment of the rational / political view of value judgment has been provided in literature on software development quality. Software developers operating in an organizational environment have been suggested to make decisions regarding the use of shortcut-taking behaviors based on two concerns: concern for quality and concern for career (Austin 2001). Within this perspective, concern for quality addresses the value that a software developer places on project success, while concern for career addresses a fear that the developer "may 'look bad' in the eyes of the principal if he confesses that he is behind schedule and his fellow agent does not" (Austin 2001, p. 197). Shortcut-taking behavior, then, is suggested to be a function of both a software developer's concern for quality and concern for career. Indeed, this perspective suggests that more than task-related perceptions of relative advantage are important when evaluating software developer behavior.

Drawing on this set of literature provides a perspective regarding the types of relative advantage which will be salient in a software development context. Specifically, the preceding perspectives suggest that value perceptions within an organizational context can be viewed along ***instrumental*** and ***political*** dimensions, where instrumental addresses the value of a SDPM practice for completing project-related tasks and political refers to the value of a SDPM practice with regards to how it improves the individual's image within the organization.

2.4. Institutional Theory

As discussed earlier, another gap in reasoned action models of individual behavior lies in an anemic treatment of the institutional environment within which the individual operates (Orlikowski and Barley 2001). Two aspects of this theory can inform an investigation of factors that influence a software developer's use of SDPM practices. First, institutional theory provides a perspective for understanding the primary sources of institutional pressure within the domain of individual adoption behavior. Second, the theory also provides a means of predicting individual responses to imposed institutional pressure. The section below will outline these two issues in relation to a software developer's use of SDPM practices.

At its core, institutional theory "emphasizes that organizations are open systems – strongly influenced by their environments – but that it is not only competitive and efficiency-based forces that are at work. Socially constructed belief and rule systems exercise enormous control over organizations – both how they are structured and how they carry out their work" (Scott 2003, p. 119-120). Institutional theory provides a perspective for understanding the role that institutional forces play in usage decisions by suggesting that (1) organizational choice is constrained by external pressures, (2) organizational survival depends on responsiveness to external demands, (3) organizations are legitimacy seekers, and (4) organizations are interest driven (Scott 2003; Selznick 1996). Applied at an individual level, institutional theory suggests that “the behaviors of individuals within organizations are significantly influenced by the prevailing organizational norms, values, culture, and history” (Purvis, Sambamurthy and Zmud 2001, p. 120).

Institutions are suggested to be “composed of cultural-cognitive, normative, and regulative elements that, together with associated activities and resources, provide stability and meaning to social life” (Scott 2001, p. 48). The institutional environment, then, can be seen as exerting its influence on individuals through each of these three elements. ***Cultural-cognitive*** elements within the institutional environment arise through the shared meaning that exists within any environment, such as when a managerial practice is viewed as part of the standard operating procedure within an organization. ***Normative*** elements are demonstrated through a moral framework (i.e., perception of behavior that is considered right or wrong) underlying the institutional environment, evidenced by individual perceptions of referent others’ expectations. Finally, ***regulative*** elements are illustrated through the formal rules and governance structures created within the institutional environment, as often expressed within documented organizational policies and procedures.

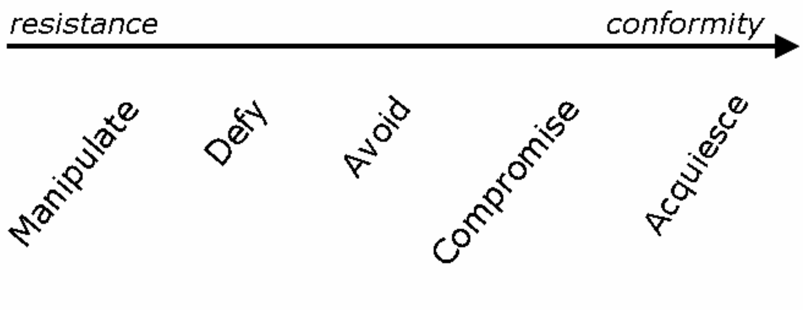
These three elements within the institutional environment have been variously addressed in IT adoption literature. While early models of technology adoption ignored the potential impacts of the institutional environment (Davis 1989), later models have embraced normative and/or cultural-cognitive factors. For example, an institutional environment’s normative influence on individual adoption of a technology is often modeled through social influence (a.k.a., subjective norms and social factors), or “the degree to which an individual perceives that important others believe he or she should use the new system” (Venkatesh et al. 2003, p. 451). Additionally, cultural-cognitive influence within the institutional environment is often suggested to impact adoption behaviors through the presence of facilitating conditions within the environment, evidenced through managerial and/or organizational support perceptions (Purvis et al. 2001; Thompson et al. 1991; Venkatesh et al. 2003). Addressing normative (e.g., social norms) and cultural cognitive (e.g., facilitating conditions) aspects of the institutional environment has also been the norm within research looking at the adoption of methodologies by software developers (Hardgrave et al. 2003; Khalifa and Verner 2000).

Two things are of note when considering how institutional factors have been addressed in adoption-focused research. First, the regulative environment is often ignored with regards to its impact on an individual's adoption decision. Second, it is the exception rather than rule to find more than one of the institutional elements represented in models of technology adoption. As suggested by Scott (2003, p. 135), "in any fully developed institutional system, all three of these forces or elements are present and interact to promote and sustain orderly behavior". Thus, when evaluating the use of a SDPM practice within a software development context, it is important to consider all three sources of influence within the institutional environment. These three forces are evidenced through (a) the degree of SDPM practice routinization within the development organization (cultural-cognitive), (b) the prevalence of social norms that exist within the development organization (normative), and (c) the degree that the SDPM practice is formally documented within the organization's project management methodology (regulative).

In addition to providing a framework for understanding the sources of institutional pressure, institutional theory provides a perspective for understanding how organizations respond to institutional pressures which in turn can inform how a software developer might view / use SDPM practices in light of those pressures. First, an institutional perspective proposes that organizations have an overwhelming need to conform to the institutional environment in order to enhance the likelihood of survival (DiMaggio and Powell 1983; Meyer and Rowan 1977). Applied at an individual level, this suggests that members of an organization might exhibit behaviors that are perceived as enabling conformity with the institutional environment in order to enhance their legitimacy within that organization. Behaviors, then, can be viewed as partly driven by an organizational worker's valuation of the behavior in terms of its symbolic merits. In addition to the previously discussed instrumental and political dimensions of relative advantage, perceptions of the symbolic relative advantage associated with a behavior might also play an important role in shaping an individual's behavioral choices.

Second, institutional theory also suggests that conformity to institutional pressures can be varied based on the level of active agency expressed by an organization or individual (Goodrick and Salancik 1996). For example, research has shown that organizational adoption of work-family programs (i.e., child care and flexible workplace initiatives) vary in their conformity based on institutional factors such as dependence on the institutional environment (Goodstein 1994). As illustrated in Figure 2.4-1, organizational responses to institutional pressures are suggested to range from absolute conformity (acquiescence) to all-out resistance (manipulation) and are the result of institutional factors such as the cause of institutional pressure (legitimacy and/or efficiency rationale), the constituents exerting the institutional pressure (multiplicity and/or dependence rationale), the content expressed through the institutional pressure (consistency of pressures with organizational goals and/or constraints resulting from the pressure), organizational level of control regarding the imposed pressure (regulatory coercion and/or voluntariness of diffusion), and the context underlying the institutional pressure (environmental uncertainty and/or interconnectedness) (Oliver 1991). Applied at an individual level, this suggests that a software developer will conform their use of SDPM practices in different ways depending on pressures within the institutional environment.

Figure 2.4-1: Oliver's (1991) Strategic Responses to Institutional Processes



2.5. Conceptualization of Usage

Reasoned action models of individual behavior typically evaluate the manifestation of a specific behavior (i.e., voting, consumer purchases, exercise) as their ultimate dependent variable (O'Keefe 2002). In the case of IS adoption literature, this often means that researchers focus on usage behavior with regards to technology innovations such as personal computers (Compeau and Higgins 1995), CASE tools (Purvis et al. 2001), programming languages (Agarwal and Prasad 2000), or software methodologies (Hardgrave et al. 2003; Khalifa and Verner 2000). While usage has often been acknowledged as a very complex process, IT researchers have tended to focus on extent-based measures. This typically results in a view of use which is frequency-based (number of times used) (Davis 1989) and/or focused on depth of use (number of features used) (Igbaria, Zinatelli, Cragg and Cavaye 1997). While these measures account for the outward manifestations of usage behavior, they fail to address the nature that characterizes that usage. A need to address more rich adoption measures has been voiced in other domains, such as that within organizational adoption research. For example, TQM adoption research has stressed that “for administrative innovations, the appropriate question may not only be whether organizations adopt but how they adopt” (Westphal, Gulati and Shortell 1997, p. 370) and organizational practice adoption research has suggested adoption behavior consists of implementation (degree to which external or objective behaviors required by the practice are exhibited) and internalization (degree to which employees view the practice as valuable and express commitment to it) components (Kostova and Roth 2002). In a similar manner, the way in which a practice is used by software developers is just as important as if it is used and as such must be addressed within this research. The remaining question, then, relates to the types of usage that are salient when considering how SDPM practice usage is enacted.

The previous discussion of institutional theory demonstrates how responses to the institutional environment are often discussed along a continuum represented with conformity to resistance anchors. Desanctis & Poole (1994) addressed a related issue when discussing individual and group appropriation of a technology. In their theorizing, the authors suggested that one aspect of appropriation relates to faithfulness (i.e., conformity) in relation to the spirit and structural feature set within a given technology. Faithful appropriation with regards to a technology's spirit is of special concern here since it addresses an important aspect of the nature underlying usage behavior. Specifically, it is important to consider the degree to which the SDPM practice is used in the same manner as intended by the development organization because (a) predicting software developer deviance from prescribed development practices is of utmost concern for management (DeGrace and Stahl 1990) and (b) faithful compliance in light of institutional pressures has been suggested to provide benefits such as enhancing legitimacy within the institutional environment (Staw and Epstein 2000) or improved decision quality in a GSS environment (Wheeler and Valacich 1996).

A second, and equally important, issue underlying use of the SDPM practice relates to how the practice is customized within a given project. Adaptation-focused usage speaks to an efficiency-rationale whereby the adopting entity desires to maximize value by customizing the innovation to the problem at hand (Westphal et al. 1997). The benefits of customizing behavior for a given task have been widely discussed in theories of task-technology fit, where effective GSS use is suggested to largely be a function of the fit between the task and subsequent use of a technology (Dennis, Wixom and Vandenberg 2001; Zigurs and Buckland 1998). In the context of management practices, “positive” deviation (i.e., adaptation to the given task) is often required if an individual is to generate above-normal value (Pascale and Sternin 2005). Thus, investigating software developer usage of SDPM practices should also acknowledge the degree to which the practice is customized, or adapted, within a given project.

Interestingly, Oliver's discussion of organizational responses to institutional pressures hints at adaptation aspects of usage in addition to the explicit focus on conformity (i.e., faithfulness). While responses are suggested to be a function of an organization's willingness to faithfully adopt some institutional norm, these same responses also illustrate an organization's activeness with regards to adaptation of the norms. For example, habitual acquiescence is represented by "unconscious or blind adherence to preconscious or taken-for-granted rules or values…(whereby) an organization may be unaware of institutional influences and, accordingly, precluded from responding to them strategically" (Oliver 1991, p. 152). With this type of response, not only is conformity to institutional norms high but also adaptation of the norms are low since its taken-for-granted status inhibits the organization from considering alternative modes of operation. Similarly, each of the remaining strategic responses proposed by Oliver (compromise, defy, and manipulate) can be characterized more richly along two dimensions (conformity and adaptation) rather than purely via a conformity continuum.

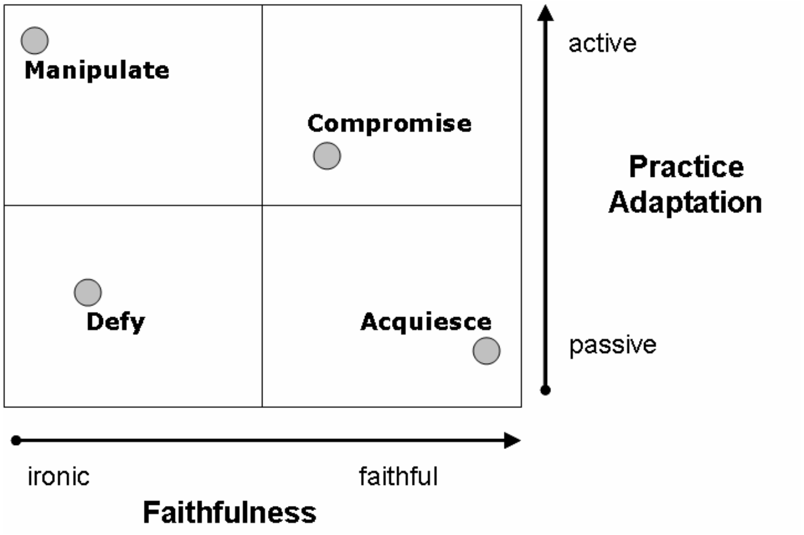
Applied at an individual level, this two dimensional view of responses to institutional pressures provides a means of understanding usage behavior, specifically when evaluating a software developer's use of SDPM practices. ***Faithfulness*** (used instead of conformity since it more closely aligns with terminology proposed in Adaptive Structuration Theory) and ***practice adaptation*** are suggested to inform the type of SDPM practice usage to be expected from a software developer. Drawing on the work of Oliver (1991), a description of the dimensions and their associated response are detailed in Table 2.5-1 and graphically illustrated in Figure 2.5-1. Important to recognize is the exclusion of an avoidance response, which focuses on concealment (providing a façade that the behavior is being followed, but not actually following it), buffering (attempts to reduce the extent to which external scrutiny occurs), and/or escape (leaving the environment entirely). The focus on this study is usage behaviors in an organizational context, and avoidance responses are representative of non-usage behavior which make them inappropriate for this research.

Table 2.5-1: Individual Responses to Institutional Pressures

Usage Response	Description	Level of Faithfulness	Level of Practice Adaptation
Acquiesce	Unqualified conformity with the norm, exhibited through habit, imitation or conscious compliance.	Very high	Very low
Compromise	Partial conformity with the norm with an active desire to meet individual requirements. Often manifested in balancing, pacifying or bargaining behaviors.	High	High

Defy	Active resistance demonstrated through contrary means such as dismissing, challenging, or attacking the norms.	Low	Low
Manipulate	Full-out resistance to the norms by attempting to “actively change or exert power over the content of the expectations themselves or the sources that seek to express or enforce them” (Oliver 1991, p. 157) in such manners as co-optation, influence and/or control.	Very low	Very high

Figure 2.5-1: Individual Responses to Institutional Pressures



2.6. Exigencies and Usage Behavior

A SD project is temporary by definition (PMI 2003), and many unanticipated events can occur during the life of the project (from inception to completion). Assuming issues are managed within carefully controlled project processes, a project manager can ensure that most unexpected changes are handled in such a way that the project can be completed within specification. However, there are often events that circumvent carefully planned project procedures and inflict an adverse result on project deliverables. These external pressures or stressors can also cause an individual to act in ways that are contrary to their original expectations.

Literature investigating individual reactions to episodes of stress provides a perspective for interpreting behavior that is inconsistent with cognitive preferences. Of particular interest within the current study is the suggestion that “psychological stress enhances the utilization of suboptimal cognitive processes and the appearance of cognitive errors and biases” (Zakay 1993, p. 60). This suggests that beliefs formed through cognitive evaluation and consideration might be a poor predictor of usage behavior when sufficient stressors exist to force suboptimal decision making. Research on the battlefield conduct of American soldiers in World War II supports this idea. Post-WWII analysis of battlefield data found that, despite significant levels of weaponry training for infantry personnel, a large proportion of soldiers were found to either not fire or purposely misfire their weapons during episodes of battlefield stress (Grossman 1995). Further support for this idea can be found in literature evaluating the impacts of time pressure on decision making behavior. In environments where time pressures are high, individual processing of information has been shown to often shift from alternative-based to attribute-based processing (Payne, Bettman and Luce 1996). Specifically, research has demonstrated that time pressure can negatively impact an individual's pursuit of competing alternatives (Payne, Bettman and Johnson 1988) and in turn increase the likelihood of focusing on one attribute deemed to be most important (Edland 1994) when required to produce a single solution.

While it would be rare for individuals undertaking software development to experience the level of stress encountered on a military battlefield, nonetheless this example combined with the research on time pressure illustrates that pressure exerted from outside an individual's control can result in a behavior that is contrary to original cognitions. In the context of a software development project, external pressures often result from unexpected changes in project priority, individual job responsibilities, financial resource availability, personnel resource availability, and/or organizational stability. Each of these changes, or operational exigencies, can act as a source of stress for the software developer and serve to alter their realized behavior.

2.7. Conclusion

The preceding discussion has provided a lens for understanding a software developer's usage of SDPM practices within an organizational development project. Reasoned action models of individual behavior illustrate that SDPM practice usage arises through cognitive evaluations of SDPM practice complexity and relative advantage. Also, environmental issues such as SDPM practice social norms and facilitating conditions are expected to be of concern in the usage decision while issues such as voluntariness of use, gender, age have the potential of altering relationships in a model of SDPM practice usage and as such must be accounted for. The decision-making literature provides an important perspective of value judgments in the usage process, suggesting that relative advantage should be viewed from both instrumental and political perspectives. Institutional theory demonstrates that (a) symbolic relative advantage perceptions and (b) the degree of SDPM practice routinization as represented through management and organizational support (i.e., cultural-cognitive), combined with SDPM practice social norms (i.e., normative) and the degree of SDPM practice codification (i.e., regulative) must be considered when evaluating SDPM practice usage behavior. Institutional theory and diffusions literature suggest that SDPM practice usage behavior should be addressed through extent-based, faithful and adaptation components in order to provide a robust view of usage behavior. Finally, operational exigencies are then suggested to provide an important explanation for variations between developer usage cognitions and SDPM practice usage behavior.

The ultimate dependent variable in this study will be software developer use of the SDPM practice, characterized along extent, faithfulness, and customization dimensions. Antecedents proposed in the research model are suggested to flow from reasoned action, institutional and operational exigency sources that work together to impact the ultimate usage decision. Drawing on these perspectives, the resulting conceptual research model is provided in Figure 2.7-1 with Table 2.7-1 containing the definition of each construct. The next chapter will build on this framework to propose a simplified model of SDPM practice usage.

Figure 2.7-1: Conceptual Research Model

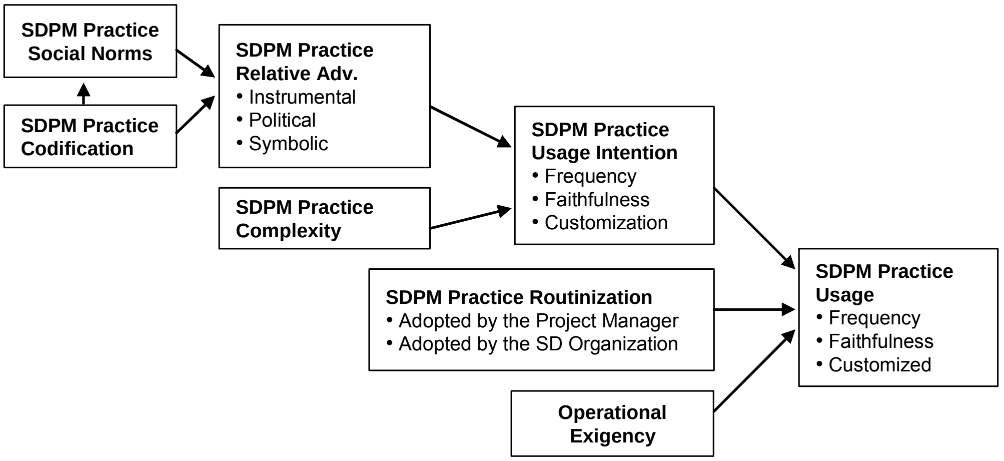

SDPM Practice Social Norms
SDPM Practice Codification
SDPM Practice Relative Adv.
• Instrumental
• Political
• Symbolic
SDPM Practice Complexity
SDPM Practice Usage Intention
• Frequency
• Faithfulness
• Customization
SDPM Practice Routinization
• Adopted by the Project Manager
• Adopted by the SD Organization
Operational Exigency
SDPM Practice Usage
• Frequency
• Faithfulness
• Customized

Table 2.7-1: Overview of Construct Definitions

SDPM practice complexity	a software developer's perception concerning the degree of difficulty associated with the understanding and use of the SDPM practice for a particular project
SDPM practice relative advantage – instrumental	the degree to which a software developer believes that using the SDPM practice will help them attain gains in performance for the software development project
SDPM practice relative advantage – political	the degree to which a software developer believes that using the SDPM practice will help them create or sustain a positive impression with others in the organization
SDPM practice relative advantage – symbolic	the degree to which a software developer believes that using the SDPM practice will help them achieve or maintain congruence with the rest of the organization
SDPM practice codification	the degree to which the SDPM practice has been documented within the organization's formal project management methodology
SDPM practice social norms	the degree to which a software developer perceives that important others (i.e., those in their workgroup) believe he or she should use SDPM practice for a given project
SDPM practice routinization	the degree to which a practice has become embedded within the software development organization, represented by the following two dimensions: • ***adopted by the project manager*** – the degree to which a project manager requires that the SDPM practice is used for a particular project • ***adopted by the SD organization*** - the degree to which the software development organization requires that the SDPM practice is used for all software development projects
SDPM practice usage intention – frequency	the degree to which a software developer intends to use the SDPM practice for a given development project
SDPM practice usage intention – faithfulness	a software developer's intention to use the SDPM practice in a manner consistent with the intentions of the development organization
SDPM practice usage intention – customization	a software developer's intention to modify the SDPM practice for a given development project
Operational exigency	the degree of urgency exerted within a given software development project by external forces

SDPM practice usage – frequency	the degree to which a software developer uses the SDPM practice for a given development project
SDPM practice usage - faithfulness	degree to which a software developer uses the SDPM practice in a manner consistent with the intentions of those individuals that created the practice
SDPM practice usage - customized	the degree to which a software developer has modified the SDPM practice for a given development project

Chapter III: Research Model & Hypotheses

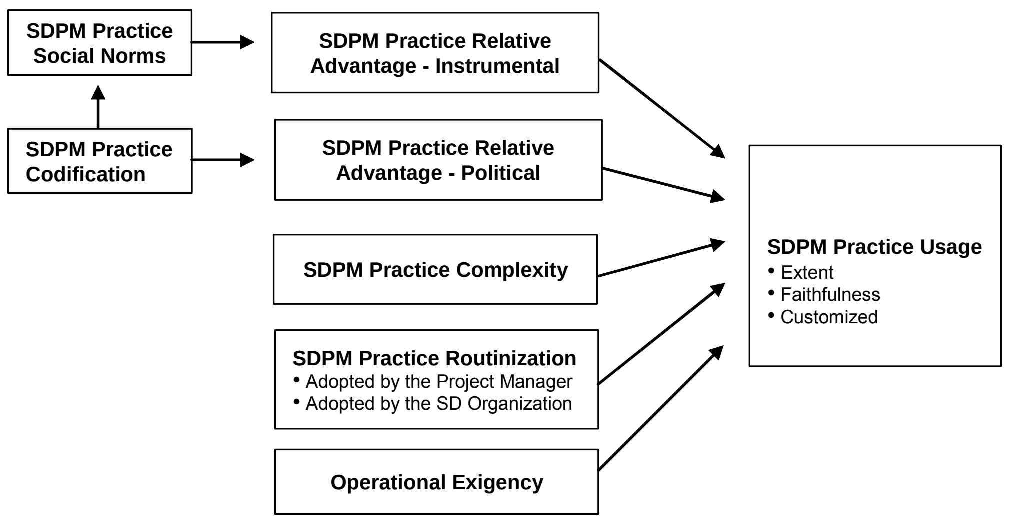
The goal of this research is to understand the factors that influence a software developer's usage of SDPM practices. The following section introduces the research model to be considered in this dissertation and then provides an impetus for the relationships proposed in the model.

3.1. Research Model

In considering the conceptual model outlined in Figure 2.7-1, it became clear that the scope had to be managed in order to ensure that the fundamental dissertation goals could be accomplished. Specifically, a decision was made to create a simplified model which would address the research question in this dissertation, but which could be built upon at a later time in order to address the overall conceptual model. The simplified model differs from the conceptual research model in two ways. First, a decision was made to remove symbolic relative advantage perceptions from the research model. A review of adoption research revealed that while instrumental and political relative advantage had been specifically addressed, no treatment of symbolic relative advantage could be found. The issue of symbolic relative advantage is important and deserves a significant amount of attention in order to properly conceptualize and operationalize it within adoption research. As such, the approach in this dissertation will be to address symbolic relative advantage in an exploratory post-hoc fashion at a later time.

Second, a decision was also made to drop usage intentions from the research model since the relationship with usage is expected to be tautological (i.e., individual intentions to use the SDPM practice extensively will result in extensive usage) within the study’s context. The context of this study demands that, rather than focusing on intention formation, the pertinent question lies with explaining the effects of reasoned action, institutional pressure, and operational exigency on usage behavior of the SDPM practice itself. This approach is consistent with studies interested in explaining usage behavior of IS systems (Adams, Nelson and Todd 1992) and software development methodologies (Khalifa and Verner 2000) where intentions were dropped in order to provide a sharper focus on actual usage behavior. The simplified research model which serves as the focus of this dissertation is pictured in Figure 3.1-1.

Figure 3.1-1: Simplified Research Model


SDPM Practice Social Norms
SDPM Practice Codification
SDPM Practice Relative Advantage - Instrumental
SDPM Practice Relative Advantage - Political
SDPM Practice Complexity
SDPM Practice Routinization
• Adopted by the Project Manager
• Adopted by the SD Organization
Operational Exigency
SDPM Practice Usage
• Extent
• Faithfulness
• Customized

3.2. Hypotheses

Understanding the motivators of human action is a complex task that has been undertaken by researchers from a diverse set of fields (Fishbein and Ajzen 1975; Rogers 1995; Simon 1997; Tolbert and Zucker 1983; Triandis 1980). Consequently, the constructs that are posited to influence usage in this study are considered from the perspectives of reasoned action, decision-making and institutionalization. Before addressing the research model though, specific characteristics of SDPM practices will be discussed in order to better understand how software developer usage behavior might be influenced. Next, a reasoned action perspective will be utilized to highlight the cognitive aspects of behavioral choice. A decision-making perspective will then be applied to accentuate the conceptualization of value judgments that serve to influence SDPM practice usage. Further, institutional theory will be drawn upon to demonstrate the importance of the institutional environment on a software developer's usage decision. Finally, the discussion will conclude by addressing operational exigencies that impact SDPM practice usage behavior.

3.2.1. SDPM Practice Characteristics

Before the research hypotheses are presented, it is important to address (1) why SDPM practices matter in the context of a software development project and (2) what attributes of SDPM practices are likely to be meaningful when trying to understand usage behavior.

3.2.1.1. Types of SDPM Practices

The context of this research requires that for a SDPM practice to be considered it must be commonly used in software development environments but subject to group / individual agency. Because of their prevalence in software development environments, scope change control and structured walkthrough practices are often championed as a means of improving software development project success. However, since these two practices are enacted by each individual software developer, there is ample opportunity for agency issues to arise. Thus, these two SDPM practices will be used to evaluate the proposed research model. Scope change control practices will be considered since they are often used in projects as a means to manage project changes in light of predefined project expectations. The Project Management Institute has stressed their importance in a project setting since change is almost a guarantee in projects of any size (PMI 2003). This need is further exacerbated in a software development context since development cycles are often rapid and iterative, requiring the progressive elaboration of product scope over the life of the project (McConnell 2004). Structured walkthroughs will also be considered due to their prevalence in software development projects. In their simplest form, structured walkthroughs have been defined as "a peer group review of any product" (Yourdon 1989, p. 4) where the peer group can be technical, managerial or application users and the review can be either at the code or interface level.

3.2.1.2. SDPM Practices and their Impact on Software Development Projects

While scope change control processes and/or structured walkthroughs can be deemed mandatory within a particular development organization, it is the contention of this study that a set of individual, social and institutional factors combined with external pressures will work together to influence ***how*** the SDPM practice is actually used within the context of a given software development project. If one assumes that the software developer has to use a

SDPM practice in the way envisioned by the organization in order to improve project performance, then we must strive to understand factors which will explain the type of SDPM practice usage that is exhibited.

There are two primary characteristics of SDPM practices which make them prone to not be used in a manner consistent with the organization's expectations. First, the formal training that software developers receive is often focused on the development of technical abilities (e.g., coding practices) and not on the softer skills required to complete projects on time and within specification. Thus, software developers often view SDPM practices as not being relevant to their main task of coding, providing them an opportunity to focus less attention on enacting these practices as intended by the organization. This idea is succinctly illustrated by Yourdon (1989, p. 5) when he provides rational for why developers might not want to utilize structured walkthroughs:

> *"To a typical programmer or systems analyst, the notion of spending an hour reading through someone else's program listing or dataflow diagram makes no sense. Moreover, the thought of letting someone else look at his work strikes him as a waste of time, if not an invasion of privacy. This is even more true today in the world of microcomputers, where the industry extols the feats of lone 'cowboy' programmers who write dazzling new programs – all by themselves – on the IBM PC or Macintosh computer".*

The second attribute of SDPM practices that make usage distortion a distinct reality in SD projects relates to the progressive elaboration that underlies most projects. As stated in the Project Management Body of Knowledge, “the project scope will be broadly defined early in the project and made more explicit and detailed as the project team develops a better and more complete understanding of the objectives and deliverables” (2003, p. 6). SD projects will evolve over time, and the use of SDPM practices is directly impacted by these changes. For example, to manage scope change, one must have a clear definition of the project scope at each stage of the project. However, software developers are often removed from the actual specification of project deliverables and can be confused regarding scope changes that have or have not been formally integrated into the project. Thus, the scope change control process can become messy and, in turn, encourage a developer to question its value and adopt usage behavior that is inconsistent with organizational expectations. Now that the appropriateness of addressing SDPM practice usage has been discussed, relationships in the proposed research model will be discussed.

3.2.2. Reasoned Action Sources of Influence

Behavioral choice has often been analyzed through a reasoned action lens, such as is proposed in the Technology Acceptance Model and its derivatives (Davis 1989; Venkatesh and Davis 2000; Venkatesh et al. 2003). Reasoned action models can be traced back to the Theory of Reasoned Action (Fishbein and Ajzen 1975), a model which suggests that individual behavior results from intentions regarding the behavior. These intentions are said to be shaped from an individual's attitudes and subjective norms with respect to the behavior in question. Subjective norms will later be discussed as a normative influence that flows from the institutional environment, so the following narrative will focus on attitudes as a precursor to the formation of SDPM practice usage. A reasoned action perspective suggests that individuals cognitively evaluate each behavior before deciding whether it should be undertaken. Specifically, this perspective suggests that individuals are more prone to intend to exhibit a behavior if the benefits of complying outweigh the perceived costs.

Software developers often make usage decisions that are consistent with their internal beliefs and attitudes concerning the SDPM practice (Hardgrave et al. 2003). Behavioral research has demonstrated that internal beliefs and attitudes are salient indicators of usage behavior, often stronger and more consistent than the influence of social pressures (O'Keefe 2002). Two drivers of attitude, perceptions regarding ***complexity*** and ***relative advantage***, have proven to be of particular interest when investigating how usage intentions are formed (Agarwal 2000). Individual perceptions are especially important in the context of SDPM practices since following them can often be perceived a secondary to the developer's primary responsibility of creating and implementing software. These constructs, along with their impacts on SDPM usage, are discussed in the following paragraphs. It is important to note that much of the prior work on adoption and use of an innovation has focused on usage intentions instead of actual usage behavior, and that this work is utilized in the following discussion to more accurately uncover the impacts of reasoned action on SDPM practice usage.

3.2.2.1. SDPM Practice Complexity

Individual perceptions regarding complexity, defined as the degree to which a behavior is perceived as relatively difficult to understand and enact (Thompson et al. 1991), have often been negatively associated with an individual's decision to exhibit a particular behavior. As discussed in diffusions literature, increasing levels of complexity serve to increase the perceived costs of utilizing an innovation, attenuating the innovation's value and ultimately hindering its usage. For example, perceived complexity of an innovation has been suggested to be negatively related to the innovation's rate of adoption within a social system (Rogers 1995). Further, technology adoption research has long suggested a direct positive relationship between ease of use perceptions and usage behavior, especially in the early stages of adoption (Adams et al. 1992).

SDPM practice complexity, defined as a software developer's perception concerning the degree of difficulty associated with the understanding and use of the SDPM practice for a particular project, is similar to perceived ease of use (Davis et al. 1989) and perceived complexity (Hardgrave et al. 2003) constructs presented in IT literature. Authors have recently demonstrated both significant (van der Heijden 2004) and non-significant (Hardgrave et al. 2003) relationships between complexity and usage intention constructs. In the case of software development methodology intention formation (Hardgrave et al. 2003), the non-significant relationship between complexity and usage intentions might have resulted from the fact that adopting a new development methodology requires a radical modification of core work processes (e.g., coding) over the long-term that reduces the importance of short-term complexity evaluations. In contrast to the radical nature of methodology adoption, SDPM practice usage, implemented as a control to ensure that software will be delivered on time and within specification, is often determined on a project by project basis since it is secondary to the primary act of software development. The distance between a SDPM practice and the developer's core work function suggests that complexity will be salient when predicting SDPM practice usage. As perceptions of complexity increase, software developers will be less prone to decide the practice should be followed in three respects.

First, software developer perceptions of SDPM practice complexity are expected to have a negative influence on the ***Extent of SDPM Practice Use***, defined in this work as the degree to which a software developer uses the SDPM practice for a given development project. Software developers will be less prone to frequently use the SDPM practice if they struggle to understand it in the context of a specific project. Specifically, an increase in the perceived costs of using the SDPM practice will detract from the associated value, and in turn encourage the individual to pursue behaviors perceived to possess more value. This suggestion is consistent with findings related to the relationship between ease of use and extent-focused usage in IT adoption studies (Adams et al. 1992) and complexity and adoption behaviors in diffusions literature (Rogers 1995).

> ***H1*** *SDPM Practice Complexity will be negatively related to a software developer's Extent of SDPM Practice Use.*

Second, it is also expected that the perceived complexity of the SDPM practice will have an impact on the nature of usage that is exhibited by the software developer. ***SDPM Practice Faithful Use***, defined as the degree to which a software developer uses the SDPM practice in a manner consistent with the intentions of those individuals that created the practice, underlies a software developer's willingness to enact a behavior that is consistent with the norms that exist within the development organization. In her consideration of organizational strategic responses to institutional pressures, Oliver suggested that as complexity increases (i.e., the existence of multiple, conflicting constituent expectations), so would the likelihood of an organization actively resisting those same institutional pressures because of a belief that "the satisfaction of one constituent often requires the organization to ignore or defy the demands of another" (1991, p. 162). In the same way, software developers who believe that the SDPM practice is complex are more likely to actively resist compliance such that SDPM practice usage will be less than faithful.

> ***H2*** *SDPM Practice Complexity will be negatively related to a software developer's SDPM Practice Faithful Use.*

Finally, SDPM practice complexity is also expected to negatively impact a software developer's adaptation of the practice with regards to the current software development project. Adaptation often occurs so that an individual can derive the greatest value from utilizing the innovation (Westphal et al. 1997). In the case of a SDPM practice innovation, adaptation within each project will allow a software developer to maximize the potential value gained from utilizing the practice. ***SDPM Practice Customized Use***, defined as the degree to which a software developer has modified the SDPM practice for a given development project, requires that the developer understand the practice in order to customize it for the project at hand. Thus, adaptation is less likely to occur as a software developer's perception of practice complexity increases.

> ***H3*** *SDPM Practice Complexity will be negatively related to a software developer's SDPM Practice Customized Use.*

3.2.2.2. SDPM Practice Relative Advantage

In a technology adoption context, perceptions of usefulness (i.e., relative advantage) have often been conceptualized as the degree to which an individual believes that the behavior in question will help them perform a task better (Davis 1989). Perceived relative advantage has often been suggested to positively influence an individual's behavior. For example, a positive relationship between task-focused usefulness perceptions and usage intentions has consistently received support in IT adoption research (Venkatesh et al. 2003). However, there is some question as to whether non-task-related usefulness perceptions also play an important role in determining usage (Cooper and Bhattacherjee 2001). This is especially salient in an organizational setting where individuals may perceive a behavior to contribute to their professional career, regardless of its usefulness for the task at hand.

In their investigation of factors that drive PC utilization, Thompson et al. (1991) proposed a perceived consequence construct, which included ease of use (e.g., complexity) and perceived usefulness (job fit and long-term consequences of use) dimensions. A closer look at the two usefulness dimensions reveals one that is task-based while the other is focused on individual utility beyond the task (Johnson, Hardgrave and Doke 1999). This conceptualization of usefulness was further elaborated by Compeau et al. (1995), who found their measurement of outcome expectations to have both performance and personal dimensions. While recent authors have suggested that these two dimensions of outcome expectations actually represent the same construct (Venkatesh et al. 2003), the context of a contemporary SD environment is such that developers are often evaluated and compensated based on multiple criteria (not just development activities) and are thus likely to have multiple perspectives concerning what makes a SDPM practice useful.

A similar distinction between usefulness-related beliefs can be found in the decision-making perspective offered within management literature. In their evaluation of strategic decision making within an organization, Dean and Sharfman (1993) suggested that both procedurally-rational and political considerations were at work in the managerial decision-making process. Procedural rationality, defined as "the extent to which the decision process involves the collection of information relevant to the decision and the reliance upon analysis of this information in making the choice" (Dean and Sharfman 1993, p.

1071), and political behavior, defined as involving "acts of influence to enhance or protect the self-interest of individuals or groups" (Dean and Sharfman 1993, p. 1072), were found to be two distinct, yet not mutually exclusive, aspects of the decision making process.

A comparable view of motivators for decision-making has also been proposed within the IT literature. Austin (2001), in discussing the implications of time pressure on software development quality, proposed that two concerns must be considered when evaluating software developer decisions regarding product quality. The first, concern for career, suggests that developers might sacrifice quality if they fear that the personal consequences of admitting schedule overages might result in a significant career penalty. In addition, concern for quality is proposed to also impact quality decisions since a developer that values product quality will be more prone to undertake quality initiatives despite the existence of time pressures. Consistent with Dean and Sharfman (1993), Austin's suggestions for software developer quality motivators align with procedural-rational (concern for quality) and political (concern for self) dimensions associated with strategic decision-making. Robey and Markus (1984) drew a similar distinction by describing systems design within an organization as both a rational and political process. The following quote demonstrates how these two dimensions are suggested to be distinct, yet interdependent within the organization (Robey and Markus 1984, p. 12):

> *"The rituals of systems development perpetuate the prevailing ideology of rationality and provide an acceptable cover for unexpressible political motives in the dealings between users and designers. Overt conflict and manipulation are thereby controlled, lending stability and order to systems development. In effect, the rituals of systems development enable participants to act in their self interests without discrediting the organization's rational ideology."*

Evaluating contemporary relative advantage operationalizations in IT literature reveals two different types of usefulness perceptions that align with the proposed dimensions of decision-making motivations: *instrumental* and *political*. While there is a general acceptance that usefulness perceptions influence individual behavior, there is a lack of clarity regarding the differential impacts of instrumental and political relative advantage. As suggested before, this distinction is essential in the investigation of SDPM practice use since software developers operate in organizations where success is often defined by more than just software development performance within a specific project. The following paragraphs will discuss each type of usefulness perception and their expected impact on SDPM practice usage.

3.2.2.3. SDPM Practice Relative Advantage and Extent of Usage

In this research, the ***instrumental relative advantage of a SDPM practice*** will be defined as the degree to which software developers believe that using the SDPM practice will help them attain performance gains for the software development project. Software developers are often evaluated through the product that they create, and practices that are perceived as contributing to improved software products will be especially important to the developer. A software developer's frequent usage of the SDPM practice is proposed to be positively associated with their perception of the practice's ability to facilitate completion of the task. This assertion is consistent with a recent finding that software developer value perceptions positively impact development methodology usage that is extent-focused (Khalifa and Verner 2000).

H4 *The SDPM Practice Relative Advantage – Instrumental will be positively related to a software developer's Extent of SDPM Practice Use.*

Political relative advantage from the SDPM practice, defined as the degree to which a software developer believes that using the SDPM practice will help them create or sustain a positive impression with others in the organization, is also highly relevant in the study of software developer use of SDPM practices since the use of the practices are easily observable by others in an organizational setting. The visibility of SDPM practice use suggests that individuals might feel pressure to participate in practice usage regardless of other usefulness perceptions, especially in environments where the practice has been ingrained into the development process. A belief that utilizing the practice will help the software developer enhance their image is expected to have a positive impact on the extent to which the individual uses the SDPM practice. This idea aligns with PC usage research that has found positive perceptions of long-term consequences associated with use (i.e., use will increase the opportunity to gain job security and use will increase the opportunity for more meaningful work) (Thompson et al. 1991).

> ***H5*** *SDPM Practice Relative Advantage - Political will be positively related to a software developer's Extent of SDPM Practice Use.*

3.2.2.4. SDPM Practice Relative Advantage and Faithful Usage

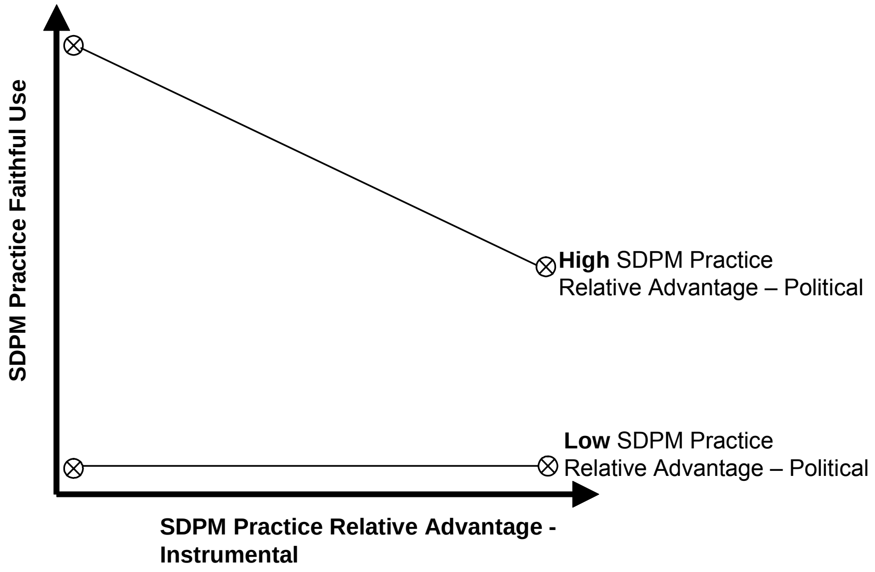
Software developer use of the SDPM practice consistent with the development organization's desires is expected to be impacted by perceptions of both instrumental and political SDPM practice value perceptions. Research evaluating the organizational adoption of Total Quality Management (TQM) programs in a heath care environment found that organizations viewing TQM adoption as a valuable means to conform with the institutional environment (i.e., late adopters) were likely to exhibit acquiescence adoption behavior (e.g., faithful adoption) as evidenced through the organization strictly adopting the institutionally accepted TQM program structure (Westphal et al. 1997). An institutional perspective suggests that the rationale for this behavior comes from an organization's desire to survive, thus motivating the organization to pursue behaviors that will help them conform to the institutional environment (DiMaggio and Powell 1983). Specifically, "in the adoption of ambiguous administrative innovations that involve actual inspection by an external agency at the level of operational routines, complete decoupling (of operational routines from formally adopted programs) may not occur, and organizations may instead accommodate institutional demands by conforming to socially legitimate operational definitions of institutional goals" (Westphal et al. 1997, p. 371).

Applying these findings at an individual level suggests that employees, motivated to "survive" within the organization, will faithfully adopt a behavior when they view it as valuable in helping them establish or enhance their standing within the organization. This influence on faithfulness can be understood through an institutional theory lens which suggests "the behaviors of individuals within organizations are significantly influenced by the prevailing organizational norms, values, culture, and history" (Purvis et al. 2001, p. 120). Political value arises when the SDPM practice is viewed as helping the software developer improve their image with others in the organization, and as such is expected to positively impact faithful use of the practice.

However, the impact of political value judgments on faithful use is expected to be tempered by perceptions of the instrumental value associated with the SDPM practice. As discussed previously, instrumental relative advantage speaks to an efficiency rationale whereby the software developer understands that using the practice will help them complete assigned project tasks. As software developers become more strongly convinced that the SDPM practice can help improve the completion of project-related tasks, they will be more likely to focus on maximizing task-efficiency and consequently be less swayed by the usage rationale expounded by the development organization. In other words, the move towards a more efficient use of the SDPM practice will often come at the expense of faithful usage.

Thus, faithful use of the SDPM practice is expected to be highest when political value is perceived to be high but instrumental value is perceived to be low. In situations where political value is high, the presence of high instrumental value perceptions are expected to attenuate the faithful use of a SDPM practice. Finally, regardless of instrumental value perceptions, low political value is expected to increase the likelihood of ironic use of the SDPM practice. These relationships are graphically illustrated in Figure 3.2-1 and verbalized in H6.

Figure 3.2-1: The Impact of Relative Advantage on Faithful Use



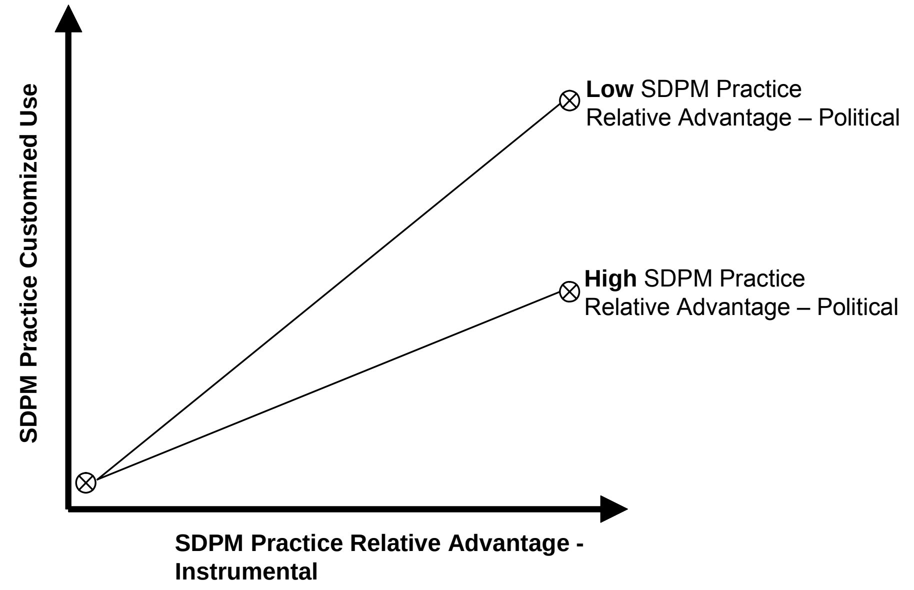
H6 *The highest level of SDPM Practice Faithful Use will occur when there is low SDPM Practice Relative Advantage – Instrumental and high SDPM Practice Relative Advantage – Political, followed by high SDPM Practice Relative Advantage – Political and high SDPM Practice Relative Advantage – Instrumental, then by low SDPM Practice Relative Advantage – Political.*

3.2.2.5. SDPM Practice Relative Advantage and Customized Use

Software developer perceptions of political and instrumental relative advantage are also expected to impact SDPM practice adaptation. First, a SDPM practice's perceived instrumental value is expected to positively impact a software developer's adaptation of the SDPM practice to the development project. In their evaluation of organizational adopters of TQM programs, researchers found that "early adopters, motivated by technical efficiency gains from adoption, are more likely to customize quality practices to the organization's unique needs and capabilities" (Westphal et al. 1997, p. 387). These results demonstrate that efficiency-minded adopters, in this case early adopters, are more likely to adapt general processes to meet specific needs. This has also been demonstrated from an IT perspective in a recent study investigating the response of users to the introduction of new IT. This research illustrated that users who view a system as an opportunity to improve personal efficiency and effectiveness (i.e., instrumental relative advantage) will increase the likelihood of undertaking adaptation activities with regards to the new IT (Beaudry and Pinsonneault 2005). Adaptation within a development project will occur when the software developer seeks to maximize the task-related value of using the SDPM practice, suggesting that the software developer would only undertake adaptation if the SDPM practice was perceived to be instrumentally valuable.

However, political value assigned to the SDPM practice is expected to provide a drag on adaptation behaviors despite any perceptions of instrumental relative advantage. High political value perceptions regarding the SDPM practice suggest that the practice has become ingrained deeply into the modus operandi of the development organization. Once this occurs, the taken-for-granted nature of the practice is expected to attenuate the likelihood that a developer will adapt it within a specific project. Thus, SDPM practice Customized Use is expected to be highest when instrumental value is perceived to be high but political value is perceived to be low. In situations where instrumental value is high, the presence of high political value perceptions is expected to attenuate the adaptation of a SDPM practice. Finally, regardless of political value perceptions, low instrumental value is expected to decrease the likelihood of adapting the SDPM practice to the project at hand. These relationships are graphically illustrated in Figure 3.2-2 and outlined in H7.

Figure 3.2-2: The Impact of Relative Advantage on Customized Use



H7 *The highest level of SDPM Practice Customized Use will occur when there is low SDPM Practice Relative Advantage – Political and high SDPM Practice Relative Advantage – Instrumental, followed by high SDPM Practice Relative Advantage – Political and high SDPM Practice Relative Advantage – Instrumental, then by low SDPM Practice Relative Advantage – Instrumental.*

3.2.3. Institutional Sources of Influence

Organizational work requires an interaction with others to produce something greater than can often be achieved alone. Within a software development milieu, this environment often dictates that the individual developer interact with business members, business analysts, IT management, and/or other software developers during the process of designing and creating a usable product. Institutionalization, involving “processes by which social processes, obligations, or actualities come to take on a rule-like status in social thought and action” (Meyer and Rowan 1977, p. 341), provides a measure of identity for the organization from which the project participant draws inferences about acceptable behavior. Behavioral research over the past several decades has demonstrated the importance of considering institutional factors when the individual being studied is operating within an organizational context (Bock, Zmud, Kim and Lee 2005; Hardgrave et al. 2003; Lewis, Agarwal and Sambamurthy 2003; Orlikowski, Yates, Okamura and Fujimoto 1995; Purvis et al. 2001). Three institutional sources of influence have been suggested to interact to promote and sustain orderly behavior: 1) regulative where institutions are viewed as providing systems of rules or governance systems, 2) normative where institutions are viewed as providing a moral framework for the conduct of social life, and 3) cultural-cognitive where institutions are viewed as providing a means by which social reality is constructed (DiMaggio and Powell 1983; Scott 2003). Consequently, a software developer working within an organizational context is expected to be influenced by normative (SDPM practice social norms), regulative (SDPM practice codification) and cultural-cognitive (SDPM practice routinization) forces within the institutional environment.

3.2.3.1. SDPM Practice Social Norms

The first aspect of institutional influence are social norms, traditionally defined as "a person's perception that most people who are important to him think he should or should not perform the behavior in question" (Fishbein and Ajzen 1975, p. 302). Models of human behavior that have been commonly accepted in behavioral research, such as the Theory of Reasoned Action (TRA), the Theory of Planned Behavior (TPB) and Triandis' model of human behavior, suggest one must consider social norms when trying to interpret individual action (Ajzen 1985; Fishbein and Ajzen 1975; Triandis 1980). Researchers have suggested that individuals may choose to exhibit a behavior that goes against their perceptions of its value if they believe people important to them think they should (Venkatesh and Davis 2000). Organizational literature on control has referred to this as clan control and has suggested that its influence on individual behavior, while often subtle, can be substantial (Ouchi 1979). This assertion is supported by technology adoption research that has found social norms to be a significant predictor of usage intentions (Taylor and Todd 1995; Venkatesh and Davis 2000).

Drawing on contemporary definitions of the term (Venkatesh et al. 2003), ***SDPM practice social norms*** will be defined in this research as the degree to which a software developer perceives that important others (i.e., those in their workgroup) believe he or she should use SDPM practice for a given project. Reasoned action models have traditionally suggested that social norms directly impact individual behavioral intentions (Fishbein and Ajzen 1975), but authors have also suggested a social information processing perspective where the social environment is suggested to shape attitudes and beliefs as illustrated in the following quote (Salancik and Pfeffer 1978, p. 227):

> *"The social context has two general effects on attitude and need statements: (1) it provides a direct construction of meaning through guides to socially acceptable beliefs, attitudes and needs, and acceptable reasons for action; (2) it focuses an individual's attention on certain information, making that information more salient, and provides expectations concerning individual behavior and the logical consequences of such behavior."*

Consequently, recent studies have begun to re-evaluate the traditional view of social norms offered in reasoned action models by considering the impacts of social norms on beliefs. For example, Hardgrave et al. found that a software developer's perception of social pressure with regards to using a particular software development methodology was positively related to both perceptions of methodology usefulness and usage intentions (2003). This finding is consistent with recent work on knowledge sharing behavior proposing that "when the behavior being studied is strongly reflective of collective action, subjective norms are likely to affect behavioral intentions directly ***and*** indirectly through attitude" (Bock et al. 2005, p. 100). The impact of social norms on attitude becomes even more relevant when the behavior has a moral component to it. For example, research investigating an individual’s intention to illegally copy software has found that the impact of social norms on intention formation is entirely mediated by usefulness perceptions (Chang 1998). The use of SDPM practices often takes on a moral status in software development environments, where developers are led to believe that not following practices can lead to utter failure. For example, in Gilb and Finzi’s *Principles of Software Engineering Management* (1988), they suggest that “projects which fail to specify their goals clearly, and fail to exercise control over even one single critical attribute, can expect project failure to be caused by that one attribute”. The “moral” aspect of SDPM practices suggests that the relationship between social norms and usage behavior will be completely mediated by relative advantage perceptions.

Diffusions literature has suggested that the perceived value of an innovation increases as it reaches critical mass (Rogers 1995). Further, technology adoption research on the impact of network externalities suggests that “the benefit that a consumer derives from the use of a good often depends on the number of other consumers purchasing compatible items” (Katz and Shapiro 1986, p. 822). In both perspectives, expanded diffusion of an innovation enhances the potential for achieving instrumental value, such as that gained through an increased availability of complementary innovations. In the case of SDPM practices, instrumental value is difficult to achieve unless an appropriate number of people also use the practice. For example, a change control process used by only one software developer in a team of 10 would not allow the team to realize the full benefits of using the process. However, the instrumental value associated with using the change control practice will increase as usage becomes normalized within the development environment since it will increase the likelihood that all project changes are appropriately managed. Further, high levels of agreement in the environment that the SDPM practice should be used reduces the amount of friction that the software developer might encounter when attempting to use the practice. In this vein, a positive relationship between task-related relative advantage and social norms has received support in literature focusing on methodology use by software developers (Hardgrave et al. 2003). This relationship is expected to hold in the current context where a software developer is assigning instrumental value to the SDPM practice in light of the normative environment.

H8 *SDPM Practice Social Norms will be positively related to a software developer's SDPM Practice Relative Advantage – Instrumental.*

While a relationship is expected between social norms and instrumental value perceptions, no relationship is suggested with political relative advantage. Political relative advantage is realized when the individual perceives that the practice will contribute to an enhanced status within the formal organization, such as through increased monetary compensation or title. Social norms speak to the web of referent others, or informal organization, as perceived by the software developer which do not necessarily possess the power to alter the formal structure of the organization. As such, social norms are not expected to influence political relative advantage perceptions but rather exert their influence on usage behavior with regards to the formation of instrumental value perceptions.

3.2.3.2. SDPM Practice Codification

The second source of institutional influence is expected to result from ***SDPM practice codification***, conceptualized as the degree to which a SDPM practice has been documented within the organization's formal project management methodology. While research that has shown that perceptions of organizational mandate (evidenced through the expressed organizational policies) positively influence a software developer's intention to use a development methodology (Hardgrave et al. 2003), this research will propose that the relationship is mediated by a software developer's perceptions regarding SDPM practice relative advantage. Specifically, the direct effect of SDPM practice codification on relative advantage can be expected when one extends the conceptualization of relative advantage to include political valuation judgments.

Research on organizational responses to institutional pressures suggests that regulative control mechanisms (i.e., documented policies) act to encourage compliance with the institutional environment by increasing the perceived value of acquiescence behavior (Oliver 1991). Codification of the practice is evidence that the practice has been embedded within the organization's policy for working on projects, which serves as a signal to software developers that their compliance is desired. Thus, software developers are expected to increase the political value assigned to the SDPM practice when the practice has been codified in the organization's development methodology.

> ***H9*** *SDPM Practice Codification will be positively related to a software developer's SDPM Practice Relative Advantage – Political.*

Codification of the SDPM practice suggests that the development organization has been able to transfer some tacit knowledge within the development process into explicit knowledge, allowing the knowledge to more easily transfer from individuals to the group (Nonaka 1994). An increased availability of explicit knowledge with regards to the SPDM practice would seem to improve software developer understanding, which might in turn increase the likelihood of a software developer ascribing meaning and value to the practice in completing project tasks. However, there is no guarantee that a software developer will amplify their perceptions of instrumental value in the face of increased understanding. In fact, an increased codification of the SDPM practice might serve to negatively impact instrumental value perceptions if the practice is poorly constructed.

So, under what circumstances might codification of the practice enhance the likelihood of improving instrumental value judgments? As the practice becomes more completely documented within the organization's formal software development methodology, individual usage of the practice will be more easily identified as compliant or non-compliant. When studying organizational IS development implications of regulative influence, Nicolaou suggested that "implicit in this mechanism is the threat of punishment or the use of force if an organization does not comply with standard practices" (1999, p. 135). Viewing codification as a means of exerting regulative influence within an environment suggests that non-compliance might induce negative consequences for individuals in the software development group. Developers in the organization are more likely to comply with SDPM practice usage as it is increasingly codified within the organization's software development methodology in order to avoid negative sanctions. In turn, the normative environment will become more favorable regarding SDPM practice usage. Thus, codification of the practice is expected to have a positive impact on instrumental value perceptions only when it first positively impacts the social norms regarding usage.

H10 *The positive relationship between SDPM Practice Codification and SDPM Practice Relative Advantage – Instrumental will be completely mediated through SDPM Practice Social Norms.*

3.2.3.3. SDPM Practice Routinization

The ***SDPM practice routinization*** construct presented in this study is defined as the degree to which a practice has become embedded within the software development organization. Drawing on conceptualizations of routinization in IT literature, this construct is understood via management support (Bock et al. 2005; Lewis et al. 2003; Purvis et al. 2001) and organizational support (Bock et al. 2005; Hardgrave et al. 2003; Venkatesh et al. 2003) dimensions. Management support, or ***adopted by the project manager***, is defined as the degree to which a project manager requires that the SDPM practice is used for a particular project while organizational support, or ***adopted by the SD organization***, is defined as the degree to which the software development organization requires that the SDPM practice is used for all software development projects. Routinization underlies the taken-for-granted status of the SDPM practice within the development organization that exists regardless of software developer beliefs. Institutional theory suggests that a powerful institutional pressure, such as when a SDPM practice is highly routinized, will increase the likelihood of conformity behavior (Oliver 1991). Routinization, then, works in concert with reasoned action sources of influence to directly shape usage behavior and is expected to directly impact usage in terms of extent and types of usage behavior rather than impacting use through perceptions of relative advantage.

Routinization of the SDPM practice occurs as the practice becomes ingrained into the operations of the development organization, taking on a rule-like status within the development process. Routinization reduces friction in the usage decision process since the SDPM practice is already perceived as being “right”. Within this context, increased routinization of the SDPM practice is expected to positively impact the extent to which a software developer will use the SDPM practice.

H11 *SDPM Practice Routinization will have a positive influence on a software developer’s Extent of SDPM Practice Use.*

The duality of structure and agency inherent in organizational routines (Feldman and Pentland 2003) suggests that it is important to evaluate ways in which routinization of the SDPM practice can impact types of usage. Westphal and colleagues (Westphal et al. 1997), in their research on the organizational adoption of TQM programs, provide a lens through which we can address the role of routinization on the type of SDPM practice use exhibited by a software developer. First, routinization is expected to influence faithful usage behavior. Their study of TQM program diffusion found that organizations tended to adopt TQM practices for legitimacy reasons if they were a late adopter of the innovation, expressed through an implementation of TQM that, rather than customized to situational needs, was isomorphic with the institutional environment (Westphal et al. 1997). Applied in our setting, this suggests that software developers will tend to use the SDPM practice faithfully when it has been routinized within the environment. Specifically, as the practice becomes more ingrained in how projects are undertaken within the organization, software developers are expected to increase the likelihood of following the practice in a manner consistent with the intentions of the development organization.

H12 *SDPM Practice Routinization will have a positive influence on a software developer's SDPM Practice Faithful Use.*

Second is the impact of routinization on adaptation behavior. As suggested by Feldmand and Pentland, "when people enact routines, they can maintain the ostensive aspect of the routine, but they can also choose to deviate from it" (Feldman and Pentland 2003, p. 108). Westphal et al.'s (1997) findings suggest that organizations are likely to customize TQM practices for efficiency rational if the organization is an early adopter of TQM within their respective industry. During the early adoption stage of an innovation, relatively few entities will have adopted the innovation and as such routinization of the innovation will be low (Rogers 1995). Early adopters in the study tended to be those that viewed TQM practices as a means for inducing technical efficiency gains (e.g., perceived an instrumental relative advantage). Thus, an organization's tendency to customize TQM practices was found to be most likely when TQM had not been heavily routinized within the institutional environment. This same behavior is expected as software developers react to the routinization of SDPM practices within the development organization. In this case, routinization of the SDPM practice is expected to exhibit a negative relationship with the software developer's adaptation of a SDPM practice.

H13 *SDPM Practice Routinization will have a negative influence on a software developer's SDPM Practice Customized Use.*

3.2.4. Operational Exigency

Operational exigency refers to the degree of urgency exerted within a given software development project by external forces. These external forces can be applied through many different events, such as an unexpected change in any of the following:

- project priority
- individual job responsibilities
- financial resource availability

- o personnel resource availability
- o organizational stability

An increase in operational exigency has specific implications for software developer use of a SDPM practice. New product development (NPD) literature has presented one possible consequence of increased urgency as fire fighting, discussed as "the allocation of scarce resources to solve unanticipated problems or 'fires'" (Repenning 2001, p. 286). A fire fighting mentality suggests that individuals facing ample resource constraints are likely to behave in ways that focus on short term benefits despite the possibility of negative long term consequences. In a NPD environment, an unhealthy focus on fire fighting within the organization has been suggested to hinder an organization's ability to successfully utilize NPD-focused processes (Repenning 2001). Applying this perspective to a SD context, an increase in perceived project urgency is expected to drain a software developer's desire or ability to focus on practices which are implemented for long-term benefits, such as SDPM practices. In environments where decision speed is a considered a premium (i.e., software development), fire fighting can even extend beyond one project and infect the modus operandi for how projects are undertaken within a development organization. Authors have referred to this as the speed trap, where decision makers are driven to focus on making the decision *right now* at the expense of making the *right decision now* (Perlow, Okhuysen and Repenning 2002).

Given the rapid and often complex nature of contemporary software development efforts (McConnell 2004), operational exigencies are expected to impede a software developer's behavior regarding use of the SDPM practice in three ways. The first relates to the impact of operational exigencies on extent of SDPM practice use. In an examination of time pressure on software development quality, Austin addressed the concern that unexpected project complications (i.e., operational exigencies) often result in shortcut taking behavior by the developer, or "decisions made in private that are motivated by a desire to stay on schedule, but are not in the best interests of the project" (Austin 2001, p. 195). While his work focused on methods which can reduce shortcut taking behavior by software developers, the basic premise suggests that project-related urgencies tend to negatively impact a software developer's behavior with respect to practices that are not directly related to completing the required development tasks.

H14 *Operational Exigency will have a negative relationship with a software developer's Extent of SDPM Practice Use.*

Additionally, operational exigency is expected to have a negative impact on the faithfulness of a developer's SDPM practice use. An institutional perspective suggests that an organization will be less likely to comply with institutional pressures when those pressures are exerted through multiple constituents and are somewhat conflicting (Oliver 1991). In a software development context, the presence of operational exigency suggests that competing agendas exist which serve to cloud a software developer's decision regarding practice usage. Thus, the presence of operational exigency is expected to reduce the likelihood of SDPM practice usage that is faithful.

H15 *Operational Exigency will have a negative relationship with a software developer's SDPM Practice Faithful Use.*

Finally, operational exigency is also expected to negatively impact a software developer's adaptation of SDPM practices within the development project. As discussed earlier, adaptation is likely to occur when the software developer believes that efficiencies can be attained by using of the practice. Research addressing user adaptation strategies in response to new IT events have suggested that individuals perceiving the event to be threatening and view their control over the event as low will tend to focus on self-preservation strategies that emphasize emotional adaptation instead of problem-focused adaptation (Beaudry and Pinsonneault 2005). In a software development context, developer perceptions of external pressure suggest a reduction in perceived control over threats to project success (i.e., completing the development tasks on time and within specification). As operational exigency continues to grow it is then expected that problem-focused adaptation, or SDPM practice Customized Use, will be reduced.

H16 *Operational Exigency will have a negative relationship with a software developer's SDPM Practice Customized Use.*

3.3. Conclusion

The preceding discussion has proposed that reasoned action, decision-making, and institutional perspectives can be utilized in concert with operational exigency in order to provide an explanation of software developer use of a SDPM practice. A summary of the research hypotheses is included in Table 3.3-1.

Table 3.3-1: Overview of Dissertation Hypotheses

H1	SDPM Practice Complexity will be negatively related to a software developer's Extent of SDPM Practice Use.
H2	SDPM Practice Complexity will be negatively related to a software developer's SDPM Practice Faithful Use.
H3	SDPM Practice Complexity will be negatively related to a software developer's SDPM Practice Customized Use.
H4	The SDPM Practice Relative Advantage – Instrumental will be positively related to a software developer's Extent of SDPM Practice Use.
H5	SDPM Practice Relative Advantage - Political will be positively related to a software developer's Extent of SDPM Practice Use.
H6	The highest level of SDPM Practice Faithful Use will occur when there is low SDPM Practice Relative Advantage – Instrumental and high SDPM Practice Relative Advantage – Political, followed by high SDPM Practice Relative Advantage – Political and high SDPM Practice Relative Advantage – Instrumental, then by low SDPM Practice Relative Advantage – Political.
H7	The highest level of SDPM Practice Customized Use will occur when there is low SDPM Practice Relative Advantage – Political and high SDPM Practice Relative Advantage – Instrumental, followed by high SDPM Practice Relative Advantage – Political and high SDPM Practice Relative Advantage – Instrumental, then by low SDPM Practice Relative Advantage – Instrumental.
H8	SDPM Practice Social Norms will be positively related to a software developer's SDPM Practice Relative Advantage – Instrumental.

H9	SDPM Practice Codification will be positively related to a software developer's SDPM Practice Relative Advantage – Political.
H10	The relationship between SDPM Practice Codification and SDPM Practice Relative Advantage – Instrumental will be completely mediated through SDPM Practice Social Norms.
H11	SDPM Practice Routinization will have a positive influence on a software developer's Extent of SDPM Practice Use.
H12	SDPM Practice Routinization will have a positive influence on a software developer's SDPM Practice Faithful Use.
H13	SDPM Practice Routinization will have a negative influence on a software developer's SDPM Practice Customized Use.
H14	Operational Exigency will have a negative relationship with a software developer's Extent of SDPM Practice Use.
H15	Operational Exigency will have a negative relationship with a software developer's SDPM Practice Faithful Use.
H16	Operational Exigency will have a negative relationship with a software developer's SDPM Practice Customized Use.

Chapter IV: Research Methodology

Building on the proposed research model, this chapter will focus on the research methodology utilized for this study. This discussion begins by providing an overview of the data collection approach. Second, an overview of the focal PM practices that are to be investigated in the context of this study is provided. Finally, the operationalization of constructs arising from social, individual and institutional sources of influence on the software developer is presented.

4.1. Data Collection Methods

The primary focus of this study is to understand SDPM practice usage behaviors within the context of a specific SD project in a contemporary organization. As such, data was collected directly from individuals operating within an organizational context. An effective way of accomplishing this is through semi-structured interviews and the administration of survey-based instruments. In order to reduce the common method variance problems that have been suggested to plague behavioral research (Doty and Glick 1998), multiple and non-concurrent data sources are utilized. A semi-structured interview and web-based survey was used to gather organization-level information from senior management within the SD group. Next, a web-based survey was utilized to collect project-level data from the project manager for each development project. A web-based survey instrument was then administered to software developers at two different points in time, and they responded to items within the context of a specific SD project. These different data collection methods are discussed in greater detail in the following paragraphs.

4.1.1. Semi-structured Interview & Web-based Survey: Software Development Manager

The first step in the data collection process was to conduct a semi-structured interview (either in person or via the phone) with the manager of software development efforts within each organization. An interview schedule has been provided in Appendix V. The interview was used to verify that the organization met the constraints detailed in Table 4.6. Once the organization was verified as an appropriate data source for this study, data was collected from the software development manager via the web-based survey provided in Appendix VI. Data to be collected included identifying projects / project managers that are candidates for this survey along with the information detailed in Table 4.1-1.

Table 4.1-1: Data Collected in the Software Development Manager Semi-Structured and Web-based Interviews

Data to be Collected	**Response Description**
SDPM Practice Codification	7 pt. Likert-based scale (Strongly Agree to Strongly Disagree)
SDPM Practice Routinization	7 pt. Likert-based scale (Strongly Agree to Strongly Disagree)
Management Expectations	7 pt. Likert-based scale (Strongly Agree to Strongly Disagree)
# of software developers in the software development group	Integer
Annual budget of the software development group	Integer
Software Development Manager Professionalization[4]	Integer
• Does the software development manager hold membership with any PM professional society?	Yes / No
• Have the software development manager received formal PM training in the last 2 years?	Yes / No
• Does the software development manager hold any project management certifications?	Yes / No

Finally, and essential to the success of this dissertation, the interview was used to engender support for the research so as to encourage the participation of project managers and software developers in the data collection efforts discussed next.

4.1.2. Web-based Survey: Project Manager

Utilizing the project manager contact information collected in the semi-structured interview, project managers were then contacted via e-

[4] Application Development Manager Professionalization is represented as a summation of the three Yes/No items that are listed after it.

mail to request their participation in a questionnaire-based survey (Appendix VII). The purpose of this questionnaire was to collect project-level information that applies to a software developer's SDPM practice usage decision. Understanding the importance of follow-up to increase response rates (Dillman 2000), responses were monitored and subsequent e-mails re-sent to project managers in one week intervals for a total of three weeks for those that did not complete the survey. Project managers were asked to provide a description of their development project and identify software developers currently assigned. Additional information collected during this phase is detailed in Table 4.1-2.

Table 4.1-2: Data Collected in the Project Manager Questionnaire

Data to be Collected	**Response Description**
Percentage of project that has been completed	Percentage
# of developers assigned to the project	Integer
# of hours assigned to the project	Integer
Strategic Value of the Project[5]	Integer
• Expected financial value of the project	7 pt semantic differential (low value to high value)
• Budget assigned to the project	7 pt semantic differential (small budget to large budget)
• Strategic necessity of the project	7 pt semantic differential (not necessary to very necessary)
Complexity of the project	7 pt semantic differential (simple to complex)
Innovativeness of the project	7 pt semantic differential (conventional to innovative)
Project Manager Professionalization[6]	Integer
• Does the project manager hold membership with any PM professional society?	Yes / No
• Has the project manager received formal PM training in the last 2 years?	Yes / No
• Does the project manager hold any project management certification?	Yes / No

4.1.3. Web-based Survey: Software Developer (Time 1)

The next step was to utilize software developer contact information provided in the project manager survey in order to request participation from software developers via e-mail (Appendix VIII).

[5] Strategic Importance of the Project is represented as an average of the three items that are listed after it.

[6] Project Manager Professionalization is represented as a summation of the three Yes/No items that are listed after it.

Similar to the project manager survey, responses were monitored and subsequent e-mails re-sent to software developers in one week intervals for a total of three weeks for those that had not completed the survey. Data to be collected from software developers in this survey are listed in Table 4.1-3.

Table 4.1-3: Data Collected in the Software Developer Questionnaire (Time 1)

Data to be Collected	**Response Description**
# years of development experience	Integer
# years of experience with the organization's scope change control practice	Integer
# years of experience with a scope change control practice in any organization	Integer
# years of experience with the organization's structured walkthrough practice	Integer
# years of experience with a structured walkthrough practice in any organization	Integer
Gender	Male / Female
Age	Integer
Voluntariness of using the practice	7 pt. Likert-based scale (Strongly Agree to Strongly Disagree)
Management support	7 pt. Likert-based scale (Strongly Agree to Strongly Disagree)
SDPM practice relative advantage – instrumental	7 pt. Likert-based scale (Strongly Agree to Strongly Disagree)
SDPM practice relative advantage – political	7 pt. Likert-based scale (Strongly Agree to Strongly Disagree)
SDPM practice social norms	7 pt. Likert-based scale (Strongly Agree to Strongly Disagree)
SDPM practice complexity	7 pt. Likert-based scale (Strongly Agree to Strongly Disagree)
Software Developer Professionalization[7]	Integer
• Does the software developer hold membership with any PM professional society?	Yes / No
• Has the software developer received formal PM training in the last 2 years?	Yes / No
• Does the software developer hold any project management certification?	Yes / No

[7] Software Developer Professionalization is represented as a summation of the three Yes/No items that are listed after it.

4.1.4. Web-based Survey: Software Developer (Time 2)

Finally, software developers that participated in the Time 1 survey were e-mailed to request their participation in the second survey (Appendix IX). This e-mail was sent ***approximately one month*** after their response to the Time 1 survey in order to allow sufficient time between the surveys for SDPM practice use to be realized. Data to be collected during this survey is detailed in Table 4.1-4.

Table 4.1-4: Data Collected in the Software Developer Questionnaire (Time 2)

Data to be Collected	Response Description
Operational Exigency	7 pt. Likert-based scale (Strongly Agree to Strongly Disagree)
SDPM Practice Usage – Extent	Integer
SDPM Practice Faithful Use	7 pt. Likert-based scale (Strongly Agree to Strongly Disagree)
SDPM Practice Customized Use	7 pt. Likert-based scale (Strongly Agree to Strongly Disagree)

Table 4.1-5 summarizes the collection of construct data within the various data collection points. The semi-structured interview was conducted face to face or over the phone, while the remaining questionnaire-based surveys were administered via a web-based survey (accessed via an e-mailed link).

Table 4.1-5: Construct Data Collection Summary

	STEP 1 Semi-structured Interview and Web-based Survey (Software Dev. Mgr.)	STEP 2 Questionnaire-based Survey (Project Mgr.)	STEP 3 Questionnaire-based Survey (Software Dev. – T1)	STEP 4 Questionnaire-based Survey (Software Dev. – T2)
SDPM Practice Codification	☑			
SDPM Practice Routinization	☑			
Management Expectations	☑			
Strategic Importance of the Project		☑		
Management Support			☑	
SDPM Practice Voluntariness			☑	
SDPM Practice Complexity			☑	
SDPM Practice Relative Advantage – Instrumental			☑	
SDPM Practice Relative Advantage – Political			☑	
SDPM Practice Social Norms			☑	
Operational Exigency				☑
SDPM Practice Usage – Extent				☑
SDPM Practice Faithful Use				☑
SDPM Practice Customized Use				☑

4.1.5. Pilot Study

An initial pilot study for this research was conducted in the summer of 2004. Conducting this study (1) confirmed the effectiveness of a web-based survey data collection approach, (2) helped to refine and expand portions of the research model, and (3) provided a means for generating effect sizes which will be utilized to determine the appropriate sample size for the primary study. Before the primary data collection was conducted, a second pilot study was undertaken in order to (a) ensure the face validity of the various construct operationalizations and (b) re-evaluate mechanics of the data collection procedure. The pilot study was implemented through two separate focus groups, each with 3 to 5 software developers and project managers from the same organization. Face validity, where "trained or untrained individuals would look at the test and decide whether or not the test measures what it was supposed to measure" (Kerlinger and Lee 2000, p. 668), is an important first step in determining the suitability of items in measuring desired constructs in a given context (Shadish, Campbell and Cook 2002). During the focus group, each construct item was discussed to determine whether they captured the intended concept. For those found to be inadequate, the discussion continued to determine how the item might be reworded to capture the essence of the construct. Further, the data collection procedure outlined in earlier sections was reviewed in order to verify its suitability and potential effectiveness in collecting data. Results from the pilot study were used to tweak construct measurement and data collection approaches for the primary study.

4.2. Participant Constraints

This study required that organizations, projects and software developers included in this study met the specific set of criteria presented in Table 4.2-1 and detailed in the following sections.

Table 4.2-1: Participant Constraints

Level	**Constraint**
Organization	1. Internal software development function 2. Must profess to use at least one SDPM practice being investigated in this study (scope change control or structured walkthrough)
Project	1. Requires at least 20 days to complete 2. Person designated to oversee the project (e.g., project manager) can't respond as a software developer for that project
Software Developer	1. Must work in an organizational setting 2. No more than 75% complete with their portion of the project at the time when the first survey is administered 3. Must have an opportunity to utilize at least one of the SDPM practices between the administration of the initial and final surveys

4.2.1. Required Characteristics of Participating Organizations

The focus in this research was on evaluating software developer use of SDPM practices in an organizational environment. As such, organizations participating in the study were required to meet two primary criteria. First, they had to have an internal software development function which develops software for either internal consumption or use by those external to the organization. Second, each organization had to profess to utilize one or more of the focal SDPM practices examined in this study (scope change control or structured walkthrough) within their software development projects. The software being produced by the organization could either be for internal use or developed specifically for external clients.

4.2.2. Required Characteristics of Participating Projects

The research question being addressed also forced two constraints on the SD projects to be considered in this research. First, the measurement of actual use is temporally separated from many of the perceptual measures in this study, demanding projects of ample size. Specifically, SD projects included in this study had to require at least 20 working days to complete. Further, since project manager and software developer responses are collected separately, projects were only considered where the individual overseeing the project was not the only software developer for the project. In other words, a software developer in the study could not be their own project manager. While the number of software developers within the project was collected as a means of assessing project size, projects were considered as long as at least one software developer responded to the survey.

4.2.3. Required Characteristics of Participating Software Developers

Finally, the focus of this study forced several constraints on software developer participants in addition to those already discussed. First, the software developer had to be working in an organizational setting. Second, the developer could be no more than 75% complete with their portion of the project at the time when the first survey was administered. This constraint was necessary since the first survey measures user perceptions and necessitates a period of time following the survey where the individual can make choices regarding the behavior in question.

4.3. Control Variables

While the constraints mentioned above allowed the collection of data from only appropriate individuals, it was also important to recognize that characteristics within the sample could generate noise that prevent a meaningful interpretation of the data. Thus, the study necessitated that adequate control data is collected to help remove the potential noise from subsequent results. Similar to the constraints discussed in Table 4.6, controls are discussed in reference to the organization, project and software developers being examined in this study. Table 4.3-1 provides and overview of the controls utilized in this study, each of which are discussed in greater detail in the paragraphs below.

Table 4.3-1: Study Controls

Level	**Control (Data Source)**
Organization	1. Software development group size (software development manager) - # software developers - annual budget
Project	1. # of developers assigned to the project (project manager) 2. # of hours assigned to the project (project manager) 3. Strategic importance of the project (project manager) - Financial value of the project - Budget assigned to the project - Competitive necessity of the project
Software Developer	1. # years of experience with the organization's scope change control practice (software developer) 2. # years of experience with the organization's structured walkthrough practice (software developer) 3. Voluntariness of using the scope change control practice (software developer) 4. Voluntariness of using the structured walkthrough practice (software developer) 5. Management support for the scope change control practice (software developer) 6. Management support for the structured walkthrough practice (software developer)

4.3.1. Organizational Controls

Data was collected from individuals operating in different organizations, requiring the use of organizational level controls. Since the software development efforts occur within the context of the organization’s software development function, data was collected from the manager of software development efforts. Specifically, the size of the software development group, as measured by the number of software developers and the annual operating budget was collected.

4.3.2. Project Controls

The focus on use within a project context also mandates the use of some project-level controls. The first of these relates to project size, as reflected through the number of developers and the number of hours assigned to complete the project.

The second relates to the project’s strategic importance to the organization, defined as the degree to which a project is considered valuable by management in comparison to other software development projects in the organization, as represented by the financial value, budget and competitive necessity of the project in comparison to other projects within the software development group. The three items below were utilized to address the strategic importance of the project in which a developer worked. Since these items are reflective of the project value in relation to other projects within the organization, project managers were chosen as the respondents for assessing strategic importance. Each item was measured on a 7-point Likert-based scale where 1 represents Strongly Disagree and 7 represents Strongly Agree.

SI1 In relation to other software development projects undertaken within your organization, what level of financial value does the organization expect to derive from the result of this project?

SI2 In relation to other software development projects undertaken within your organization, what is the size of the budget assigned to this project?

SI3 In relation to other software development projects undertaken within your organization, to what degree is the successful completion of this project necessary to improve the organization’s ability to be competitive?

4.3.3. Software Developer Controls

Software developer behavior is the ultimate focus of this study which requires that certain individual-level controls be employed. The focus on SDPM practice usage suggests that experience in relation to each practice must be captured. In addition, previous research on adoption has suggested that perceptions of voluntariness can have an impact on usage behavior (Hardgrave et al. 2003; Venkatesh et al. 2003), and as such suggests that voluntariness should be assessed with respect to each software developer. Adapting measures of voluntariness employed in previous research (Hardgrave et al. 2003; Moore and Benbasat 1991), the first three items (VOL1, VOL2 and VOL3) were included in the initial survey instrument along with two additional items developed for this study (VOL4 and VOL5) and administered to software developers in the first survey instrument.

Each item was measured on a 7-point Likert-based scale, where 1 represented Strongly Disagree and 7 represents Strongly Agree.

VOL1 My use of our (scope change control \ walkthrough) practice is voluntary.

VOL2 My supervisor does not require me to use our (scope change control \ walkthrough) practice.

VOL3 Although it may be helpful, using our (scope change control \ walkthrough) practice is certainly not compulsory in my job.

VOL4 Whether I use our (scope change control \ walkthrough) practice is entirely up to me.

VOL5 Use of our (scope change control \ walkthrough) practice is at the discretion of the employee concerned.

Finally, researchers have also suggested the level of perceived management support within an organization is an important predictor of usage behavior (Jeyaraj, Rottman and Lacity 2006). To accommodate this concept, five items were developed to measure the degree to which a developer perceives that management supports the use of the SDPM practice within a given project. Each item was measured on a 7-point Likert-based scale, where 1 represented Strongly Disagree and 7 represents Strongly Agree.

MS1 Management in our group completely supports my use of our (scope change control \ walkthrough) practice on this project.

MS2 Management in our group has provided the necessary training to enable my use of our (scope change control \ walkthrough) practice on this project.

MS3 Management in our group has provided sufficient time to permit the use of our (scope change control \ walkthrough) practice on this project.

MS4 Management in our group has provided the necessary resources to enable my use of our (scope change control \ walkthrough) practice on this project.

MS5 Management in our group is fully committed to my use of our (scope change control \ walkthrough) practice on this project.

4.4. Focal SDPM Practices

As discussed previously, scope change control and structured walkthrough practices are the SDPM practices targeted in this study. SDPM practices are expected to be defined by management within the software development group, and as such will vary widely in their instantiation across the responding organizations. However, both scope change control and structured walkthrough practices have a general definition that has been applied for this study. A discussion of what is meant by each of these practices is included in the next section.

4.4.1. Scope Change Control

According to the Project Management Institute, "Project Scope Management includes the processes required to ensure that the project includes all the work required, and only the work required, to complete the project successfully. Project scope management is primarily concerned with defining and controlling what is and is not included in the project" (PMI 2003, p. 103). In this vein, scope change control is discussed as a formal process which is "concerned with influencing the factors that create project scope changes and controlling the impact of those changes" (PMI 2003, p. 119).

4.4.2. Structured Walkthrough

A structured walkthrough "is simply a peer group review of any product... Walkthroughs can take place at various times in the development of a system. Also, a walkthrough can have a range of formats and can involve different groups of people. Despite the variation, the underlying activity remains the same: A group of peers - people at roughly the same level in the organization - meet to review and discuss a product...they can take place between system developers and end users, or among a group of end users who are building their own system" (Yourdon 1989, pp. 4 - 5). Structured walkthroughs are also referred to as code reviews, design reviews, or inspections.

4.5. Construct Operationalization

Drawing on reasoned action models of individual behavior, decision-making perspectives, and institutional theory, the next sections will discuss operationalizations of constructs presented in the research model.

4.5.1. Reasoned Action Sources of Influence

Both complexity and relative advantage constructs illustrate the commonly acknowledged reasoned action sources of behavioral influence. While complexity is conceptualized in a similar fashion to previous behavioral studies, relative advantage is suggested to exist in two different manifestations within this study. Each of these reasoned action sources of influence are represented through the perceptions of the software developer, and as such are collected through a survey instrument completed by each software developer. The operationalization of these three constructs will be discussed in the following paragraphs.

4.5.1.1. SDPM Practice Complexity

The first individual source of influence relates to the perceived difficulty associated with using the SDPM practice. Perceived complexity has commonly been suggested to influence individual attitudes and behavioral intentions to exhibit a behavior (Davis 1989; Davis et al. 1989). Since it is a measure of individual perceptions, the construct has most often been captured through responses to self-report items measured on Likert-based scales.

In the context of this study, ***SDPM practice complexity*** is defined as a software developer’s perception concerning the degree of difficulty associated with the understanding and use of the SDPM practice for a particular project. The construct has been operationalized by adapting the unified effort expectancy construct developed by Venkatesh, et al. (2003), with the addition of one item developed for this study (CPLX5). These items were captured through the first survey instrument administered to software developers. Each of the five items below was measured on a 7-point Likert-based scale, with 1 representing Strongly Disagree and 7 representing Strongly Agree.

CPLX1 I think our (scope change control \ walkthrough) practice is clear and understandable.

CPLX2 It is easy for me to become skillful at using the (scope change control \ walkthrough) practice.

CPLX3 I find our (scope change control \ walkthrough) practice easy to use.

CPLX4 Learning to use our (scope change control \ walkthrough) practice is easy for me.

CPLX5 Our (scope change control \ walkthrough) practice is easily implemented.

4.5.1.2. Relative Advantage

Relative advantage is conceptualized in this research as being manifested through instrumental and political components. This distinction is important since the use of a SDPM practice can have implications for the software developer's ability to complete the project at hand as well as on the progression of their career within the organization and subsequent image within the company. Both of these constructs have been addressed in various ways in literature, but never within the larger umbrella of individual usefulness perceptions. Instrumental and political conceptualizations of relative advantage are presented in the next several paragraphs.

4.5.1.2.1. SDPM Practice Relative Advantage – Instrumental

Task-focused usefulness perceptions are the most common conceptualization of relative advantage employed in adoption literature. For instance, task-focused perceived usefulness is a core construct in TAM research, defined as “the user’s subjective probability that using a specific application system will increase his or her job performance within an organizational context” (Davis et al. 1989). Similarly, ***SDPM practice relative advantage – instrumental*** is utilized in this study and defined as the degree to which a software developer believes that using the SDPM practice will help them attain gains in performance for the software development project. Operationalization of the construct has been adapted from the Perceived Usefulness construct in Hardgrave, et al. (2003). Two of the original items were dropped since they did not focus on task-related usefulness perceptions. One item was developed specifically for this study (RAI5), providing the following five items that are captured on a 7-point Likert-based scale where 1 is Strongly Disagree and 7 is Strongly Agree.

RAI1 Using our (scope change control / walkthrough) practice improves my performance on this project.

RAI2 Using our (scope change control / walkthrough) practice increases my productivity for this project.

RAI3 Using our (scope change control / walkthrough) practice enhances my quality of work for this project.

RAI4 Our (scope change control \ walkthrough) practice is useful for this project.

RAI5 Our (scope change control \ walkthrough) practice allows me to work more efficiently on this project.

4.5.1.2.2. SDPM Practice Relative Advantage – Political

While task-related measures of perceived usefulness are most common in adoption research, they are by no means the only usefulness perceptions that can impact individual behavior. The software developers being examined in this study operate within an organizational setting, and as such are subject to the political forces inherent in a social context. Political behavior provides a means to "enhance or protect the self-interest of individuals or groups" (Dean and Sharfman 1993, p. 1072). ***SDPM practice relative advantage - political*** is defined as the degree to which a software developer believes that using the SDPM practice will help them create or sustain a positive impression with others in the organization.

The political nature of perceived usefulness has been addressed in literature through several different constructs. First, Moore & Benbasat utilized an image construct, defined as “the degree to which use of an innovation is perceived to enhance one’s image or status in one’s social system” (1991, p. 195). Image has been considered an important component of individual perceptions of relative advantage (Rogers 1995), and has often been included in the measurement of perceived usefulness. Further, Compeau and her colleagues also addressed this type of usefulness through an Outcome Expectations (Personal) construct which relates to “expectations of change in image or status or to expectations of rewards, such as promotions, raises or praise” (Compeau, Higgins and Huff 1999, p. 148).

Items for the SDPM practice relative advantage - political construct are drawn from two sources, the first three (RAP1, RAP2 and RAP3) being adapted from items suggested to measure image (Moore and Benbasat 1991). The remaining two items (RAP4 and RAP5) have been adapted from items in the Outcome Expectations (Personal) construct that are focused on political aspects of usefulness (Compeau et al. 1999). All five items were measured on 7-point Likert-based scales where 1 represents Strongly Disagree and 7 represents Strongly Agree.

RAP1 Software developers in my organization who use the (scope change control / walkthrough) practice have more prestige than those who do not.

RAP2 Using our (scope change control / walkthrough) practice will improve my image within the organization.

RAP3 Because of my use of our (scope change control / walkthrough), others in the organization see me as a more valuable employee.

RAP4 If I use our (scope change control / walkthrough) practice my coworkers will perceive me as competent.

RAP5 Software developers who use our (scope change control / walkthrough) practice are regarded highly within the organization.

4.5.2. Institutional Sources of Influence

A second area of influence on a software developer's behavior results from institutional sources of influence that exist within an organizational context. Consistent with institutional theories, this study will propose that institutional influence is exerted through regulative, normative, and cultural-cognitive mechanisms (DiMaggio and Powell 1983; Scott 2003) as discussed in the following paragraphs.

4.5.2.1. SDPM Practice Social Norms

The first source of institutional influence arises through a normative mechanism, addressed in this research through social norms. Behavioral models such as the Theory of Reasoned Action (TRA), the Theory of Planned Behavior (TPB), Triandis' model of human behavior, the Technology Acceptance Model 2 (TAM2), and the Unified Theory of Acceptance and Use of a Technology (UTAUT) have all recognized the importance of addressing the impact that social pressures can have on individual behavior (Ajzen 1985; Davis 1989; Fishbein and Ajzen 1975; Triandis 1980; Venkatesh and Davis 2000). Social norms have been discussed with other terms such as social influence (Venkatesh et al. 2003), subjective norm (Fishbein and Ajzen 1975), and social factors (Thompson et al. 1991). Despite the differing terms, social norms are typically defined as "a person's perception that most people who are important to him think he should or should not perform the behavior in question" (Fishbein and Ajzen 1975, p. 302). In this study, ***SDPM practice social norms*** are defined as the degree to which a software developer perceives that important others (e.g., those in their workgroup) believe he or she should use the SDPM practice for a given project. Since this construct deals with perceptions of the developer, social norms were captured through the first survey instrument administered to software developers.

The validated operationalization of subjective norm by Taylor and Todd provide the framework for the first two social norm items (SN1 and SN2). The remaining three items (SN3, SN4 and SN5) were developed for the purposes of this study, and the resulting questions were measured on a 7-point Likert scale, where 1 represents Strongly Disagree and 7 represents Strongly Agree.

SN1 Co-workers whose opinion I value think that I should use our (scope change control \ walkthrough) practice when working on assigned software development projects.

SN2 Co-workers who are important to me think that I should use our (scope change control \ walkthrough) practice when working on assigned software development projects.

SN3 Co-workers whose opinion I value think that using our (scope change control \ walkthrough) practice is important when working on assigned software development projects.

SN4 In general, co-workers whose opinion I value support the use of our (scope change control \ walkthrough) practice.

SN5 Co-workers whom I regard highly have supported the use of our (scope change control \ walkthrough) practice.

4.5.2.2. Management Expectations

Management expectations, defined as the extent to which SDPM practice use is incorporated into the software developer's formal performance evaluation, was used to collect this information. In order to determine what items should be included in the instrument, a list was generated and discussed with members of the dissertation committee who were willing to suggest modifications, additions and deletions. Their feedback resulted in the four items presented below. Software development managers within each group were asked to answer the four items detailed below since they are expected to be the most appropriate source for information relating to developer performance evaluations. Each item was measured on a 7-point Likert-based scale where 1 represents Strongly Disagree and 7 represents Strongly Agree.

ME1 An important component of each software developer's formal performance evaluation is the degree to which they use our (scope change control / structured walkthrough) practice.

ME2 Periodic managerial reviews are conducted to determine the degree to which software developers are using our (scope change control / structured walkthrough) practice.

ME3 Software developer performance is evaluated in part with regards to their use of our (scope change control / structured walkthrough) practice.

ME4 In our organization, software developer use of our (scope change control / structured walkthrough) practice is formally reviewed during periodic performance evaluations.

4.5.2.3. SDPM Practice Codification

The second source of institutional influence on individual behavior results from regulative pressures exerted through the documented policies and practices within the organization. ***SDPM practice codification***, defined as the degree to which the SDPM practice has been documented within the organization's formal project management methodology, was used to address these regulative pressures. Since this construct addresses the contents of an organization's development methodology, it was captured through five items developed for the purpose of this study. Codification items were assessed during the semi-structured interview with the manager over software development efforts. Responses to each item were on a 7-point Likert-based scale where 1 represents Strongly Disagree and 7 represents Strongly Agree.

COD1 A detailed description of our (scope change control \ walkthrough) practice is included within the organization's formal project management methodology.

COD2 Our (scope change control \ walkthrough) practice is thoroughly documented within the organization's formal project management methodology.

COD3 A comprehensive description of our (scope change control \ walkthrough) practice exists within the organization's formal project management methodology.

COD4 A clearly documented process for our (scope change control \ walkthrough) practice is available to software developers in this organization.

COD5 A detailed description of our (scope change control \ walkthrough) practice is readily available to software developers in this organization.

4.5.2.4. SDPM Practice Routinization

The final source of institutional influence arises from cultural-cognitive sources, addressed in this research through a SDPM practice routinization construct. Conceptualizations of institutionalization in IT adoption literature have often focused on that achieved through organizational support (Bock et al. 2005; Purvis et al. 2001; Venkatesh et al. 2003) and/or management support (Lewis et al. 2003; Purvis et al.

2001). Consistent with these perspectives concerning the source of institutionalization, ***SDPM practice routinization*** is defined as the degree to which the software development organization requires that the SDPM practice is used for all software development projects. The five items developed for the purpose of this study were answered by the individual who manages software development efforts within the development group and were measured on a 7-point Likert-based scale where 1 represents Strongly Disagree and 7 represents Strongly Agree.

ROUT1 Our (scope change control \ walkthrough) practice is conventionally used in projects by those within the software development group.

ROUT2 Our (scope change control \ walkthrough) practice is routinely used on projects within the software development group.

ROUT3 Using our (scope change control \ walkthrough) practice in projects is second-nature to those within the software development group.

ROUT4 Using our (scope change control \ walkthrough) practice is customary for projects within the software development group.

ROUT5 Those within the software development group regularly employ our (scope change control \ walkthrough) practice in their projects.

4.5.3. SDPM Practice Usage and Operational Exigency

Both reasoned action and institutional pressures are positioned in this model as antecedents to SDPM practice usage. Further, operational exigency is expected to influence developer usage of the

SDPM practice. SDPM usage and operational exigency operationalizations are presented in the following sections.

4.5.3.1. SDPM Practice Usage

Usage is discussed along three dimensions in this study: extent, faithfulness and adaptation. SDPM practice usage data was captured via the second software developer survey instrument, administered approximately one month after the first software developer survey.

4.5.3.1.1. SDPM Practice Usage - Extent

Extent of use has often been addressed in technology adoption literature via frequency (number of times used) and/or depth measures (number of features used) (Adams et al. 1992; Davis 1989; Igbaria, Guimaraes and Davis 1995; Taylor and Todd 1995; Thompson et al. 1991). Consistent with these studies, SDPM practice extent of usage was captured by asking the software developer to identify the number of times that the practice has been used since the first survey was conducted.

USEE1 How many times have you used the (scope change control \ walkthrough) practice since you completed the initial survey on --/--/----? (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, or 10 or more)

4.5.3.1.2. SDPM Practice Faithful Use

Drawing on the work of authors attempting to measure faithfulness as it is discussed in Adaptive Structuration Theory (Chin, Gopal and Salisbury 1997), a measure of ***SDPM practice faithful usage***, defined as the degree to which a software developer uses the SDPM practice in a manner consistent with the intentions of those individuals that created the practice, was adapted for the study at hand. Specifically, the references to an electronic meeting system were modified to reflect the SDPM practice. Each item was measured on a 7-point Likert-based scale where 1 represents Strongly Disagree and 7 represents Strongly Agree.

FAITH1 The individuals who developed our (scope change control \ walkthrough) practice would disagree with how I used it.

FAITH2 If I described my usage of our (scope change control \ walkthrough) practice to the individuals who developed it, they would tell me that I used it improperly.

FAITH3 The individuals who developed our (scope change control \ walkthrough) practice would view my use as inappropriate.

FAITH4 I didn’t use our (scope change control \ walkthrough) practice in a manner consistent with how the individuals who developed the practice would believe it should be used.

FAITH5 The individuals who developed our (scope change control \ walkthrough) practice would not think that I am using our practice in the most appropriate fashion.

4.5.3.1.3. SDPM Practice Customized Use

Adaptation behaviors are often exhibited when entities customize a practice or set of practices to the current situation in order to increase efficiencies (Westphal et al. 1997). Because of a desire to understand ***how*** adaptation is manifested, it has often been assessed in technology adoption literature through qualitative means (Beaudry and Pinsonneault 2005; Majchrzak, Rice, Malhotra, King and Ba 2000; Tyre and Orlikowski 1994). Important to this study, however, is the understanding of how factors within the existing project will impact software developer adaptation behavior. As such, a survey-based approach for measuring practice adaptation was developed to determine software developer adaptation. In order to determine what items should be included in the instrument, a list was generated and sent to several experts in adoption research who were willing to suggest modifications, additions and deletions. Their feedback resulted in the five items presented below. Responses to these items were collected via a survey instrument administered to software developers at the second time period. Each item was measured on a 7-point Likert-based scale where 1 represented Strongly Disagree and 7 represented Strongly Agree.

CUST1	I customized our (scope change control \ walkthrough) practice for this project to better meet my needs.

CUST2 I have modified the (scope change control \ walkthrough) practice to meet my needs on this project.

CUST3 I have adapted our (scope change control \ walkthrough) practice for this particular project.

CUST4 I tailored our (scope change control \ walkthrough) practice to fit my needs on this particular project.

CUST5 I had to change some aspects of our (scope change control \ walkthrough) practice to meet my needs on this project.

4.5.3.2. Operational Exigency

Operational exigency is defined as the degree of urgency exerted within a given project by external forces. Since no existing instruments capture the essence of this construct, items were developed specifically for the purposes of this study. In order to determine what items should be included in the instrument, a list was generated and sent to several experts in adoption research who were willing to suggest modifications, additions and deletions. Their feedback resulted in the seven items presented below. Understanding that the construct addresses software developer perceptions of urgency, responses to these items were collected via a survey instrument administered to software developers at the second time period. Each item was measured on a 7-point Likert-based scale where 1 represents Strongly Disagree and 7 represents Strongly Agree.

EXG1 Individuals outside the project team have substantially increased the project's scope.

EXG2 Individuals outside the project team have exerted substantial pressure to accelerate the project's completion.

EXG3 The priority of this project has been significantly increased since it was started.

EXG4 Job responsibilities outside this project have adversely impacted my ability to complete my work on this project as originally expected.

EXG5 Changes in the amount of financial resources available to this project have adversely impacted our ability to complete this project as originally expected.

EXG6 Changes in how development personnel have been allocated to this project have adversely impacted our ability to complete this project as originally expected.

EXG7 Unexpected events affecting our organization have adversely impacted our ability to complete this project as originally expected.

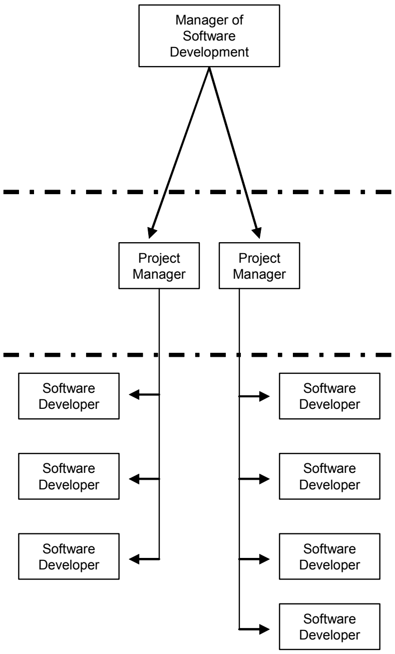
Chapter V: Revised Theory

5.1. Summary of Data Collection

Negotiations for data collection began in March of 2006 with companies from three separate sources: (1) thirteen companies that I had personal relationships with; (2) eighteen companies that were part of a research institute at a southwest university ; and (3) thirty-eight companies that expressed interest in this research resulting from a request for participation e-mail sent out by the leadership of the Project Management Institute’s special interest group on information systems (PMI-ISSIG) to their more than fifteen-thousand world-wide membership base. The negotiation process included a large number of on-site visits, phone conversations, and e-mail correspondence, most frequently resulting in several discussions with multiple individuals at each organization. The presentation included in “Appendix IV – Participation Sales Presentation” was used as a means of structuring the conversations and soliciting participation.

From the sixty-nine organizations that considered participation in this research, ten organizations decided to participate as of April, 2007, formalizing their decision by signing an organizational consent to participate form (Appendix I - Organizational Informed Consent Form). Per the data collection methods description provided in Chapter IV and outlined in Figure 5.1-1, I collected data in four phases from three different types of individuals within each of those organizations.

Figure 5.1-1: Data Collection Process



Data collection method: semi-structured interview & questionnaire

Data to be collected (semi-structured interview):

- Description of the project management practices
- Expectations of practice use in the development process
- Names and contact information for project managers

Data to be collected (questionnaire):

- # developers in the software development group
- Annual budget of the software development group
- Level of their project management professionalization
- Practice routinization
- Practice regulation (codification and management expectations)
- Names and contact information for project managers & their associated projects

Data collection method: questionnaire

Data to be collected:

- Project details (description, time allocated, # developers, % complete, budget and financial value, level of strategic and operational importance, level of complexity, level of innovativeness)
- Level of their project management professionalization
- Names and contact information for software developers on their project(s)

Data collection method: questionnaire

Data to be collected (time 1):

- Age / Gender
- # years development experience
- # years experience with the practice
- Level of project management professionalization
- Management support
- Practice complexity
- Practice voluntariness
- Practice relative advantage (instrumental and political)
- Practice social norms

Data to be collected (time 2):

- Operational exigency
- Practice usage (extent, ends conformity, and means conformity)

Table 5.1-1 provides a breakdown of the overall response rate, including details within each participating organization. Totals from the table indicate that 94% of the software development managers (17 of 18), 73% of the project managers (30 of 41), and 65% of the software developers (67 of 103) that were surveyed fully completed their participation. Since the ultimate unit of analysis in this research is the software developer, only organizations where at least one software developer completed the second survey could be included in the research. This resulted in seven organizations that provided potentially usable responses for this study and a total of 67 software developers that completed both surveys. Table 5.1-2 indicates that out of the 67 software developers, 56 were employees of the organization surveyed, while 11 were identified as contract workers.

Table 5.1-1: Participant Response Overview

Organization ID	Industry		Software Development Manager	Project Manager	Software Developer	Total Participants
Company 1	Insurance	# Surveyed	3	5	14	22
		# Complete Responses	3	3	9	15
Company 2	Insurance	# Surveyed	2	2	3	7
		# Complete Responses	2	1	0	3
Company 3	Energy	# Surveyed	1	2	0	3
		# Complete Responses	1	0	0	1
Company 4	Insurance	# Surveyed	1	6	12	19
		# Complete Responses	1	3	8	12
Company 5	Government	# Surveyed	1	2	6	9
		# Complete Responses	1	2	0	3
Company 6	Energy	# Surveyed	3	7	11	21
		# Complete Responses	2	7	7	16
Company 7	Information Technology	# Surveyed	1	2	6	9
		# Complete Responses	1	1	2	4
Company 8	Transportation	# Surveyed	4	7	28	39
		# Complete Responses	4	6	24	34
Company 9	Transportation	# Surveyed	1	3	9	13
		# Complete Responses	1	3	9	13
Company 10	Defense	# Surveyed	1	5	14	20
		# Complete Responses	1	4	8	13
		Total # Surveyed:	**18**	**41**	**103**	**162**
		Total # Complete Responses:	**17**	**30**	**67**	**114**

Table 5.1-2: Software Developer Employee Type Breakdown

Employee Type	Count
Full-time Employee	55
Part-time Employee	1
Contract Worker	11
Total:	**67**

To summarize, after approximately one year of intensive marketing and negotiations, responses were successfully obtained from 56 software developers who were working on eighteen distinct projects within six different companies (after removing responses from contract workers). Through extensive consultation with the dissertation chairs, a decision was made to complete the dissertation using the existing data set. However, this decision required that the structural model proposed in chapter 3 and summarized in Figure 3.1-1 had to be revised since the proposed relationships could be properly tested, given the sample size constraints that ensued. The remainder of Chapter V will discuss the revised research model and the theory underlying it.

5.2. Revised Research Model

The original research question being examined in this dissertation was as follows: ***What are the primary factors that influence the nature and extent of a software developer's use of a SDPM practice within a SD project?*** Regardless of the constraints that the sample size imposed on analytical techniques that could be employed, it was essential that the revised research model meet the dissertation's core goal of understanding factors that drive developer practice usage.

Addressing the extent of SDPM practice use, as originally intended, proved to be impossible given the data. Of the 103 responses received, 52 developers indicated that the SDPM practice was not used during the one month period between software developer surveys. While the original intention was to interpret these cases as a developer's decision not to use the SDPM practice, it became clear that no use could also represent a situation where the developer had no need to use the practice during the one month period. Since no data was collected regarding the necessity of use, the meaning of usage frequency could not be interpreted properly within this research.

Fortunately, a paramount concern in this research also involves understanding *how* the SDPM practice would be used, given an opportunity for usage. Two usage measures captured in this study, customized and faithful use, provide a means of addressing the *nature* of use in a software development context. Unlike frequency of use which focused on SDPM practice use during the month period between software developer surveys, both customized and faithful use addressed developer usage during the life of the development project up to the time of the second survey. This distinction allows for the examination of customization and faithfulness within the dissertation.

While practice customization provides the developer a means of using the SDPM practice more effectively within a project, faithful use provides a mechanism for enacting usage behavior that will be viewed favorably by management. Research has shown that the way in which software development methodologies are actually used is often contingent on developer needs and project requirements (Fitzgerald 1998), so it is important to investigate factors that will shape customized and faithful use of a SDPM practice that exists within the organizations development methodology.

To focus this research on factors influencing how the practice is used, the research question has been revised to question the factors that motivate the *nature* of usage. Specifically, the research question can be restated as the following: ***What are the primary factors that influence a software developer's customized and faithful use of a SDPM practice within a SD project?***

5.3. Research Hypotheses

The following paragraphs will present a revised set of research hypotheses (RH), addressing the impacts of developer beliefs and the institutional environment on both customized and faithful usage. First, the concept of institutional environment will be re-defined and described in relation to a software development context. Next, political and instrumental relative advantage will be discussed with respect to their influence on customized and faithful use. Arguments will then be proposed regarding the impact of relative advantage perceptions and the institutional environment on customized and faithful SDPM practice usage.

5.3.1. Defining the Institutional Environment

An individual's social environment is suggested to shape their attitudes and beliefs, as illustrated in the following quote (Salancik and Pfeffer 1978, p. 227):

> *"The social context has two general effects on attitude and need statements: (1) it provides a direct construction of meaning through guides to socially acceptable beliefs, attitudes and needs, and acceptable reasons for action; (2) it focuses an individual's attention on certain information, making that information more salient, and provides expectations concerning individual behavior and the logical consequences of such behavior."*

Behavioral research over the past several decades has demonstrated the importance of considering institutional factors when the individual being studied is operating within an organizational context (Bock et al. 2005; Hardgrave et al. 2003; Lewis et al. 2003; Purvis et al. 2001). Institutionalization, involving "processes by which social processes, obligations, or actualities come to take on a rule-like status in social thought and action" (Meyer and Rowan 1977, p. 341), provides a measure of organizational identity from which the employee draws inferences about acceptable behavior. Institutionalization acts to both enable and constrain individual behavior within the organization (Scott 2003). A key issue for this examination of software developer behavior, then, is to capture ways in which the institutional environment is manifested.

The original research model proposed in this dissertation drew on traditional views of institutional theory, where institutional pressure on an organization was suggested to arise from cultural-cognitive, regulative and normative sources external to the organization (Scott 2003). Within this perspective, institutions were most frequently pictured as external sources of influence in relation to the organization. While this perspective offered many important insights into the current research context, it failed to address institutional elements that exist and influence individual behavior within the organization (i.e., via socialization, power and coercive pressures). This presents a limitation to this work that considers how institutional pressures *emanating from the organization* impinge on workers *within the organization*. To address this limitation, an “institution-as-organization” view was utilized (Zucker 1987), where institutional elements are suggested to exist within the organization. In this perspective, the most interesting distinction within the institutional environment is not the conventional mimetic, regulatory, normative influences but rather the formal/informal distinction. As such, the following reframing of the institutional environment draws on this perspective to provide a means of explaining institutional influence on software developer behavior.

Referring to organizations as institutions, Zucker (1987, p. 446) noted that "institutional elements arise primarily from small group or organization-level processes". This idea regarding the source of institutional elements is consistent with perspectives on metastructuration, where formal and informal activities within the organization are suggested to “influence the structuring activities of technology users” (Orlikowski et al. 1995, p. 441). In a software development context, organization-level processes entail practice regulation while small group processes are instituted as social norms regarding SDPM practice use. Each of these sources of the institutional environment is discussed in detail in the following paragraphs.

5.3.1.1. Institutional Elements Arising from the Formal Organization - SDPM Practice Regulation

SDPM practice regulation emerges from the organization’s formal authority structure and provides a means of enforcing practice use within a development group. Regulative influence, referring to organizational structures that reward compliance for or penalize deviance from a specific behavior, provides a means of shaping developer behavior with regards to SDPM practice usage. Within a software development environment, two factors influence the strength of the regulative environment. First, developers are made aware of expectations for behavior by the way in which the practice has been formally documented and communicated. As researchers have noted, managerial expectations can not be evaluated unless they have been meaningfully codified within the development process (DeMarco and Lister 1999). As a means of collecting information on SDPM practice codification, the software development manager from each organization was presented with five questions detailing the degree to which the SDPM practice has been documented (detailed on p. 114). While codification speaks to the clarity of practice expectations, there is still question regarding the degree to which practice use is actually mandated and evaluated within the development group. Management expectations, defined as the extent to which SDPM practice use is incorporated into the software developer’s formal performance evaluation, was used to collect this information (detailed on p. 114).

Capturing the degree of practice documentation and management expectations for use provides a representation of SDPM practice regulation within the software development group. As such, responses from these two scales were combined into one composite variable called **SDPM practice regulation**, providing a means of assessing the degree to which practice use was regulated within the development group.

5.3.1.2. Institutional Elements Arising from the Informal Organization - Social Norms

Software development projects often require interdependent work within a group of individuals, creating an environment where peer influence can serve to influence practice use. As professionals, software developers are particularly susceptible to cognitive and normative pressures to facilitate the development of common standards of behavior and discourse necessary for coordinated action (DiMaggio and Powell 1983). Social pressure has been shown to be a clear means of encouraging developer behavior in a software development project environment (Hardgrave et al. 2003). The SDPM practice social norm construct, as discussed on p. 112, captures such informal homogenization of thought and behavior and is thus used to address the institutional environment aspect arising from small group interaction.

5.3.1.3. Institutional Environment Classification

Addressing both formal organizational and peer group sources of institutional elements provides a means for understanding the institutional environment as experienced by a software developer, and enables predictions for faithful and customized practice usage based on an individual's position within the institutional environment. Table 5.3-1 illustrates the classification scheme used to address the institutional environment in this research. While a classification scheme can sometimes oversimplify a phenomenon, it can also be useful to bound a domain and focus theory (Mintzberg 1978).

A two-by-two table was created using both the formal (practice regulation) and informal (social norms) dimensions of institutionalizing activities as the axes labels, each having low and high conditions. Environments where practice regulation and social norms are high have been termed "High Institutionalization". This environment is characterized by a SDPM practice that is (a) thoroughly documented, (b) defined as a part of the software developer's performance evaluation, and (c) considered by a software developer's peers to be important within the development process. It is generally expected that environments of High Institutionalization will directly impact a software developer's customization and faithful use of the SDPM practice and not interact with relative advantage perceptions. Situations low in practice regulation and social norms are labeled "Low Institutionalization" and characterized by an environment where the practice has not been well documented, usage is not required by management, and peers are not viewed as valuing the SDPM practice. Developers working in an environment characterized by Low Institutionalization of the SDPM practice will, in the absence of institutional pressure, be driven to shape usage based on their own perceptions and preferences.

Within an "Informal Institutionalized" environment, peers will view the practice as important for development projects but management will not define, demand or support its use. As such, software developers will be left to define usage behavior based on their own perceptions and the influence of the peer group. Finally, "Formal Institutionalization" exists where the practice regulation is high but social norms are low. In a software development context, Formal Institutionalization forces a software developer to consider organizational requirements for practice usage against personal beliefs, which if favorable can act as a means of compensating for a lack of peer support.

Importantly, this re-conceptualizing of institutional pressure within an organization provides two key advantages over the prior view utilized in this dissertation. First, the revised perspective is more consistent with institutional theory directed towards the individual working within an organization since it focuses on how institutional elements arise and are sustained within an organization. This perspective differs from traditional views of institutional theory which worked to explain environmental pressures which impacted the trajectory of an organization. Second, treating the institutional environment as a categorical variable in this research accommodates the limited sample size in this research by conserving degrees of freedom.

Table 5.3-1: Institutional Environment Classification

Formal Dimension: Practice Regulation	Informal Dimension: Social Norms	
	Low	High
High	Formal Institutionalization	High Institutionalization
Low	Low Institutionalization	Informal Institutionalization

5.3.1.4. Low Institutionalization and Usage Behavior

Before proceeding, it is important to address the predictability of behavior in environments where both practice regulation and social norms do not support the use of the SDPM practice. In these cases, the weak institutional environment provides little guidance regarding practice usage and the developer will have to rely on an internal motivation to determine how usage should be manifested within their project. At the organizational level, it has been noted that “when institutions are uncertain, and alternative ways for conforming to them are available, organizations will use the resulting discretion to pursue their pluralistic and strategic interests" (Goodrick and Salancik 1996, p. 5). Similarly, this is likely to be the case with the institutionalization of individuals' behavior within organizations. In a low institutionalization environment, usage behavior will be driven by factors beyond the institutional environment such as perceptions of job importance related to the behavior (Leonard-Barton and Deschamps 1988). This leads to the expectation that low institutionalization will not have a direct impact on either SDPM practice customization or faithfulness of use. Subsequently, no research hypotheses have been defined for the relationship between low institutionalization and either type of usage.

5.3.2. Relative Advantage

Perceptions of relative advantage have received strong support in research as a precursor to innovation adoption, both in terms of the degree to which the innovation offers a perceived improvement in performing a specific task (instrumental) and the degree to which the innovation is perceived to enhance an individual's image within a social setting (political) (Moore and Benbasat 1991; Rogers 1995). Further, research on technology adoption has consistently found both instrumental and political relative advantage to be salient predictors of adoption and post-adoption attitudes (Compeau et al. 1999;

Karahanna, Straub and Chervany 1999; Thompson et al. 1991). Questions remain, however, regarding the impact of instrumental and political relative advantage perceptions on the nature of usage exhibited by individuals.

Marketing research has found that consumers will consider an issue more thoughtfully when they perceive it as being personally relevant (Petty and Cacioppo 1984). Relative advantage perceptions of the SDPM practice provide a means of increasing the personal relevance of the practice, and as such encourage mindful consideration of the practice. Mindfulness, defined as "containing components of (a) openness to novelty; (b) alertness to distinction; (c) sensitivity to different contexts; (d) implicit, if not explicit, awareness of multiple perspectives; and (e) orientation in the present" (Sternberg 2000, p. 12), has frequently been found to positively impact individual creativity (Langer and Moldoveanu 2000) and provides a means of understanding how relative advantage perceptions might impact customized and faithful use.

Regarding instrumental relative advantage perceptions, individuals will be prompted to act mindfully regarding the task-related characteristics of the practice. Instrumental relative advantage, which is focused solely on the task utility of using a practice, presents a developer with the opportunity to reflect more deeply about the task-related qualities of the practice. Mindfully considering a SDPM practice's task-related characteristics allows a software developer to more thoroughly evaluate the consequences of practice usage, enabling them to consider novel ways of using the practice for a given project. It is appropriate, then, to suggest that instrumental relative advantage will be directly related to a developer's customization of the SDPM practice.

Perceptions of political relative advantage, on the other hand, provide developers an opportunity to consider the professional-related implications of practice use in greater detail. When political value is viewed as relevant, mindfulness will focus on ways in which practice use can be enacted to engender favor within the organization and not on task-related issues. At an organizational level, faithful compliance to popular management techniques has been shown to provide greater legitimacy within the institutional environment (Staw and Epstein 2000). In a similar way, usage of the SDPM practice that is consistent with the practice's spirit can provide a means of further enhancing one's image within the organization. This leads to the expectation that political relative advantage will have a direct impact on faithful use of the SDPM practice but no predictable influence for practice customization.

The following paragraphs will now draw on the previous conceptualizations of the institutional environment and relative advantage in order to propose conditions which encourage customized and faithful use.

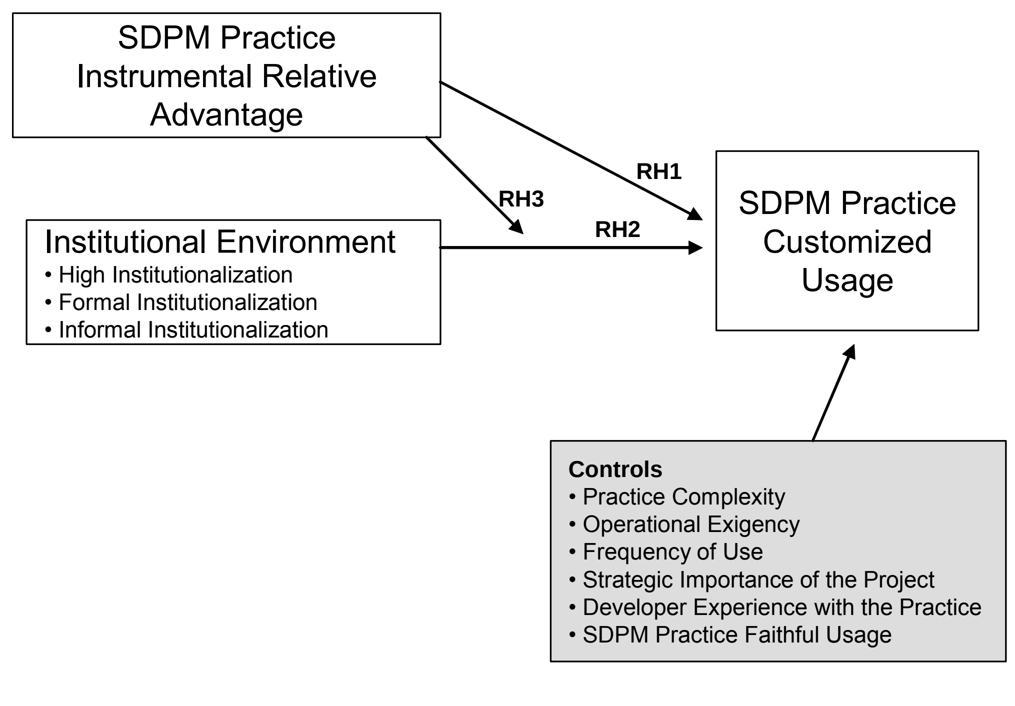
5.3.3. Customized Use

In my time as a programmer, I have seen many ways of doing programming, and some are more formal than others. I have also noticed that when programmers follow the "official" way, things often don't go very well. When things do go well, it is often because the programmers didn't follow the "official" way. (DeGrace and Stahl 1990)

Why would a software developer choose to customize a SDPM practice within a given SD project? While the SD literature is silent regarding this question, the management literature has something to say about the issue. The first perspective involves a proactive rationale, where "best practices" must be customized to the individual's specific context if they are to maximize the benefits of that behavior (Pascale and Sternin 2005). Second is a reactive rationale for customization, where adaptive organizational responses are most likely in response to dissonance between an employee's positive perceptions of the organization's identity and negative perceptions regarding the organization's image in the community (Dutton and Dukerich 1991). Both of these rationales illustrate that customization behavior involves action in response to a realized need. In this light, customized SDPM practice usage is expected to be driven primarily by issues that relate to the developer's desire to act in ways which will end in positive project returns.

Figure 5.3-1 provides a graphical view of the conditions posited as promoting customized use of the SPDM practice. In this model, customized usage is suggested to flow from two primary sources: (a) software developer beliefs regarding the instrumental relative advantage of the practice and (b) the institutional environment surrounding SDPM practice usage composed of both social norms and practice regulation. The following paragraphs will propose relationships between these two drivers of customized SDPM practice use by considering the amount of effort involved in appropriating the practice within an existing development project.

Figure 5.3-1: Revised Research Model (Customized Use)


SDPM Practice Instrumental Relative Advantage
Institutional Environment
• High Institutionalization
• Formal Institutionalization
• Informal Institutionalization
RH1
RH2
RH3
SDPM Practice Customized Usage
Controls
• Practice Complexity
• Operational Exigency
• Frequency of Use
• Strategic Importance of the Project
• Developer Experience with the Practice
• SDPM Practice Faithful Usage

5.3.3.1. SDPM Practice Instrumental Relative Advantage

Perceptions of instrumental relative advantage describe the relevance that a software developer has assigned to the task-related potentiality of a SDPM practice. In general, "as a given issue becomes increasingly personally relevant to a receiver, the receiver's motivation for engaging in thoughtful consideration of that issue presumably increases" (O'Keefe 2002, p. 141). Recent literature has discussed this as mindfulness, where an entity is driven to act based on thoughtful consideration of the behavior and its potential consequences (Fiol and O'Connor 2003). Mindful information processing has some important consequences, such as an increased willingness to consider new perspectives and undertake novel approaches to situational problems (Butler and Gray 2006).

As software developers come to believe a SDPM practice is increasingly valuable for a given task, they are prompted to consider it in greater detail with regards to the task. In turn, the developer is provided a means of evaluating opportunities for improvement within their current project and is more likely to modify the practice accordingly. This idea has been demonstrated in a study investigating the response of users to the introduction of new IT, finding that users who view a system as an opportunity to improve personal efficiency and effectiveness will increase the likelihood of undertaking adaptation activities with regards to the new IT (Beaudry and Pinsonneault 2005).

Further, the context of SDPM practices also suggests that instrumental relative advantage will encourage customized usage behavior. Practices are frequently used in response to specific problems within the development process, often requiring developers to reconsider the most appropriate means for addressing the issue (Goulielmos 2004). By perceiving the practice to be instrumentally valuable, the developer is more likely to apply it within the development process when problem situations arise. These arguments lead to revised hypothesis 1:

RH1: A software developer's instrumental relative advantage perceptions of the SDPM practice will increase their customized use of the practice.

5.3.3.2. Institutional Environment

Research has been divided on the impact of highly institutionalized environments on individual behavior. One school of thought suggests that a strong institutional environment will constrain creative behaviors, forcing individuals to act in a manner consistent with the institutional environment (Ford and Gioia 2000). This perspective is especially salient in decision domains where individuals are tasked with making a complex choice among competing ideas. Specifically, "as decision complexity increases, solutions become increasingly error prone, means become more important than ends, and rationalization replaces rationality" (Van de Ven 1986, p. 595), discouraging an individual from taking risks associated with deviance from institutionally accepted behavior. Another school of thought has suggested that creativity can actually be enhanced in institutional environments where a specific behavior has become routinized (Feldman and Pentland 2003), such as the finding that input, behavior and output control can each serve to enhance radical innovation within a pharmaceutical organization (Cardinal 2001). What might explain this seemingly counterintuitive explanation? One thought is that environments where behavior has been deeply institutionalized free an individual from having to focus energy on decisions regarding the behavior itself and in turn allows them to pay greater attention to the task being completed (Ohly, Sonnentag and Pluntke 2006). Greater attention to the task in turn allows an individual to consider and enact novel ways of improving task work.

While software development environments are often quite complex, requiring developers to consider a large number of alternatives in the design of systems, the decision regarding SDPM practice usage involves a fairly small domain and as such can be considered a low complexity decision environment. In highly institutionalized environments, developers will be freed from mental effort required in the practice usage decision process, allowing them to more fully focus on the project work at hand. Extra attention on the project work itself is expected to maximize the possibility that the practice will be seen as tractable.

Informal institutional environments are also expected to impact practice customization, although to a lesser degree. A weak regulative environment is manifested through (a) minimal documentation of how the practice should be used and (b) little evidence of how (or if) usage will be evaluated by management. As such, the presence of weak practice regulation alone does not provide guidance regarding the degree to which a developer will customize the SDPM practice. However, social norms supporting usage clearly serve as a means of encouraging actual usage behavior (Fulk 1993) and in the face of weak practice regulation provide a mechanism for encouraging practice customization. Environments with strong clan norms regarding a behavior provide a safe and non-threatening environment that encourage “the use of employees’ creative potential” (Baer and Frese 2003, p. 50). In these environments, software developers are presented the opportunity to experiment with practice structure and adapt it as needed for their project. It is thus expected that customized usage will exist when social pressures for practice use are strong and practice regulation is weak. However, the lack of regulative pressure present in this environment suggests that motivation for customization will be less severe than in highly institutional environments.

While environments with strong social norms present a means of encouraging customization of the SDPM practice, exclusively formal institutional environments can actually work to discourage customized use. Practice regulation that supports and enforces SDPM practice use encourages software developer compliance as a means of surviving within the organization. But the presence of weak social norms reduces the visibility of a practice within the developer's peer group and consequently limits attention to the details of practice use. This situation promotes mindless adherence to formal institutional pressures, where the individual can be described as acting on "automatic pilot" (Weick, Sutcliffe and Obstfeld 1999). Mindless behavior is characterized by "routine use of preexisting categorization schemes" (Butler and Gray 2006, p. 215), and as such reduces a developer's likelihood of enacting novel or adapted uses of the SDPM practice.

The preceding arguments suggest that different institutional environments will vary in their ability to shape customization behavior, as illustrated in the following hypothesis:

RH2: SDPM practice customization will be highest in environments of High Institutionalization, less in Informal Institutionalization, and least in Formal Institutionalization.

5.3.3.3. The Interaction between Instrumental *Relative Advantage and the Institutional Environment*

Instrumental relative advantage provides a developer with a rationale for investing cognitive effort concerning how the SDPM practice is used for a given project. Developers operating within institutional environments where social norms for use are strong are already likely to be cognitively engaged regarding the use of a practice for a given project. Therefore, the contribution of instrumental value perceptions is not expected to interact with the relationship between institutional environments of strong social norms (high institutionalization and informal institutionalization) and customized use. While the lack of structure in low institutionalized environments precludes positing a relationship with customized use, formal institutionalized environments should be able to benefit markedly when the developer perceives a practice to be instrumentally valuable.

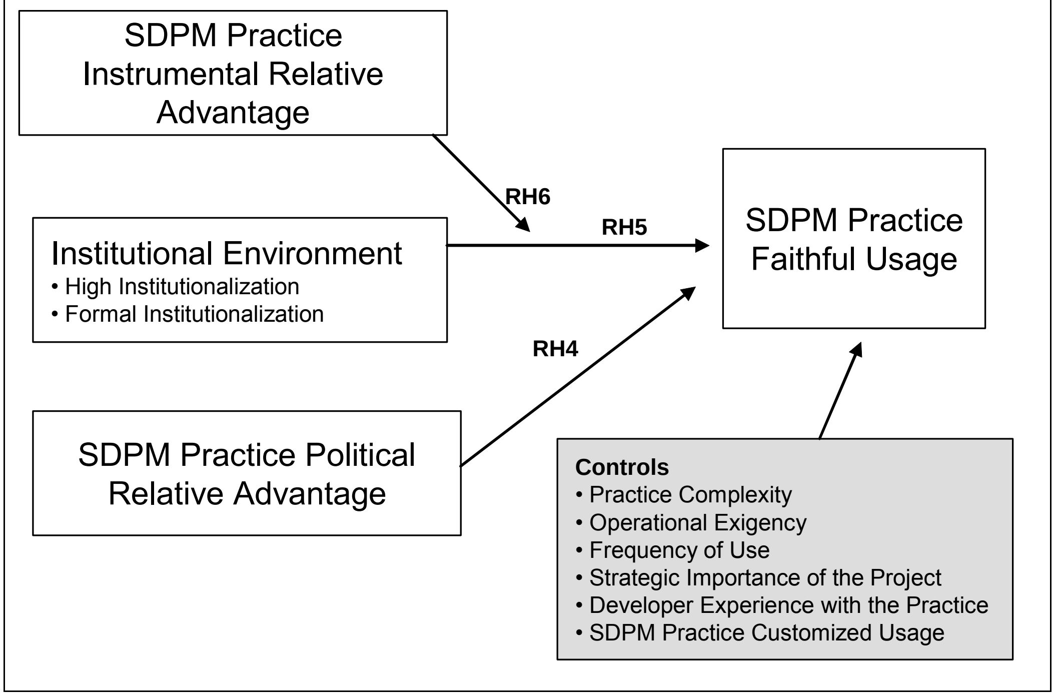
As hypothesized in RH2, customization of the SDPM practice is expected to be least in formal institutionalized environments. The effect of Formal Institutionalization on customization can be accentuated if an individual perceives personal value in using the practice for their project. Instrumental relative advantage works to create practice relevance in the mind of the developer, which in turn acts to encourage a thoughtful application of the practice (O'Keefe 2002). The attention to detail afforded by mindful consideration of an issue encourages novel solutions to routine problems (Levinthal and Rerup 2006), suggesting that instrumental relative advantage can provide a mechanism by which a developer is encouraged to act mindfully and, in turn, encourage customized use.

RH3: SDPM practice customization within Formal Institutionalization environments will be enhanced when a software developer perceives the SDPM practice to be instrumentally valuable.

5.3.4. Faithful Use

Faithful use, or the degree to which a software developer uses the SDPM practice in a manner consistent with the intentions of those individuals that created the practice, is also expected to be influenced by relative advantage perceptions and the institutional environment. Figure 5.3-2 provides a graphical illustration of the various factors expected to impact a developer's faithful use of the SDPM practice and an explanation of each proposed relationship is discussed in greater detail below.

Figure 5.3-2: Revised Research Model (Faithful Use)


SDPM Practice Instrumental Relative Advantage
Institutional Environment
• High Institutionalization
• Formal Institutionalization
SDPM Practice Political Relative Advantage
RH6
RH5
RH4
SDPM Practice Faithful Usage
Controls
• Practice Complexity
• Operational Exigency
• Frequency of Use
• Strategic Importance of the Project
• Developer Experience with the Practice
• SDPM Practice Customized Usage

5.3.4.1. Political Relative Advantage

As discussed earlier, perspectives on mindful behavior suggest that relative advantage perceptions can encourage a developer to thoughtfully consider practice usage. Importantly, political value perceptions of the SDPM practice should focus a developer’s mindful use of the practice around the image enhancing potential of practice use. A focus on image has specific implications for faithful use of the practice. While acting contrary to workplace norms can have both negative and positive implications for an employee (Warren 2003), deviance from expectations is professionally risky for an individual concerned with sustaining their image within the organization

(Rosenfeld, Giacalone and Riordan 2001). When individuals believe that a specific behavior is important in sustaining their image with others, they are more likely to utilize that behavior as a means of impression management (Leary and Kowalski 1990). By using a SDPM practice faithfully, software developers have a direct means of promoting their image within the organization.

A key issue in a software development context, then, is what group(s) serve(s) as the important reference points for the developer[8]. Specifically, a question exists regarding the degree to which a developer desires to sustain or enhance their image with management and/or peers in the development group. In situations where a developer wishes to promote their image with management, then faithful use of the practice can be expected. The impact on usage faithful to management expectations is less certain when peers serve as the primary referent point. Regardless of the referent group(s) utilized by a software developer, mindfulness perspectives suggest that elevated political value perceptions will encourage greater attention to the political consequences of practice use. As such, a politically mindful consideration of the SDPM practice is expected to increase the occurrence faithful use.

[8] While software developers in this research were led to address faithful usage in terms of "management in our software developer group", political relative advantage questions focused on image "within my organization" and as such allowed developers to select their own referent group.

RH4: A software developer’s political relative advantage perceptions of the SDPM practice will increase their faithful use of the practice.

5.3.4.2. Institutional Environment

Highly institutionalized environments are characterized by strong practice regulation and social norms. Within this type of environment, "gaps" permitting self-interested action are few and far between. Specifically, “when the standards for activities are certain, which means players agree on what is appropriate to do, practice is completely determined…either alternative practices will be unimaginable or engaging in them will undercut the legitimacy organizations need to operate” (Goodrick and Salancik 1996, p. 3). The most likely response to a strong institutional environment is acquiescence (Oliver 1991), as demonstrated in the conforming behavior of organizations who were late adopters of TQM initiatives (Westphal et al. 1997). Within a software development context, this suggests developers facing strong institutional pressures will have little choice but to acquiesce to those demands.

On the other hand, Formal Institutionalization environments are characterized by strong practice regulation stemming from the formal organization and weak social norms ensuing from the informal organization. In this context, developers are presented a clear picture of what practice use should look like and how it will be evaluated, but also understand that practice use is not valued by referent others. While a strong regulative environment serves to encourage a developer to follow the “correct” behavior during project work, weak social norms can detract from a developer’s desire to use the practice (Hardgrave et al. 2003). This situation can promote a very shallow use of the practice, or mindless use characterized by “routine use of preexisting categorization schemes” (Butler and Gray 2006, p. 215). While the regulative environment will produce SDPM practice usage, the intensity of faithfulness will be less than is expected in High Institutionalized environments.

RH5: SDPM practice faithful use will be higher in High Institutionalization environments than in Formal Institutionalization environments.

In cases of Informal Institutionalization, social norms are expected to shape a developer’s willingness to use the practice faithfully. This suggestion is consistent with organizational literature observing that organizations are more likely to adopt institutionally contested practices when other respected institutional environments consider the practice to be legitimate (Sanders and Tuschke 2007). Further, this assertion agrees with literature discussing the importance of clan control on individual behavior (Ouchi 1979). In order to predict the impact of an informal institutional environment on faithful SDPM practice use, clan expectations for how the SDPM practice should be used must be considered. Unfortunately, social norms as operationalized in this study do not clarify what peers expect proper usage to look like but rather only provide a snapshot of general expectations for use. This being the case, no relationship can be proposed between an informal institutional environment and faithful SDPM practice use.

5.3.4.3. The Interaction between Relative Advantage and Institutional Environment

As discussed earlier, Low Institutionalization environments provide little guidance regarding faithful use and Informal Institutionalization environments on faithful use can not be hypothesized in this dissertation because data was not captured regarding the type of usage expected by the peer group. Formal Institutional environments, however, do have certain impacts on faithful use, which can be moderated by developer beliefs.

In considering factors which motivate the use of systems development methodologies, researchers have suggested that environments where methodology use is highly regulated will enhance the relationship between social norms and methodology use (Iivari and Huisman 2007). This suggests a strong regulative structure can serve to compensate for a lack of peer support and encourage usage that is consistent with organizational expectations. However, clan norms that do not support practice use will create mental conflict, forcing the developer to face a difficult decision regarding faithful compliance with expectations.

This conflicting mental situation has been characterized in literature as cognitive dissonance and is suggested to be an uncomfortable and thus undesirable state in which individuals often exist. When confronted with situations of cognitive dissonance, “the reality that impinges on a person will exert pressures in the direction of bringing the appropriate cognitive elements into correspondence with that reality” (Festinger 1962, p. 11). The theory of cognitive dissonance as such offers a perspective that proposes that people desire mental consonance, and will adjust cognitions in ways necessary to bring the conflict into balance.

An individual can manipulate internal beliefs to counteract an inconsistent set of cognitions, such as a smoker who believes that weight loss afforded by smoking outweighs other potential health risks (Festinger 1962). One such internal belief within a software development environment relates to a developer’s belief that the practice is valuable for completing project tasks. By placing greater value on the task-related value of a practice, a developer is prompted to re-frame peer expectations, believing that the institutional environment would prefer them to act in a manner that optimizes organizational outcomes rather than reproduce institutional rules and compromise organizational outcomes. This leads to the assertion that instrumental practice value perceptions are expected to attenuate the positive relationship between formal institutional environments and faithful practice use.

RH6: The positive relationship between Formal Institutionalization environments and SDPM practice faithful use will be attenuated by a software developer's instrumental relative advantage perceptions of the SDPM practice.

5.4. Control Variables

Changes to the research model also required a rethinking of the controls that should be used in a study of practice use. Controls of importance are driven by the individual and project context, and are discussed in greater detail below.

5.4.1. SDPM Practice Complexity

The first control included is SDPM practice complexity, or the developer's perception concerning the degree of difficulty associated with the understanding and use of the SDPM practice for a particular project. While complexity has long been understood to be an important predictor of individual adoption (Fishbein and Ajzen 1975; Rogers 1995), research on post-adoptive behavior has suggested that usage is characterized more as a reflexive response to a specific work situation

(Jasperson, Carter and Zmud 2005). This suggests that, while important to consider, complexity is not a primary variable of interest when examining a post-adoptive phenomenon such as the nature of SDPM practice usage. However, complexity is still an important variable and as such is included as a control in this study. Complexity perceptions have frequently been suggested to create a negative drag on individual adoption of an innovation (Rogers 1995), and is subsequently expected to detract from a developer's customization and faithful use of the SDPM practice.

5.4.2. Operational Exigency

The second control discussed in this research is operational exigency, or the degree of urgency exerted within a given project by external forces. Exigencies work within a project to focus on short term needs which can detract from behaviors driven by long-term benefits such as exhibited in the use of SDPM practices. This refocus of activities has been discussed in research as a "speed trap", where individuals are forced to make a decision right now at the expense of making the right decision now (Perlow et al. 2002). When individuals are presented with significant pressure in a decision-making episode, individuals have been shown to adapt by focusing on a limited set of "important" attributes (Edland 1994; Payne et al. 1996). Since operational exigencies within a project have the effect of inducing pressure on the developer, it is expected that they will serve to reduce customization behavior and encourage faithful usage of the practice.

5.4.3. Frequency of Use

The third control utilized in this research is a developer's frequency of SDPM practice use between the two developer surveys. The original intention of this research was to also study usage frequency as a dependent variable, but I failed to acknowledge the fact that situations often exist when a practice might not be needed for a specific project (and thus not be used). In these situations, it would be impossible to determine whether non-usage was driven by beliefs and the institutional environment or just because the practice was not needed.

In examining the nature of usage, though, we must still consider if using the practice was needed for the given project. No data was collected to directly capture this issue, but the frequency of use data can be used as a proxy. By transforming the variable into a binary representation of usage (0=not used, 1=used), we can provide a means of understanding if the practice was needed for the given project. While this is not a perfect proxy, it does allow us to ensure that the act of using the practice does not mask what is truly happening within the decision process. It is expected that, given the need for practice use, developers will be more likely to customize the practice and use it in a manner which is faithful.

5.4.4. Strategic Importance of the Project

Another important control relates to the importance of the project in which the developer is working. Research has demonstrated that developer behavior to use methodologies can be heavily impacted by the importance of the project (Fitzgerald 1998), and the same is expected for the use of practices which exist within a development methodology. As such, strategic importance of the project, defined as the degree to which a project is considered valuable by management in comparison to other software development projects in the organization, is expected to positively influence both customization and faithfulness.

5.4.5. Developer Experience with the SDPM Practice

The role of developer experience in exhibiting usage behavior is also an important control for this study. Experience has often been shown as an important factor shaping individual perceptions (Xiao and Benbasat 2007). Behavioral research has suggested that "as individuals gain experience with what was initially a novel behavior, they tend to engage less frequently in reflective consideration of this behavior and rely instead on previous patterns of behavior to direct future behaviors" (Jasperson et al. 2005, p. 542). Research on development methodology usage behavior has demonstrated mixed results for the direction of experience on usage, but has generally agreed that experience often encourages a developer to (a) question and, if necessary, modify it based on explicit needs while also (b) accepting the methodology once benefits are realized (Fitzgerald 1997). This suggests that developer experience will positively influence customization and faithful use. A measure (in years) was included in this study to collect data on each developer's experience with the SDPM practice.

5.4.6. SDPM Practice Usage Type

Finally, the two research models presented in chapter 5 (Figure 5.3-1 and Figure 5.3-2) will be examined separately. It is expected that customization and faithfulness, while not ends on a continuum, will be inversely related to each other. To clarify the results for each usage type, the following analysis will also include the non dependent variable usage type as a control. For example, the analysis for customized usage will include faithful use as a control variable.

5.5. Conclusion

The preceding sections have outlined a revised set of research hypotheses regarding belief and institutional environment drivers of faithful and customized SDPM practice use. Table 5.5-1 provides an overview of the revised hypotheses proposed in this research which will be examined in the following chapter.

Table 5.5-1: Summary of Revised Dissertation Hypotheses

RH1	A software developer's instrumental relative advantage perceptions of the SDPM practice will increase their customized use of the practice.
RH2	SDPM practice customization will be highest in environments of High Institutionalization, less in Informal Institutionalization, and least in Formal Institutionalization.
RH3	SDPM practice customization within Formal Institutionalization environments will be enhanced when a software developer perceives the SDPM practice to be instrumentally valuable.
RH4	A software developer's political relative advantage perceptions of the SDPM practice will increase their faithful use of the practice.
RH5	SDPM practice faithful use will be higher in High Institutionalization environments than in Formal Institutionalization environments.
RH6	The positive relationship between Formal Institutionalization environments and SDPM practice faithful use will be attenuated by a software developer's instrumental relative advantage perceptions of the SDPM practice.

Chapter VI: Analysis and Results

6.1. Revisions to the Research Methodology

As discussed in Chapter 5, sample size constraints required that we revise the research agenda for this dissertation. The relationships proposed in the previous chapter rely on some changes to the methodology discussed previously in Chapter 4. The following paragraphs will provide a description and justification for the modifications required to test research hypotheses.

6.2. Descriptive Statistics

Totals from Table 5.1-1 demonstrate that 94% of software development managers (17 of 18), 73% of project managers (30 of 41), and 65% of software developers (67 of 103) that were surveyed fully completed their participation in this research. Table 5.1-2 illustrates that out of the sixty-seven software developers, fifty-six were identified as organizational employees while eleven were contract workers. A closer examination of the fifty-six responses revealed one developer who had entered the same answer for all survey questions. After removing this data, we were left with fifty-five usable responses from software developers within six organizations as illustrated in Table 6.2-1.

Overall, eleven managers from six organizations agreed to allow the participation of their software developers in this research. A total of eighteen projects were included in the study with an average of three software developers responding per project. Software developers responding to this survey on average were over forty years of age, had more than fourteen years of experience developing software, and were primarily male (71%). While scope change control practices were used in all participating development groups, two managers reported that structured walkthrough practices were not used in their organization. This resulted in all fifty-five software developers responding to questions concerning the use of a scope change control practice but only forty-eight of those same developers who also answered questions concerning the use of a structured walkthrough practice. Table 6.2-2 further details survey responses to understand more about developer information within each organization. Table 6.2-3 contains the specifics of each project to demonstrate characteristics within each organizational project.

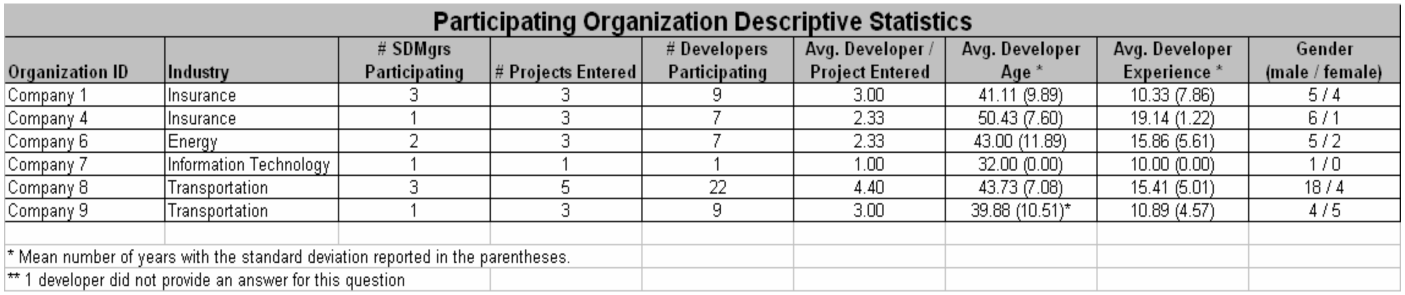
Table 6.2-1: Sample Descriptive Statistics

Sample Descriptives		
# Organizations		6
# SDMgrs Participating		11
# Projects Entered		18
# Developers Participating		55
Avg. Developer Age *		43.28 (9.14) **
Avg. Development Experience *		14.27 (5.89)
Gender		
	Male	39
	Female	16
SDPM Practice Responses		
	Scope Change Control	55
	Structured Walkthrough	48

* Mean number of years with the standard deviation reported in the parentheses.

** 1 developer did not provide an answer for this question

Table 6.2-2: Participating Organization Descriptive Statistics



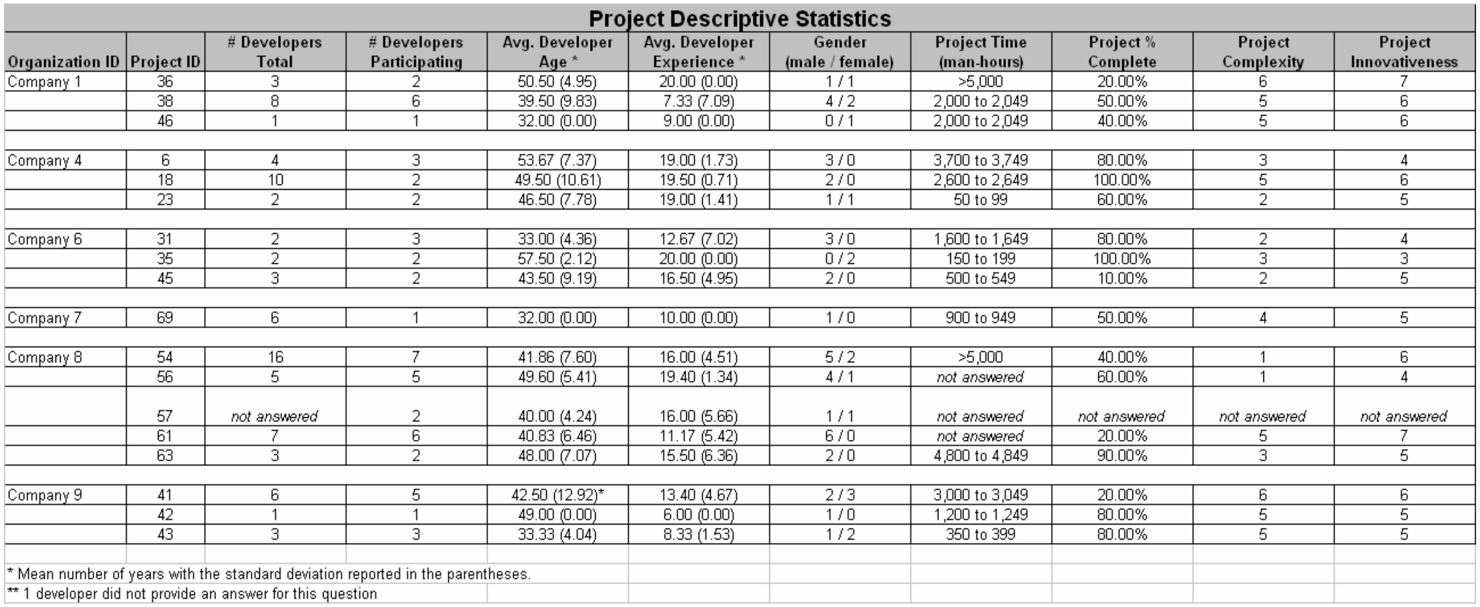
Participating Organization Descriptive Statistics

Organization ID	Industry	# SDMgrs Participating	# Projects Entered	# Developers Participating	Avg. Developer / Project Entered	Avg. Developer Age *	Avg. Developer Experience *	Gender (male / female)
Company 1	Insurance	3	3	9	3.00	41.11 (9.89)	10.33 (7.86)	5 / 4
Company 4	Insurance	1	3	7	2.33	50.43 (7.60)	19.14 (1.22)	6 / 1
Company 6	Energy	2	3	7	2.33	43.00 (11.89)	15.86 (5.61)	5 / 2
Company 7	Information Technology	1	1	1	1.00	32.00 (0.00)	10.00 (0.00)	1 / 0
Company 8	Transportation	3	5	22	4.40	43.73 (7.08)	15.41 (5.01)	18 / 4
Company 9	Transportation	1	3	9	3.00	39.88 (10.51)*	10.89 (4.57)	4 / 5

* Mean number of years with the standard deviation reported in the parentheses.

** 1 developer did not provide an answer for this question

Table 6.2-3: Participating Project Descriptive Statistics



Project Descriptive Statistics

Organization ID	Project ID	# Developers Total	# Developers Participating	Avg. Developer Age *	Avg. Developer Experience *	Gender (male / female)	Project Time (man-hours)	Project % Complete	Project Complexity	Project Innovativeness
Company 1	36	3	2	50.50 (4.95)	20.00 (0.00)	1 / 1	>5,000	20.00%	6	7
	38	8	6	39.50 (9.83)	7.33 (7.09)	4 / 2	2,000 to 2,049	50.00%	5	6
	46	1	1	32.00 (0.00)	9.00 (0.00)	0 / 1	2,000 to 2,049	40.00%	5	6
Company 4	6	4	3	53.67 (7.37)	19.00 (1.73)	3 / 0	3,700 to 3,749	80.00%	3	4
	18	10	2	49.50 (10.61)	19.50 (0.71)	2 / 0	2,600 to 2,649	100.00%	5	6
	23	2	2	46.50 (7.78)	19.00 (1.41)	1 / 1	50 to 99	60.00%	2	5
Company 6	31	2	3	33.00 (4.36)	12.67 (7.02)	3 / 0	1,600 to 1,649	80.00%	2	4
	35	2	2	57.50 (2.12)	20.00 (0.00)	0 / 2	150 to 199	100.00%	3	3
	45	3	2	43.50 (9.19)	16.50 (4.95)	2 / 0	500 to 549	10.00%	2	5
Company 7	69	6	1	32.00 (0.00)	10.00 (0.00)	1 / 0	900 to 949	50.00%	4	5
Company 8	54	16	7	41.86 (7.60)	16.00 (4.51)	5 / 2	>5,000	40.00%	1	6
	56	5	5	49.60 (5.41)	19.40 (1.34)	4 / 1	*not answered*	60.00%	1	4
	57	*not answered*	2	40.00 (4.24)	16.00 (5.66)	1 / 1	*not answered*	*not answered*	*not answered*	*not answered*
	61	7	6	40.83 (6.46)	11.17 (5.42)	6 / 0	*not answered*	20.00%	5	7
	63	3	2	48.00 (7.07)	15.50 (6.36)	2 / 0	4,800 to 4,849	90.00%	3	5
Company 9	41	6	5	42.50 (12.92)*	13.40 (4.67)	2 / 3	3,000 to 3,049	20.00%	6	6
	42	1	1	49.00 (0.00)	6.00 (0.00)	1 / 0	1,200 to 1,249	80.00%	5	5
	43	3	3	33.33 (4.04)	8.33 (1.53)	1 / 2	350 to 399	80.00%	5	5

* Mean number of years with the standard deviation reported in the parentheses.

** 1 developer did not provide an answer for this question

6.3. Sample Size

This study evaluated the use of two specific SDPM practices, scope change control and structured walkthrough, within each software development organization. All fifty-five software developers responded to questions relating to the scope change control process while forty-eight answered questions about their structured walkthrough process. This resulted in 103 total responses regarding SDPM practice use coming from 55 software developers. The sample size utilized for the remaining analysis in this chapter is 103[9].

It is important to discuss the implications of this decision before proceeding. The primary concern when including both sets of SDPM practice responses in one analysis is the inherent non-independence of data. For example, consider the two survey questions below:

- I think the procedures associated with our ***scope change control*** practice are clear and understandable.
- I think the procedures associated with our ***structured walkthrough*** practice are clear and understandable.

The above questions are phrased identically except in their reference to the SDPM practice of interest. Non-independence of

[9] While sample size constraints led to the use of scope change control and structured walkthrough data in the same analysis, it is recognized that this is a less than optimal approach. Since the expectation of this dissertation is to publish findings, efforts to collect data are still in progress and SDPM practice data will be analyzed separately once an adequate sample size has been achieved.

responses can be problematic since a developer is unintentionally encouraged to answer both questions in a similar way. In order to proactively address issues related to non-independence of data, three structures were employed in this study to minimize the potential impact. First, the two SDPM practices evaluated in this research are very different in both why and how they are used in the development process. While both SDPM practices are controls in the development process, the software developer's relationship to and involvement with each is different. Scope change control is used in response to requested changes within the project, and is often implemented through the use of written documentation provided to an external person or committee. A structured walkthrough process, on the other hand, is typically part of the development process and is implemented directly by the software developer. So while the developer most often passes the scope change control documentation to someone else for action, they are intimately involved in the implementation and realization of the structured walkthrough process. Utilizing practices which are very different from each other provides a means for each respondent to divorce their thoughts from one practice when answering questions in relation to the other.

Second, questions for each practice were placed together to further break the connection between responses. As illustrated in

Appendices VIII and IX, software developers were first asked questions about their perceptions of the structured walkthrough process. Only after these questions had been answered were the software developers permitted to move to questions in relation to the scope change control process. An exception to this occurred with developers who worked in environments where the structured walkthrough process was not utilized. For these seven developers, they were only presented questions relating to the scope change control process.

Finally, the two SDPM practices are unrelated in their execution during the software development process. A software developer's decision to perform a structured walkthrough is not tied to their decision to engage in a scope change control process. Since the two activities are independent for a software developer, responses regarding these activities are likely to be independent as well.

While these three approaches may have reduced non-independence, they do not eliminate it as a threat in the analysis of data and interpretation of findings. The most forceful way to overcome this limitation is to increase the sample size and analyze each SDPM practice data set independently. Post-dissertation data collection efforts are expected to increase the sample size such that each SDPM practice can be analyzed separately.

To restate, the analysis below will utilize 103 responses from fifty-five software developers.

6.4. Measurement Validation

A key factor in establishing the usefulness of a measure is the degree to which the construct is considered valid and reliable (Shadish et al. 2002). As such, the validity and reliability of measures utilized in this study must be examined before the proposed relationships can be statistically evaluated. Validity, or “the extent to which a measure or set of measures correctly represents the concept of study” (Hair, Anderson, Tatham and Black 1998, p. 118), is different than reliability which speaks to the consistency of equivalence and/or stability for a measure. A commonly accepted approach for evaluating the appropriateness of measures is to first establish unidimensionality (a type of validity) and then proceed to evaluating consistency of equivalence (a type of reliability) for each composite measure (Gerbing and Anderson 1988). The paragraphs below will address validity and reliability by evaluating first items answered by software developers (instrumental relative advantage, political relative advantage, social norms, complexity, operational exigency, customized use and faithful use) and then those completed by software development managers (management expectations, practice codification and practice routinization). As detailed in Chapter 4, these various measures were either (a) modified from previously validated scales or (b) developed and vetted over time through discussions with experts in the field of IS research and several from the software development profession.

6.4.1. Validation of Software Developer Constructs

Concerning unidimensionality, exploratory factor analysis (EFA) can be utilized to evaluate how well data meets a theoretically-derived structure. Several issues must be addressed in order to get meaningful results from a factor analysis (Ford, MacCallum and Tait 1986). The first concern relates to the sample size required to get meaningful results. Researchers have found that when the average communalities for all items are greater than 0.70 "accurate estimates of population parameters can be obtained with samples as small as 100" (Fabrigar, Wegener, MacCallum and Strahan 1999). The average communalities for items in this factor analysis are .76, so a sample size of 103 seems sufficient. A second issue relates to the extraction method utilized. Because items examined in this study are perceptual in nature, the decision was made to use principle axis factoring because of its conservative approach (Nunnally and Bernstein 1994). Third, the solution must often be rotated in some manner to simplify the process of interpretation. Since the most conservative approach is to assume that factors of interest will be somewhat correlated (especially since factors are perceptual measures in this study), an oblique rotation

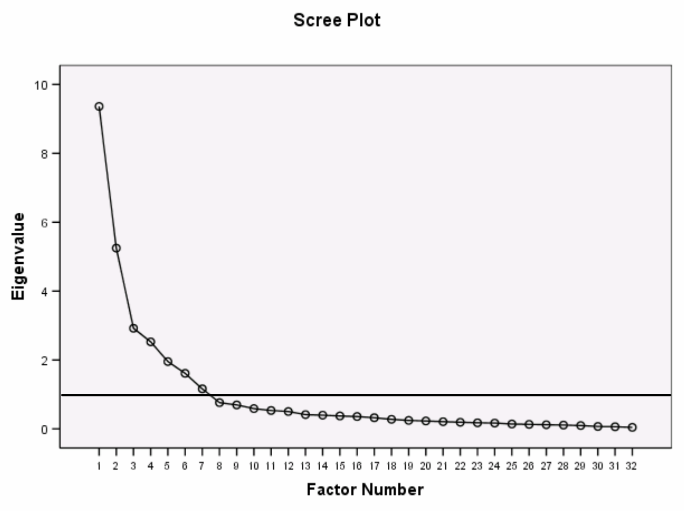
(Direct Oblimin) was utilized for the factor analysis (Hair et al. 1998). Finally, one must determine the number of factors to accept in the analysis. While the specification of constructs in this study anticipates that a factor analysis will find seven factors, a decision was made to allow the procedure to find the appropriate number of factors as a means of verifying the discriminant validity of constructs. Factors that demonstrated an Eigenvalue > 1 were considered for evaluation (Gorsuch 1983).

Table 6.4-1 illustrates that seven factors were extracted with the Eigenvalue>1 criterion, collectively accounting for more than 77% of the total variance which exceeds a commonly accepted minimum of 60% for social science research (Hair et al. 1998). The scree plot presented in Table 6.4-2 visually demonstrates that factors with an Eigenvalue < 1 contribute very little to the overall variance explained, further justifying the representation of only seven factors (Cattell 1978).

Table 6.4-1: Factor Analysis (Total Variance Explained)

Factor	Eigenvalue	% of Variance	Cumulative %
1	8.77	28.28%	28.28%
2	5.22	16.84%	45.11%
3	2.89	9.32%	54.43%
4	2.53	8.15%	62.58%
5	1.95	6.29%	68.87%
6	1.59	5.14%	74.01%
7	1.12	3.61%	77.62%

Table 6.4-2: Factor Analysis (Scree Plot)



To interpret the pattern matrix, the procedure recommended by Hair, Anderson, Tatham and Black (1998) was followed. The first step was to select the factors for consideration. The initial solution converged in fifteen iterations, producing eight factors with eigenvalues greater than 1, a finding which was inconsistent with the expectation of seven constructs. The second step was to mark all loadings which did not fall outside the -0.50 and 0.50 range. Importantly, the factor matrix revealed that, while no items loaded on more than one factor, two items failed to load on any factor and one loaded alone on the eighth factor. Because loadings were not perfect, the third step was to identify variables with a communality of less than 0.50 to be considered for deletion. The eighth factor contained only one item, EXG3 ("The priority of this project has been significantly increased since it was started”), that had a communality of 0.359. Investigation of this item revealed that rather than focusing on operational exigencies relating to the project, the question emphasized exigencies relating directly to the individual. Since there was a theoretical rationale for removing this item from the analysis, EXG3 was removed and steps 1 – 3 were conducted again. In all, the factor analysis was run seven times before a final structure was deemed acceptable. Table 6.4-3 provides an overview of the seven runs and the rationale for removing specific items from the analysis.

Table 6.4-3: Factor Analysis (Items Removed from the Analysis)

Time Run	# Factors Produced	# Items Not Loading on any Factor (<0.50)	# Items Loading on Wrong Factor	# Items Double Loading	Item Removed from Previous Run	Item Communality	Item Text	Justification for Deletion
1	8	2	1	0				
2	7	2	0	0	EXG3	0.359	The priority of this project has been significantly increased since it was started.	Rather than focusing on operational exigencies relating to the project, the question emphasized exigencies relating directly to the individual
3	7	2	0	0	CUST3	0.306	I have adapted our (scope change control / structured walkthrough) practice for this particular project.	All other CUST items reference adaptation with regards to meeting "my needs" on the project, while this one refers to adaptation in general.
4	7	2	0	0	EXG5	0.395	Changes in the amount of financial resources avaialble to this project have adversely impacted my ability to complete work on this project as originally scheduled.	Rather than focusing on operational exigencies relating to the project, the question emphasized financial constraints which are often removed from the software developer.
5	7	1	0	0	RAI4	0.624	Our (scope change control / structured walkthrough) practice is useful for this project.	Items for RAI were adapted from well established measures relative advantage, but I failed to include a reference to the person (useful to me as opposed to generally useful) on this question.
6	7	1	0	0	EXG1	0.459	Individuals outside the project team have substantially increased the project's scope.	Rather than focusing on operational exigencies relating to project work, this question focused on changes to scope which are uncertain to impact the developer's ability to complete project work.
7	7	1	0	0	RAI5	0.725	Our (scope change control / walkthrough) practice helps me successfully complete tasks within this project.	Unlike the other RAI items which focus on personal productivity, performance, or quality of work, this question focuses on the successful completion of tasks in general.

As a final step in the factor analysis process, the factors were renamed according to the construct as defined in this research. The final pattern matrix is pictured in Table 6.4-4. Factor loadings correspond to construct operationalizations presented in earlier chapters, with each item having a loading of greater than 0.50 on the expected factor. The one exception to this was SN3 ("Co-workers whose opinion I value think that using our (scope change control / structured walkthrough) practice is important when working on this project."), which had a factor loading of 0.472 on the factor labeled social norms. While this loading is low relative to the other social norm items, it was retained since the wording was consistent with the well validated form of the social norm construct employed in adoption research and was not found to be substantially different from the other social norm items.

Table 6.4-4: Factor Analysis (Factor Loadings)

	SN	CUST	EXG	CPLX	RAP	FAITH	RAI
CPLX1	0.028	-0.188	0.128	**0.792**	0.107	0.110	0.026
CPLX2	-0.043	0.093	-0.111	**0.775**	-0.020	-0.102	0.188
CPLX3	0.002	-0.001	0.026	**0.920**	-0.031	0.029	0.065
CPLX4	0.042	0.015	-0.006	**0.837**	0.019	0.017	0.068
CPLX5	0.183	-0.037	-0.052	**0.778**	0.088	-0.048	-0.132
RAI1	0.088	0.121	0.045	0.062	0.096	-0.077	**0.800**
RAI2	0.070	0.001	-0.050	0.046	-0.033	-0.135	**0.751**
RAI3	0.114	0.000	0.095	0.248	0.159	-0.034	**0.575**
RAP1	0.136	0.107	0.063	-0.045	**0.723**	-0.033	0.064
RAP2	0.107	-0.109	-0.059	0.094	**0.817**	0.077	-0.061
RAP3	-0.036	0.051	-0.073	0.006	**0.861**	0.009	0.055
RAP4	0.182	-0.084	0.013	-0.066	**0.778**	0.137	0.155
RAP5	-0.093	0.125	0.027	0.191	**0.642**	-0.176	-0.061
SN1	**0.785**	-0.012	-0.099	0.122	0.023	0.040	0.031
SN2	**0.856**	-0.011	-0.018	-0.017	0.154	0.012	-0.001
SN3	**0.472**	0.017	-0.024	0.186	-0.072	-0.110	0.251
SN4	**0.564**	0.061	0.043	0.104	-0.009	-0.150	0.275
SN5	**0.777**	0.016	0.039	-0.034	0.126	-0.134	0.017
EXG2	0.087	0.123	**0.704**	0.217	-0.106	-0.126	-0.221
EXG4	-0.142	0.031	**0.668**	-0.055	0.049	0.103	0.151
EXG6	0.220	-0.050	**0.622**	-0.269	-0.183	0.024	0.029
EXG7	-0.136	-0.050	**0.929**	0.043	0.137	0.060	0.027
CUST1	-0.001	**0.833**	-0.006	-0.039	0.077	0.025	0.000
CUST2	0.004	**0.841**	0.075	0.038	-0.130	0.142	-0.010
CUST4	-0.065	**0.724**	-0.038	-0.059	0.173	-0.004	0.012
CUST5	0.110	**0.794**	0.014	-0.025	-0.115	0.162	0.064
FAITH1	0.067	0.058	0.035	-0.020	-0.033	**0.876**	-0.077
FAITH2	-0.047	0.125	0.040	0.017	0.026	**0.796**	-0.010
FAITH3	0.044	0.026	0.032	0.022	0.021	**0.922**	-0.102
FAITH4	-0.059	0.111	0.008	0.024	0.113	**0.651**	0.008
FAITH5	-0.094	0.002	-0.052	0.026	-0.095	**0.791**	0.013

RAI = Instrumental Relative Advantage, RAP = Political Relative Advantage, CPLX = Practice Complexity, SN = Social Norms, CUST = Customized Use, FAITH = Faithful Use, EXG = Operational Exigency

The next step in verifying the appropriateness of constructs is to ensure discriminant and convergent validity. In order to accomplish this, summed scales were created for each factor where items clarified in both EFA procedures were combined and then averaged to create the new composite score. For example, each software developer was assigned a new SDPM practice complexity score (CPLX) based on the average of their answers to the five complexity items (CPLX1 through CPLX5).

Discriminant validity, or "the degree to which two conceptually similar concepts are distinct" (Hair et al. 1998, p. 118), can be evaluated by examining the loading of items on different factors in Table 6.4-4 and by comparing the square root of the average variance extracted[10] (AVE) of each construct to the correlation of compared constructs (Fornell and Larcker 1981). As the table illustrates, no items loaded on more than one factor providing evidence that measures within each factor could be discriminated from measures of other factors. Further, correlation values between each pair of constructs (reported in Table 6.4-5) were shown to be less than the square root of the AVE scores of those paired constructs. These two results support the divergent validity of constructs.

Next, convergent validity, or "the degree to which two measures of the same concept are correlated" (Hair et al. 1998, p. 118), must be examined. Convergent validity can be evaluated by examining the internal consistency of a measure and the average variance explained

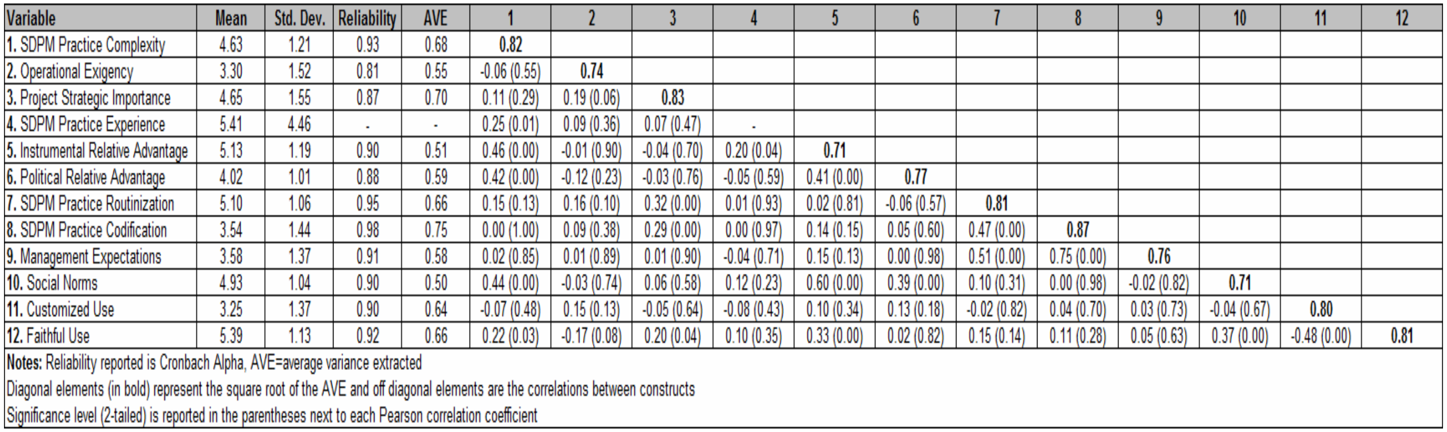
[10] AVE is defined as "the amount of variance that is captured by the construct in relation to the amount of variance due to measurement error" and is calculated using the following formula: $\sum\lambda_i^2 / [\sum\lambda_i^2 + \sum(1-\lambda_i^2)]$, where λ represents the factor loading of an item and i represents the item number within each construct (Fornell & Larcker, 1981).

(AVE) (Fornell and Larcker 1981; Nunnally and Bernstein 1994). To assess internal consistency, reliability of the newly defined summed scales was assessed by evaluating the Cronbach Alpha for each construct. As demonstrated in Table 6.4-5, reliability scores ranged between 0.81 to 0.93 and all exceeded the generally accepted minimum level of 0.80 (Nunnally and Bernstein 1994). AVE can also provide evidence of convergent validity provided the AVE score is at least 0.50 for each factor. AVE scores are reported in Table 6.4-5, and further confirm that factors demonstrate convergent validity.

Inter-construct correlations were mostly consistent with expectations, with a few exceptions. First was a significant positive correlation of practice complexity with both instrumental (r=0.46, *p*<0.01) and political (r=0.42, *p*<0.01) relative advantage. While technology adoption research typically finds that complexity and relative advantage perceptions are negatively related, a software development context requires that practices be used more completely as project complexity increases. Specifically, SDPM practices are likely to be most valuable within complex projects where implementation of the practice will subsequently be more complex. While initially surprising, the positive correlations between practice complexity and relative advantage perceptions seem to be consistent with the context of a software development environment.

Second, faithful and customized use demonstrated a significant negative correlation (r=-0.48, p<0.01). While a negative correlation was expected, such a high value calls into question the degree to which faithful and customized use are distinct concepts. The issue of how customized and faithful use might be related will be addressed in the following chapter.

Table 6.4-5: Correlation Matrix



Variable	Mean	Std. Dev.	Reliability	AVE	1	2	3	4	5	6	7	8	9	10	11	12
1. SDPM Practice Complexity	4.63	1.21	0.93	0.68	**0.82**											
2. Operational Exigency	3.30	1.52	0.81	0.55	-0.06 (0.55)	**0.74**										
3. Project Strategic Importance	4.65	1.55	0.87	0.70	0.11 (0.29)	0.19 (0.06)	**0.83**									
4. SDPM Practice Experience	5.41	4.46	-	-	0.25 (0.01)	0.09 (0.36)	0.07 (0.47)	-								
5. Instrumental Relative Advantage	5.13	1.19	0.90	0.51	0.46 (0.00)	-0.01 (0.90)	-0.04 (0.70)	0.20 (0.04)	**0.71**							
6. Political Relative Advantage	4.02	1.01	0.88	0.59	0.42 (0.00)	-0.12 (0.23)	-0.03 (0.76)	-0.05 (0.59)	0.41 (0.00)	**0.77**						
7. SDPM Practice Routinization	5.10	1.06	0.95	0.66	0.15 (0.13)	0.16 (0.10)	0.32 (0.00)	0.01 (0.93)	0.02 (0.81)	-0.06 (0.57)	**0.81**					
8. SDPM Practice Codification	3.54	1.44	0.98	0.75	0.00 (1.00)	0.09 (0.38)	0.29 (0.00)	0.00 (0.97)	0.14 (0.15)	0.05 (0.60)	0.47 (0.00)	**0.87**				
9. Management Expectations	3.58	1.37	0.91	0.58	0.02 (0.85)	0.01 (0.89)	0.01 (0.90)	-0.04 (0.71)	0.15 (0.13)	0.00 (0.98)	0.51 (0.00)	0.75 (0.00)	**0.76**			
10. Social Norms	4.93	1.04	0.90	0.50	0.44 (0.00)	-0.03 (0.74)	0.06 (0.58)	0.12 (0.23)	0.60 (0.00)	0.39 (0.00)	0.10 (0.31)	0.00 (0.98)	-0.02 (0.82)	**0.71**		
11. Customized Use	3.25	1.37	0.90	0.64	-0.07 (0.48)	0.15 (0.13)	-0.05 (0.64)	-0.08 (0.43)	0.10 (0.34)	0.13 (0.18)	-0.02 (0.82)	0.04 (0.70)	0.03 (0.73)	-0.04 (0.67)	**0.80**	
12. Faithful Use	5.39	1.13	0.92	0.66	0.22 (0.03)	-0.17 (0.08)	0.20 (0.04)	0.10 (0.35)	0.33 (0.00)	0.02 (0.82)	0.15 (0.14)	0.11 (0.28)	0.05 (0.63)	0.37 (0.00)	-0.48 (0.00)	**0.81**

Notes: Reliability reported is Cronbach Alpha, AVE=average variance extracted
Diagonal elements (in bold) represent the square root of the AVE and off diagonal elements are the correlations between constructs
Significance level (2-tailed) is reported in the parentheses next to each Pearson correlation coefficient

One other issue should be noted before proceeding. While a key dependent variable in this research is faithful use, it became clear that the wording of faithful usage questions focused on deviance from management expectations. To ensure that the measurement scheme was consistent with faithful use, each item was reverse scored so that the higher number represents more faithful usage. This was done primarily to simplify the interpretation of results and will not change the substance of results in any way.

6.4.2. Validation of Software Development Manager Constructs

A similar process of construct assessment and validation was performed for items answered by software development managers. However, examining the structure of constructs was constrained by the number of software developers who responded to the survey (11 as reported in Table 6.2-1). As a means of providing a large enough sample to evaluate the factor structure, a new survey was created and administered to project managers who had either expressed an interest in my research but had declined to participate in the overall study or were members of the Project Management Institute's Tulsa chapter. Forty-one project managers responded to a survey which included all items relating to practice codification, management expectations and practice routinization (fourteen total items). An exploratory factor analysis was conducted to ensure that items loaded as expected on specified constructs. Using the same settings as discussed above (principle axis factoring extraction method, direct oblimin – i.e., oblique – rotation, factors chosen by Eigenvalue>1, and items retained if loading and communality was greater than 0.50), the initial factor analysis converged in eight iterations and included all fourteen items loaded as expected. Table 6.4-6 demonstrates that the three factors explained 87.23% of the total variance.

Table 6.4-6: Factor Analysis (Total Variance Explained)

Factor	Eigenvalue	% of Variance	Cumulative %
1	9.45	67.51%	67.51%
2	1.70	12.14%	79.65%
3	1.06	7.58%	87.23%

All items were retained as is illustrated in the pattern matrix shown in Table 6.4-7.

Table 6.4-7: Factor Analysis (Factor Loadings)

	COD	ME	ROUT
COD1	**0.90**	0.01	-0.08
COD2	**0.97**	-0.05	-0.03
COD3	**0.74**	0.07	-0.18
COD4	**0.89**	0.05	-0.07
COD5	**0.81**	0.11	-0.13
ROUT1	0.17	-0.17	**-0.89**
ROUT2	0.03	0.06	**-0.90**
ROUT3	0.02	0.03	**-0.76**
ROUT4	0.06	0.14	**-0.80**
ROUT5	0.17	0.17	**-0.70**
ME1	0.01	**0.91**	-0.03
ME2	0.30	**0.59**	-0.16
ME3	-0.22	**0.69**	-0.31
ME4	0.33	**0.82**	0.16

COD = practice codification, ROUT = practice routinization, ME = management expectations

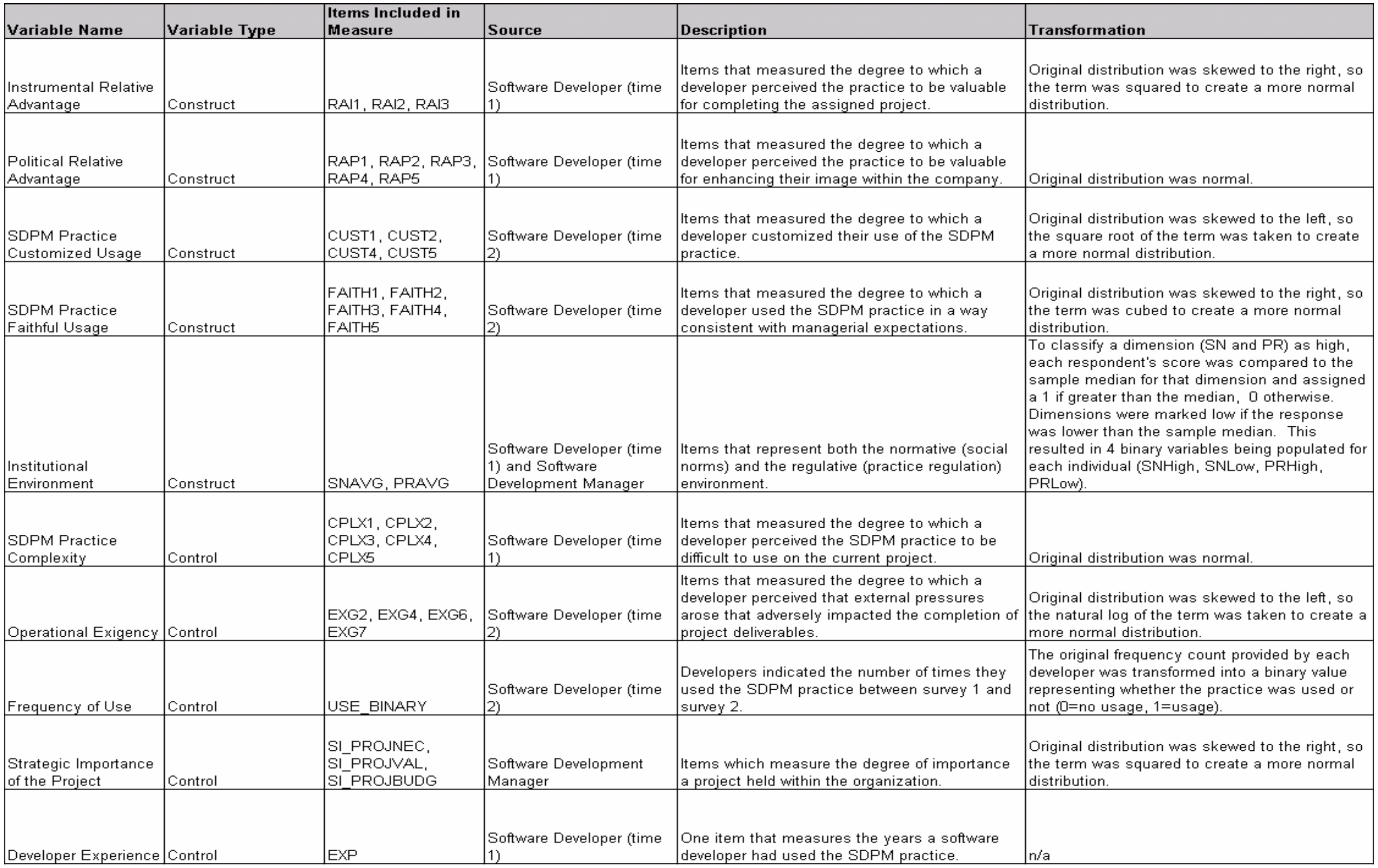
Once again, summed scales were created for each factor, where items were combined and then averaged to create the new composite score. Table 6.4-5 provides a view of the descriptives associated with the three software development manager constructs. Results demonstrate that the practice was considered highly routinized within the eleven software development groups (average of 5.38 on a scale where 7.0 represented the highest level of routinization), but that management expectations for use and the degree of documentation tended to vary more across groups. Divergent validity of constructs is demonstrated since no items loaded on more than one factor (Table 6.4-7) and correlation values were less than the square root of the AVE

(Table 6.4-5). Reliability (>0.80) and AVE (>0.50) scores reported in Table 6.4-5 also confirm that factors demonstrate convergent validity.

6.4.3. Overview of Research Constructs

Based on the preceding analysis, a set of research constructs and controls can be employed to test the research hypotheses. The hypotheses will be evaluated through a multiple regression procedure that, among other things, requires a normal distribution of the residuals (Hair et al. 1998). In order to address this requirement, each variable was evaluated for normality and, where appropriate, transformed to create a normal distribution. Table 6.4-8 describes the constructs and any transformations that were necessary for the analysis.

Table 6.4-8: Overview of Research Constructs and Controls



Variable Name	Variable Type	Items Included in Measure	Source	Description	Transformation
Instrumental Relative Advantage	Construct	RAI1, RAI2, RAI3	Software Developer (time 1)	Items that measured the degree to which a developer perceived the practice to be valuable for completing the assigned project.	Original distribution was skewed to the right, so the term was squared to create a more normal distribution.
Political Relative Advantage	Construct	RAP1, RAP2, RAP3, RAP4, RAP5	Software Developer (time 1)	Items that measured the degree to which a developer perceived the practice to be valuable for enhancing their image within the company.	Original distribution was normal.
SDPM Practice Customized Usage	Construct	CUST1, CUST2, CUST4, CUST5	Software Developer (time 2)	Items that measured the degree to which a developer customized their use of the SDPM practice.	Original distribution was skewed to the left, so the square root of the term was taken to create a more normal distribution.
SDPM Practice Faithful Usage	Construct	FAITH1, FAITH2, FAITH3, FAITH4, FAITH5	Software Developer (time 2)	Items that measured the degree to which a developer used the SDPM practice in a way consistent with managerial expectations.	Original distribution was skewed to the right, so the term was cubed to create a more normal distribution.
Institutional Environment	Construct	SNAVG, PRAVG	Software Developer (time 1) and Software Development Manager	Items that represent both the normative (social norms) and the regulative (practice regulation) environment.	To classify a dimension (SN and PR) as high, each respondent's score was compared to the sample median for that dimension and assigned a 1 if greater than the median, 0 otherwise. Dimensions were marked low if the response was lower than the sample median. This resulted in 4 binary variables being populated for each individual (SNHigh, SNLow, PRHigh, PRLow).
SDPM Practice Complexity	Control	CPLX1, CPLX2, CPLX3, CPLX4, CPLX5	Software Developer (time 1)	Items that measured the degree to which a developer perceived the SDPM practice to be difficult to use on the current project.	Original distribution was normal.
Operational Exigency	Control	EXG2, EXG4, EXG6, EXG7	Software Developer (time 2)	Items that measured the degree to which a developer perceived that external pressures arose that adversely impacted the completion of project deliverables.	Original distribution was skewed to the left, so the natural log of the term was taken to create a more normal distribution.
Frequency of Use	Control	USE_BINARY	Software Developer (time 2)	Developers indicated the number of times they used the SDPM practice between survey 1 and survey 2.	The original frequency count provided by each developer was transformed into a binary value representing whether the practice was used or not (0=no usage, 1=usage).
Strategic Importance of the Project	Control	SI_PROJNEC, SI_PROJVAL, SI_PROJBUDG	Software Development Manager	Items which measure the degree of importance a project held within the organization.	Original distribution was skewed to the right, so the term was squared to create a more normal distribution.
Developer Experience	Control	EXP	Software Developer (time 1)	One item that measures the years a software developer had used the SDPM practice.	n/a

The first independent variables in this study deal with instrumental and political relative advantage perceptions relating to the SDPM practice. While political relative advantage demonstrated a normal distribution (Figure 6.4-1), instrumental relative advantage produced a distribution that was skewed to the right (i.e., people tended to perceive the SDPM practices to be instrumentally valuable). In order to readjust the distribution, a power transformation was used where the term was squared. Figure 6.4-2 illustrates the original distribution (left) and the transformed distribution (right).

Figure 6.4-1: Histogram (Political Relative Advantage)

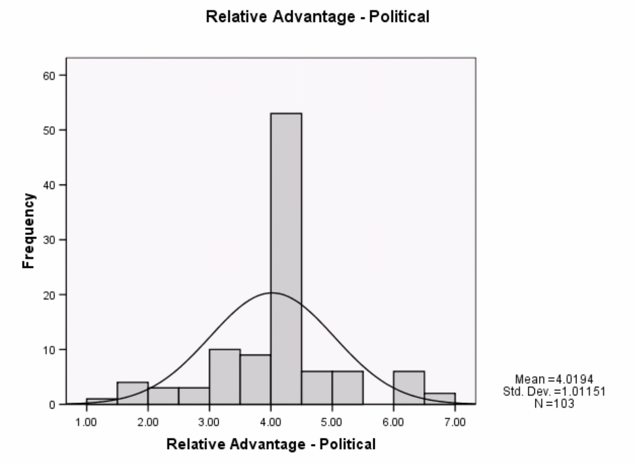
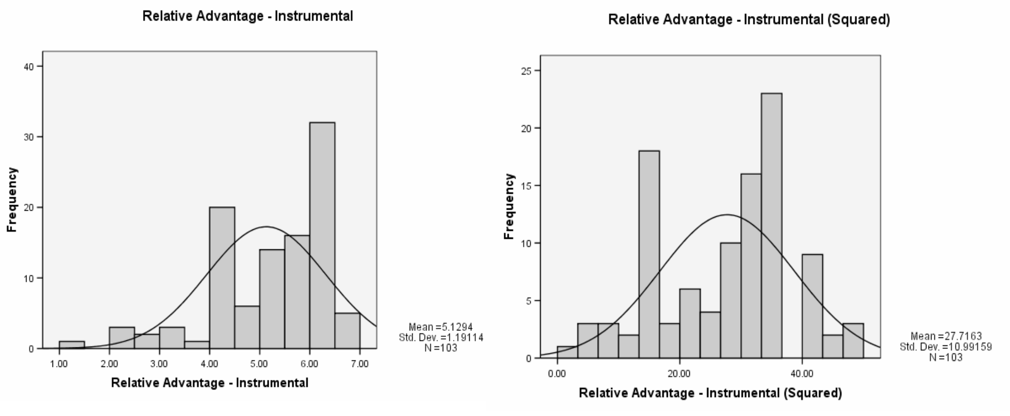
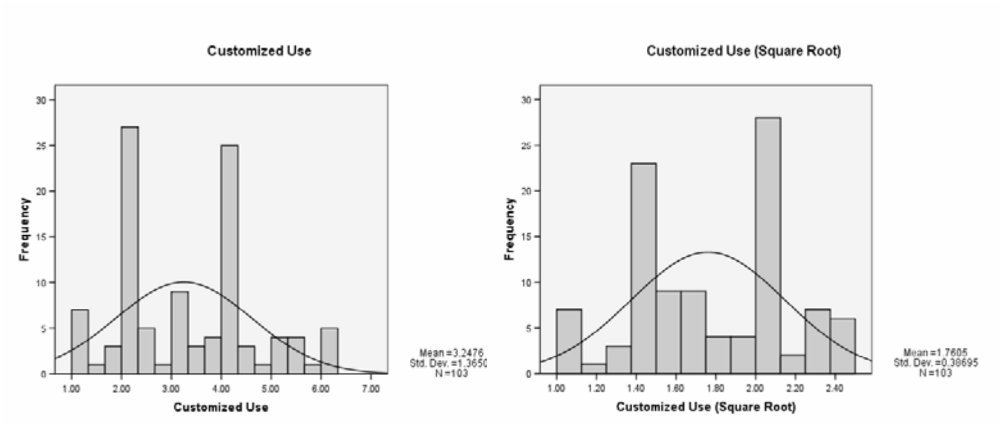


Figure 6.4-2: Histogram (Instrumental Relative Advantage)



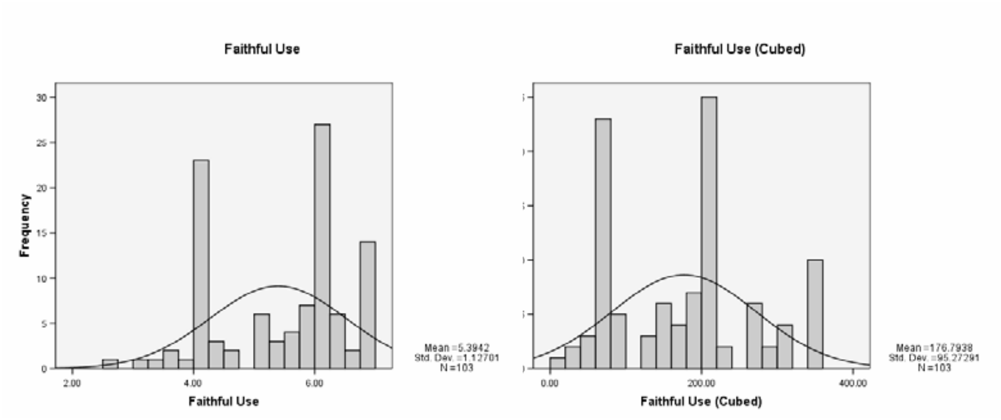
The next set of variables considered were the dependent variables in this study, customized and faithful use. Both distributions were skewed and non-normal but differed in the nature of skewness. Customized usage was found to have a fair range of developer responses, but there were a large number that rested around 2 (very low customization) and 4 (middle of the scale). In order to correct this distribution, the term's square root was taken. The resulting distribution is more characteristic of a normal distribution as displayed in Figure 6.4-3.

Figure 6.4-3: Histogram (Customized Use)



For faithful use, a large number of responses were found at two points, the middle (4) and close to the top (6) of the scale. This resulted in a strongly right-skewed distribution that was addressed using a power transformation (cubed). Figure 6.4-4 illustrates the original distribution (left) and the transformed distribution (right).

Figure 6.4-4: Histogram (Faithful Use)



Further, two control variables were also found to be skewed, one to the left (operational exigency) and one to the right (project strategic importance). For operational exigency, the natural log of the term was taken to create a normal transformation (pictured in Figure 6.4-5) while the right-skewed distribution for project strategic importance was adjusted using a power transformation (squared) as illustrated in Figure 6.4-6.

Figure 6.4-5: Histogram (Operational Exigency)

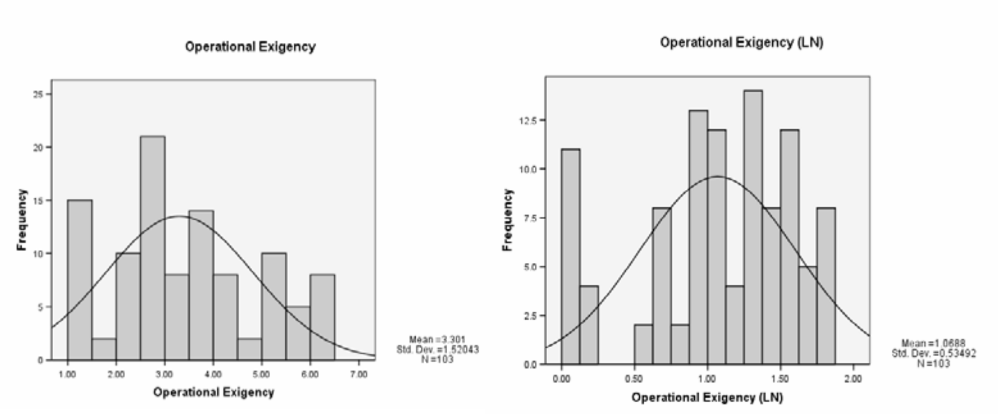
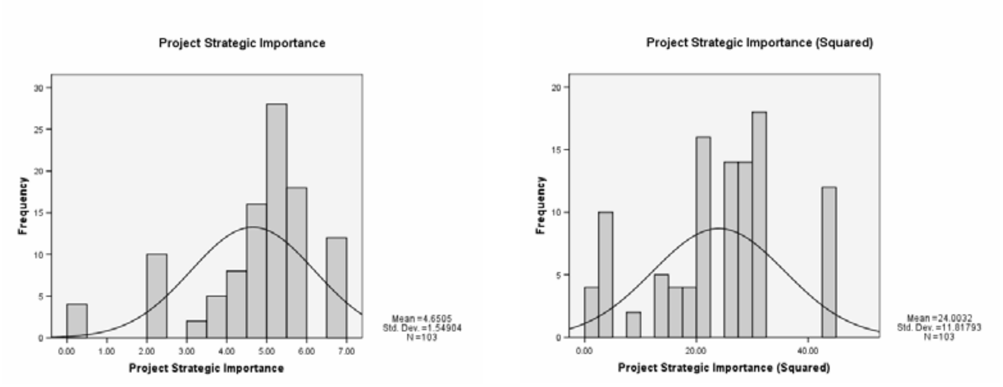
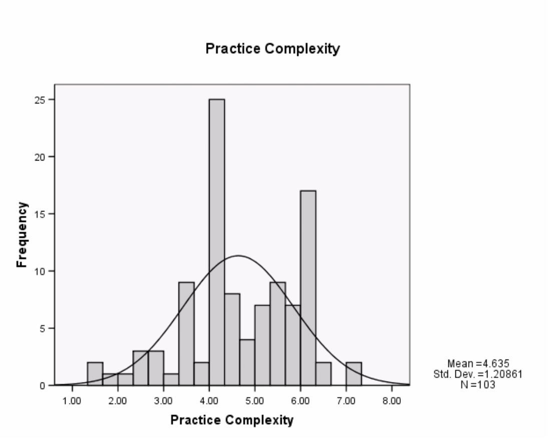


Figure 6.4-6: Histogram (Project Strategic Importance)



Concerning the remaining control variables, project complexity was found to follow a normal distribution (Figure 6.4-7) and, as such, was not modified; developer experience with the practice was left as-is (one self-report item measuring the years a developer had been using the SDPM practice); frequency of use was slightly modified to account for whether the SDPM practice was used at all between the two developer surveys (0=no, 1=yes). Actual usage of the SDPM practice during the one month period between software developer surveys was evenly split in the sample, with 52 responses indicating no use while 51 showed some degree of usage during that period.

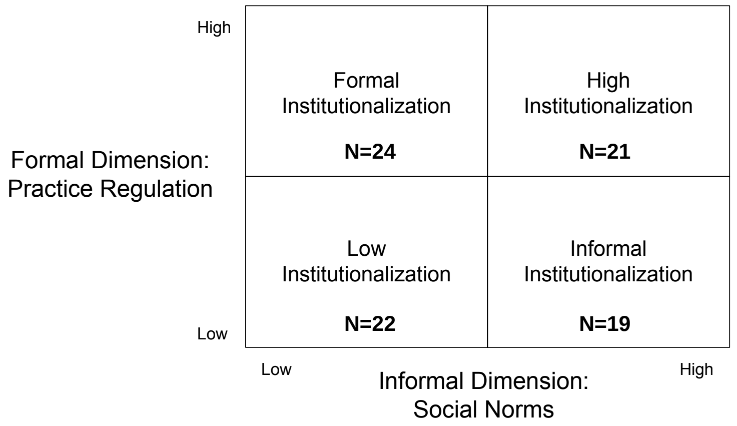
Figure 6.4-7: Histogram (SDPM Practice Complexity)



The final measure to be discussed is the Institutional Environment, composed of SDPM practice social norms and practice regulation. Hypotheses are phrased to address the impacts of low / high institutional environment on a developer's nature of usage, so a means of determining high vs. low conditions of both institutional environment dimensions had to be determined. Several options were considered, including generating an "ideal" score for each dimension, but were abandoned because of the difficulty in objectively determining what constitutes high and low conditions. Rather, a decision was made to compare individual scores for both dimensions against the sample median value. Four new dummy variables were created (HiSNHiPR, HiSNLoPR, LoSNHiPR, LoSNLoPR), which captured the individual's position in relation to others in the sample. For each variable, three scenarios could exist. Either (a) the individual could have a score higher than the sample median value, (b) a score lower than the sample median value or (c) a score equal to the sample median value. For cases where the “high” was being determined, only individual scores higher than the sample median value would be considered “high”. The same process was followed for low conditions, except a designation of “low” was assigned if the individual’s score was lower than the sample median value.

As an example, consider an actual user from the sample (id=‘1C6AED1E5BD9D371181B306390FAD0F9261EDF3D’) with a social norm score of 5.2 and a practice regulation score of 2.53. Since the sample median values are 5.0 (social norm) and 3.25 (practice regulation), the user would be assigned the following values for the four dummy variables: HiSNHiPR=0, HiSNLoPR=1, LoSNHiPR=0, LoSNLoPR=0. Cases where an individual’s SN and/or PR score was equal to the sample median always resulted in a 0 for all four dummy variables. In cases where median values are not evidenced by the individual, the preceding scheme results in individuals having one of the four variables set to 1 and the remaining three set to 0. Figure 6.4-8 shows the count of developers who existed in each condition.

Figure 6.4-8: Developer Count per Institutional Environment Condition



Formal Dimension: Practice Regulation	Low	High
High	Formal Institutionalization **N=24**	High Institutionalization **N=21**
Low	Low Institutionalization **N=22**	Informal Institutionalization **N=19**

Notes: 17 developers did not fall in any category because the developer's social norm and/or practice regulation score was equal to the sample median value.

6.5. Research Hypotheses

With variables specified and validated, the following section moves to evaluate the research hypotheses outlined in Table 5.5-1. Because the dependent variables are continuous and independent variables are both continuous and dichotomous, relationships proposed in this research can be evaluated using a multivariate regression technique. Multivariate regression provides a means of assessing both the entire model's ability to predict an outcome variable as well as picturing the contribution each independent variable makes towards that prediction (Hair et al. 1998). This study utilized hierarchical multiple regression (Cohen and Cohen 1983) to analyze two separate regression models, each with a specific SDPM practice usage type (customized or faithful use) as the dependent variable and where independent variables were entered into the model in blocks. The first block of variables included the six control variables discussed earlier in the chapter. Next, relative advantage measures and institutional environment variables were entered into the second block. Finally, hypothesized interaction terms (relative advantage * institutional environment) were entered in the third block. This approach allows one to evaluate the contribution each block of variables (controls, direct effects and interaction terms) make in predicting the dependent variable by examining changes in explained variance and $\boldsymbol{\beta}$ (standardized regression coefficient) at each step.

Interaction terms included in model three were addressed using a commonly accepted approach for situations where you have a dichotomous independent variable (i.e., formal institutionalization) and a continuous moderator (i.e., instrumental relative advantage) (Baron and Kenny 1986). In order to reduce multicollinearity resulting from the use of interaction terms, continuous variables were centered by subtracting the sample mean value from each response (Aiken and West 1991). The product of the newly centered continuous variable and the dichotomous variable was then generated to create an interaction term. Per Baron and Kenny, interaction exists if the interaction term is found to be significant (regardless of the significance of the main effects).

Revised hypotheses containing references to the institutional environment posed a specific challenge in that they were posited in relation to each other (i.e., customization more likely in high institutionalization versus informal institutionalization environments). To assess these hypotheses, the particular institutional environment was first examined in the regression model to see if the ***β*** value was significant. If so, then a mean comparison procedure was conducted to determine if a significant difference existed between the various institutional environments. Specifically, a univariate generalized linear model was utilized where the usage type means were compared for the various categories of institutional environment. If an overall significant difference was found between levels, a Simple planned comparison test was utilized to determine where differences existed between categories of the institutional environment. A significant difference between levels provided a means of evaluating the various institutional environment hypotheses.

The following sections will discuss hypotheses testing first for customized use, and then for faithful use.

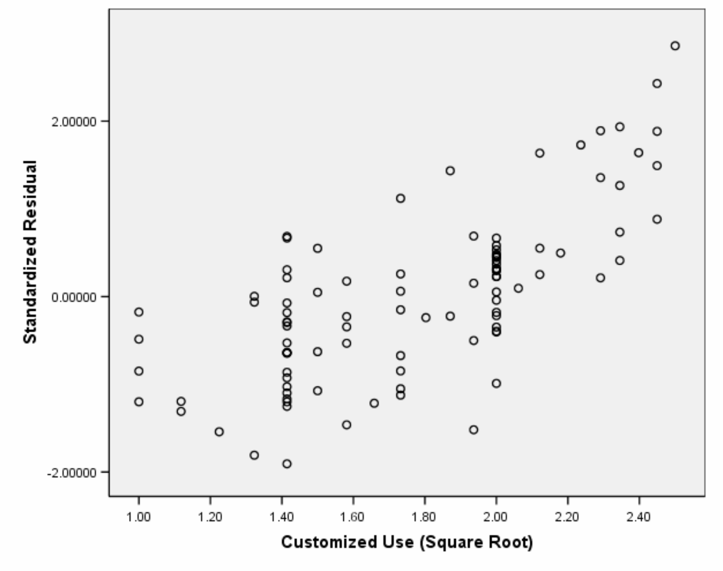
6.5.1. Testing Proposed Antecedents to Customized Use

6.5.1.1. Examining Assumptions in Multivariate Regression Analysis

Multiple regression procedures make four primary assumptions which must be considered before results can be interpreted: (a) linearity between the independent and dependent variables; (b) constant variance of the residuals; (c) independence of error terms and (d) normality of the residual distribution (Hair et al. 1998). These assumptions can only be evaluated after the model has been specified and examined and are addressed to ensure that any errors in prediction are a function of the relationships tested rather than being caused by factors not directly specified within the regression model.

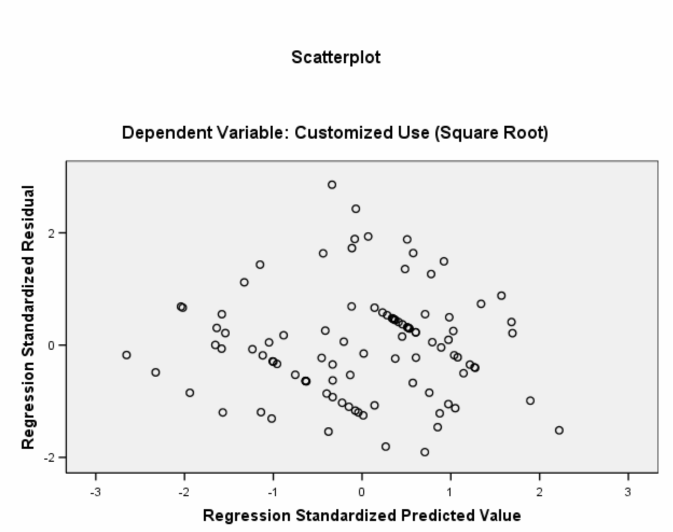
Standardized residuals scores are plotted against the dependent variable in order to assess linearity between independent variables and the dependent variable. As can be seen in Figure 6.5-1, the relationship does exhibit in an upward sloping set of points, suggesting that the assumption of linearity is not violated in this model.

Figure 6.5-1: Assessing Linearity (Customized Use)



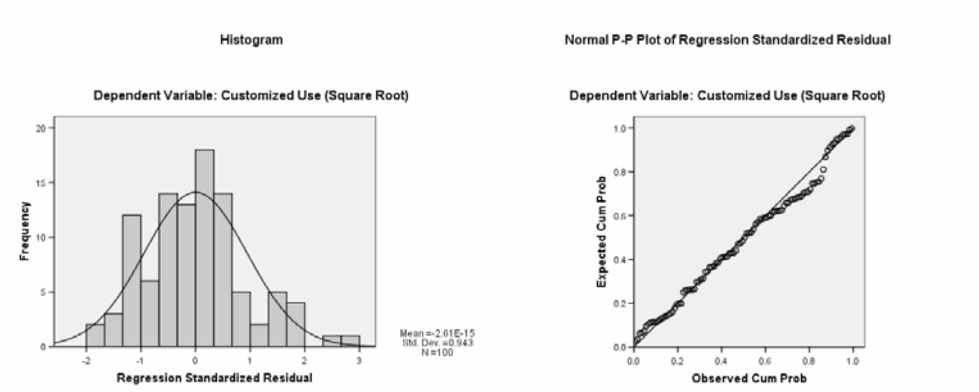
The next two assumptions (constant variance and independence of residuals) can both be assessed by a plot of the standardized residual versus the standardized predicted value. As illustrated in Figure 6.5-2, the somewhat even distribution of residuals around 0 (above and below) suggests that the assumption of homoscedasticity is held. The plots do not present any specific pattern suggesting that the assumption of independent error terms is also held. As a final approach for evaluating independence, a correlation analysis was run between the residuals for each SDPM practice. The resulting Pearson correlation value was found to be small and not significant (-0.121, p=0.424), providing further evidence of error term independence.

Figure 6.5-2: Modeling Variance of the Error Term (Customized Use)



Finally, the assumption that residuals are normally distributed is also held. One can detect departures from normality in several ways, including a visual inspection of the residuals distribution, a visual inspection of the normal probability plot, and via the Kolmogorov-Smirnov test for detecting non-normality. As pictured in Figure 6.5-3, all three demonstrate that the error term distribution does not violate assumptions of normality.

Figure 6.5-3: Normality of Error Term Distribution (Customized Use)



One-Sample Kolmogorov-Smirnov Test

		Standardized Residual
N		100
Normal Parameters[a,b]	Mean	.0000000
	Std. Deviation	.94280904
Most Extreme Differences	Absolute	.082
	Positive	.082
	Negative	-.039
Kolmogorov-Smirnov Z		.824
Asymp. Sig. (2-tailed)		.506

[a] Test distribution is Normal.

[b] Calculated from data.

One additional consideration within regression analysis is ensuring that collinearity does not exist. Collinearity represents the relationship between two or more (multicollinearity) variables, and can be problematic in regression analysis since its presence can create artificial relationships between unrelated variables and significantly skew results (Hair et al. 1998). One method of assessing whether collinearity exists is to examine the variance inflation factor (VIF) which indicates the “degree to which each independent variable is explained by the other independent variables” (Hair et al. 1998, p. 193). Collinearity is apparent when the VIF is greater than 2 (Kutner, Nachtsheim, Neter and Li 2004). Table 6.5-3 demonstrates that VIF values are all less than the 2 threshold.

6.5.1.2. Analysis of Proposed Relationships for Customized Use

Table 6.5-3 contains the results of the regression procedure used to test hypotheses relating to customized use. Because of the small sample size, a reduced model was used to conserve degrees of freedom. Before evaluating the reduced model, all direct effects (hypothesized and non-hypothesized) were included in a full model and examined to verify that using the reduced model was appropriate. Specifically, political relative advantage, low institutionalization and company variables were included in the model. As pictured in Appendix X, results demonstrate that adding the three variables minimally impacts the overall R^2 for the third model (R^2=0.503 for the full model as opposed to R^2=0.501 for the reduced model). Further, none of the three variables were found to exert a significant impact on customized use. As such, the reduced model (excluding political relative advantage, low institutionalization and company) was utilized to examine proposed hypotheses.

For the first regression model where only controls were entered, the overall regression model was found to be significant at the 0.01 level (F-value=9.365, $p \leq 0.01$) with a R^2 of 0.377. For model 2 (F-value=8.702, $p \leq 0.01$), the inclusion of institutional environment and relative advantage constructs significantly increased the model R^2 to 0.494. Introducing the interaction term into the third model, while still providing a significant overall model (F-value=8.018, $p \leq 0.01$), did not result in a significantly different model than model 2 ($R^2=0.501$). As such, model 2 was used to examine hypotheses related to customized use.

As the regression results indicate, several control variables were shown to play an important role in explaining customized use. Consistent with expectations, practice complexity ($p \leq 0.10$) and faithful SDPM practice use ($p \leq 0.01$) demonstrated a negative relationship with customization. Further, frequency of use which was included as a proxy for whether a developer needed to use the SDPM practice indicated a positive relationship with customization ($p \leq 0.01$). The remaining control variables (operational exigency, developer experience and the project's strategic importance) did not exhibit a significant relationship with customized use.

For RH1, instrumental relative advantage perceptions were suggested to have a positive impact on customized SDPM practice use. A significant positive relationship was demonstrated ($\beta=0.009$, $p \leq 0.05$), providing support for the assertion that developer perceptions of SDPM practice value for completing project tasks would result in a greater likelihood of customizing the practice to meet specific project needs.

Next, RH2 posited that customized practice use would be most likely in environments with High Institutionalization, next highest in Informal Institutionalization, and lowest in Formal Institutionalization. To fully examine this hypothesis, the coefficients for each type of institutionalization were required to be significant. High Institutionalization was found to be marginally significant (β=0.162, *p*≤0.10) while Informal Institutionalization (β=-0.106, *p*=0.237) and Formal Institutionalization (β=-0.069, *p*=0.370) did not exhibit a significant relationship. While a lack of significance for all three categories prevents a full testing of RH2, it does permit further examination of the differences between High and Informal / Formal Institutionalized environments.

Descriptive statistics for the customized use within these three institutional environments are provided in Table 6.5-1.

Table 6.5-1: Descriptive Statistics for Customized Use

	N	Mean	Std. Dev.
High Institutionalization	21	1.893	0.405
Informal Institutionalization	19	1.607	0.406
Formal Institutionalization	24	1.723	0.406

A univariate generalized linear model was used to determine if a significant difference existed in the level of customized use between the three institutional environments. The overall model was found to be marginally significant (F=2.533, $p \leq 0.10$), suggesting that a difference existed between High Institutionalization environments and at least one of the other institutional environment categories. Results from the Simple contrast provided in Table 6.5-2 demonstrate that customization was significantly more prevalent in High Institutionalization environments than that found in Informal Institutionalization ($p \leq 0.05$), but not with customization levels in Formal Institutionalization (p=0.166).

Table 6.5-2: Customized Use Contrasts between High, Informal and Formal Institutionalization Environments

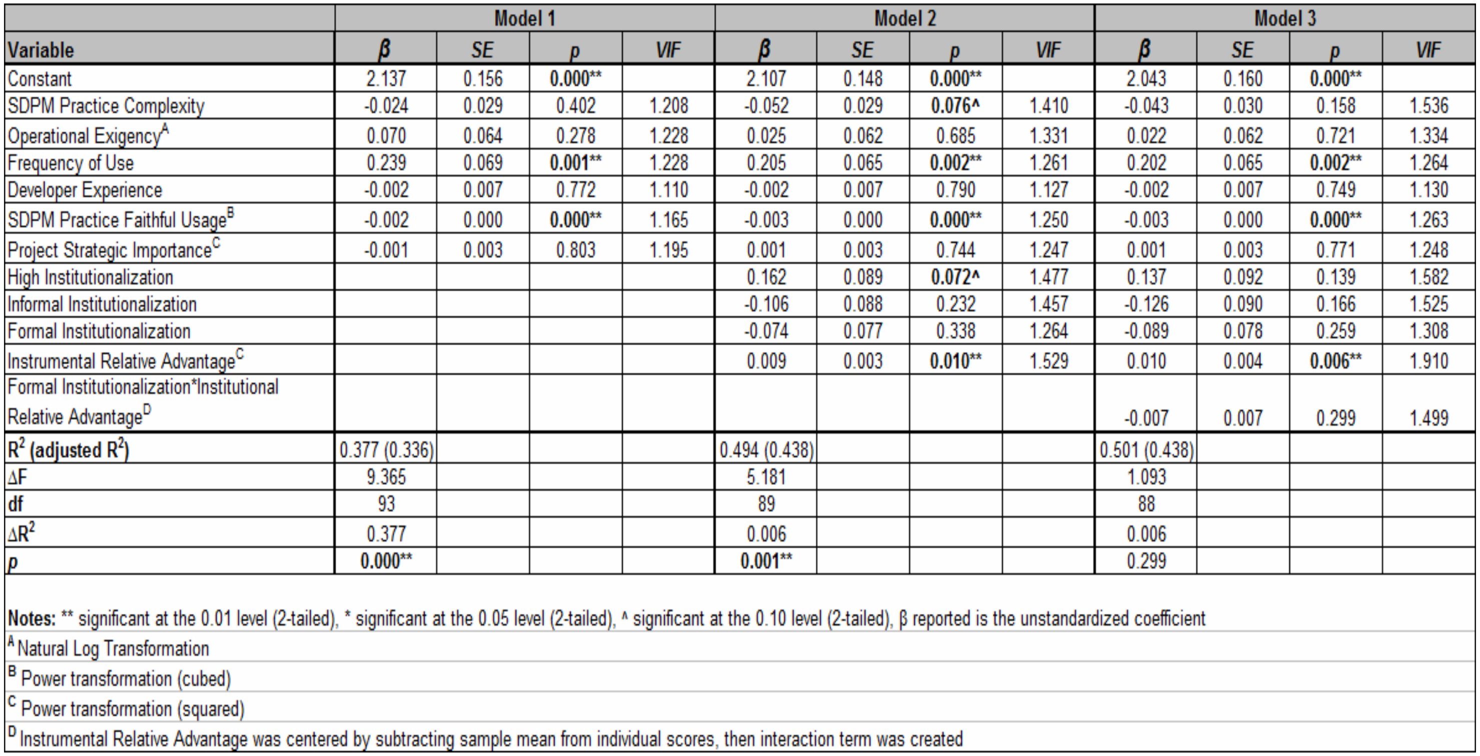
Comparison	Contrast Estimate	Std. Error	*p*
High vs. Informal	-0.286	0.128	0.030
High vs. Formal	-0.170	0.121	0.166

Thus, findings do provide marginal support for RH2 since customization was greatest in High Institutionalization environments and was found to be significantly higher than Informal (but not Formal) environments. Examination of Table 6.5-1 illustrates that the mean level of customization was greatest for developers in High Institutionalization environments (as predicted), but lowest for those in Informal Institutional environments (not expected). There are several possible explanations for why SDPM practice customization levels were not exactly as posited. First, the presence of a significant difference between High and Informal environments combined with a lack of significant difference between High and Formal environments might suggest that practice regulation plays a more important role than social norms in promoting practice customization. This finding is somewhat surprising since social norms long been suggested to strongly influence individual behavior (Ajzen 1985). However, it is likely that a developer who knows they will be evaluated by management regarding SDPM practice use will find ways to use it as effectively as possible. This is especially important in an organizational environment since effective use can provide a software developer the ability to positively differentiate themselves from peers within the development group. So while peer pressure can encourage usage, it seems possible that practice regulation provides the most important means of encouraging customized use.

Another possible explanation for the unexpected findings might lie in the small number of developers (High=21, Informal=19, Formal=24) categorized within each institutional environment. A low number of responses within each group could prevent meaningful differences from surfacing. This suggestion could be addressed by increasing the number of respondents in each institutional environment category.

RH3 was not supported since the instrumental relative advantage / Formal Institutionalization environment interaction term did not provide a significant change in the model R^2 (ΔR^2=0.007).

Table 6.5-3: Regression Results for Customized Use



	Model 1				Model 2				Model 3			
Variable	***β***	***SE***	***p***	***VIF***	***β***	***SE***	***p***	***VIF***	***β***	***SE***	***p***	***VIF***
Constant	2.137	0.156	**0.000****		2.107	0.148	**0.000****		2.043	0.160	**0.000****	
SDPM Practice Complexity	-0.024	0.029	0.402	1.208	-0.052	0.029	**0.076^**	1.410	-0.043	0.030	0.158	1.536
Operational Exigency[A]	0.070	0.064	0.278	1.228	0.025	0.062	0.685	1.331	0.022	0.062	0.721	1.334
Frequency of Use	0.239	0.069	**0.001****	1.228	0.205	0.065	**0.002****	1.261	0.202	0.065	**0.002****	1.264
Developer Experience	-0.002	0.007	0.772	1.110	-0.002	0.007	0.790	1.127	-0.002	0.007	0.749	1.130
SDPM Practice Faithful Usage[B]	-0.002	0.000	**0.000****	1.165	-0.003	0.000	**0.000****	1.250	-0.003	0.000	**0.000****	1.263
Project Strategic Importance[C]	-0.001	0.003	0.803	1.195	0.001	0.003	0.744	1.247	0.001	0.003	0.771	1.248
High Institutionalization					0.162	0.089	**0.072^**	1.477	0.137	0.092	0.139	1.582
Informal Institutionalization					-0.106	0.088	0.232	1.457	-0.126	0.090	0.166	1.525
Formal Institutionalization					-0.074	0.077	0.338	1.264	-0.089	0.078	0.259	1.308
Instrumental Relative Advantage[C]					0.009	0.003	**0.010****	1.529	0.010	0.004	**0.006****	1.910
Formal Institutionalization*Institutional Relative Advantage[D]									-0.007	0.007	0.299	1.499
R^2 (adjusted R^2)	0.377 (0.336)				0.494 (0.438)				0.501 (0.438)			
ΔF	9.365				5.181				1.093			
df	93				89				88			
ΔR^2	0.377				0.006				0.006			
p	**0.000****				**0.001****				0.299			

Notes: ** significant at the 0.01 level (2-tailed), * significant at the 0.05 level (2-tailed), ^ significant at the 0.10 level (2-tailed), β reported is the unstandardized coefficient

[A] Natural Log Transformation

[B] Power transformation (cubed)

[C] Power transformation (squared)

[D] Instrumental Relative Advantage was centered by subtracting sample mean from individual scores, then interaction term was created

6.5.2. Testing Proposed Antecedents to Faithful Use

As with customized use, a reduced model for faithful use was used to conserve degrees of freedom. Specifically, low institutionalization and company variables were included in the full model. As shown in Appendix XI, results demonstrate that adding these variables does not significantly increase the overall R^2 (R^2=0.557 for the full model as opposed to R^2=0.555 for the reduced model). In addition, neither is found to exhibit a significant impact on faithful use. As such, the reduced model (excluding low institutionalization and company) was utilized to examine the remaining hypotheses.

6.5.2.1. Examining Assumptions in Multivariate Regression Analysis

Following the methods outlined in section 6.5.1.1, assumptions were addressed for regression models with faithful use as the dependent variable. As illustrated below, assumptions of linearity (Figure 6.5-4), constant variance and independence of residuals[11] (Figure 6.5-5), and normality of residuals (Figure 6.5-6) all held when faithful use was positioned as the dependent variable. Table 6.5-4 shows that VIF scores were well under 2, indicating that multicollinearity was not a problem.

[11] The results from a correlation analysis of the residuals for both SDPM practices showed to be large and significant (0.315, p=0.033). This suggests that caution must be taken when interpreting the results of the regression analysis.

Figure 6.5-4: Assessing Linearity (Faithful Use)

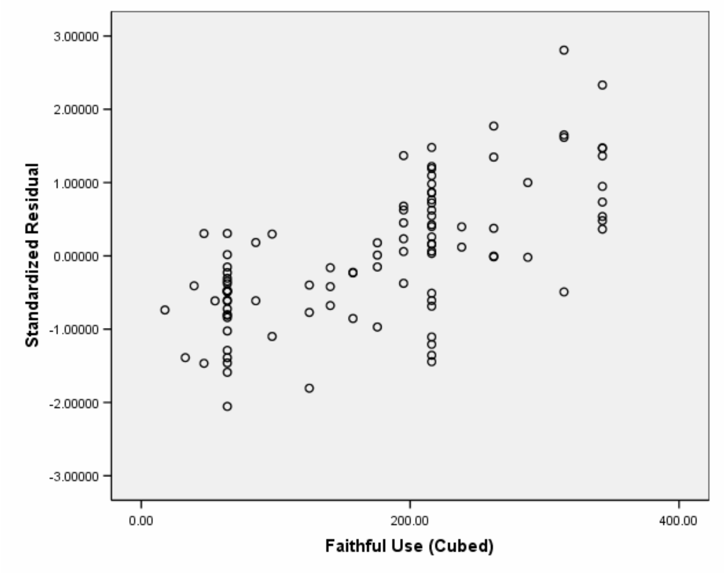


Figure 6.5-5: Modeling Variance of the Error Term (Faithful Use)

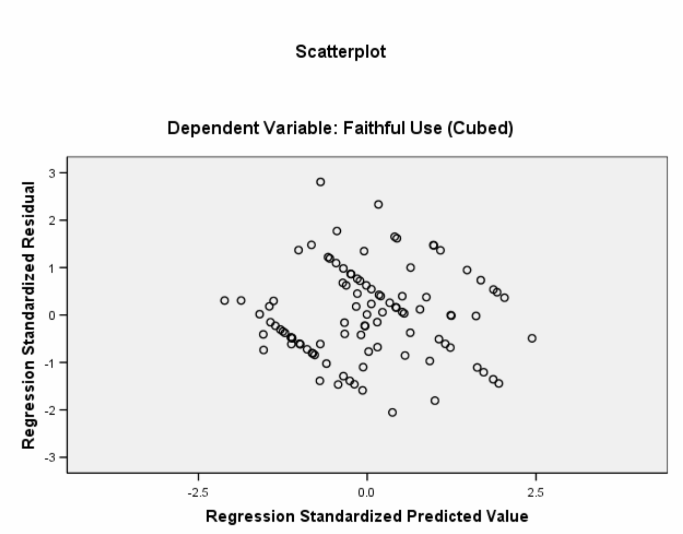
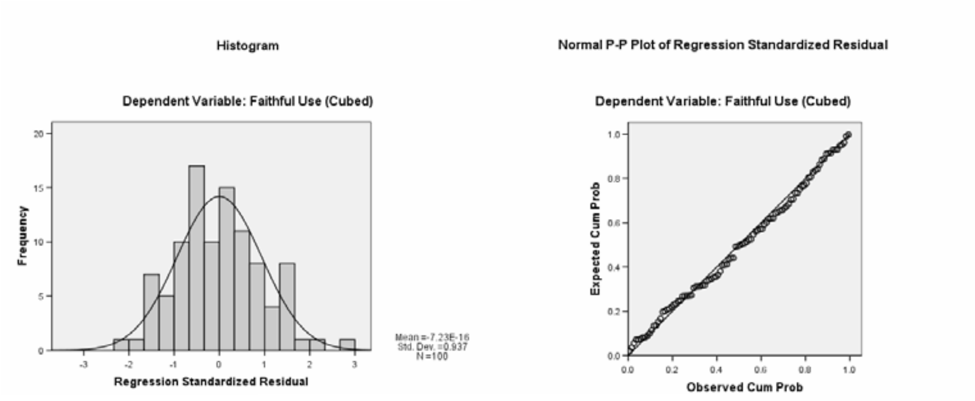


Figure 6.5-6: Normality of Error Term Distribution (Faithful Use)



One-Sample Kolmogorov-Smirnov Test

		Standardized Residual
N		100
Normal Parameters[a,b]	Mean	.0000000
	Std. Deviation	.93743687
Most Extreme Differences	Absolute	.055
	Positive	.055
	Negative	-.035
Kolmogorov-Smirnov Z		.551
Asymp. Sig. (2-tailed)		.922

[a] Test distribution is Normal.

[b] Calculated from data.

6.5.2.2. Analysis of Proposed Relationships for Faithful Use

Table 6.5-4 contains the results of the regression procedure used to test hypotheses relating to faithful use. For the first regression model where only controls were entered included, the overall regression model was found to be significant at the 0.01 level (F-value=11.100, $p \leq 0.01$) with a R^2 of 0.417. Model 2 was also significant (F-value=10.700, $p \leq 0.01$), producing a R^2 of 0.546. The third model included the instrumental relative advantage and formal institutionalization interaction term and was significant (F-value=9.994, $p\leq0.01$), but displayed a non-significant difference with the second model ($p=0.174$). This requires the use of model 2 in evaluating RH4 through RH6.

Several controls showed a significant relationship with faithful usage. Both operational exigency ($p\leq0.10$) and customized use ($p\leq0.01$) demonstrated a negative relationship with faithfulness, findings which were consistent with expectations. In addition, frequency of use showed a positive relationship with faithfulness ($p\leq0.01$). The remaining control variables (practice complexity, developer experience and the project's strategic importance) did not play an important role in explaining faithful use of the SDPM practice.

It is also important to note that instrumental relative advantage was shown to have a positive relationship with faithful use ($\beta=2.890$, $p\leq0.01$), a relationship that was not hypothesized. While not a posited relationship, a cognitive dissonance perspective might actually explain this result. As discussed earlier, individuals in situations of cognitive conflict will adjust their perceptions to create mental consonance (Festinger 1962). As a means of creating synchrony in thought, developers that perceive a practice instrumentally valuable might adjust thinking such that behavior is believed to be line with organizational expectations. Whether this perception matches reality is inconsequential in this study since the survey instrument relies on faithfulness data from the perspective of the software developer.

RH4 exhibited a significant relationship between political relative advantage and practice faithfulness, but it was in the opposite direction of what was predicted (β=-17.140, $p \leq 0.05$), failing to provide support. Findings demonstrated that developers were less faithful in their usage of the practice when it was viewed to enhance their image within the organization. This is a rather strange finding on its face, but might be explained by considering the reference point for both questions. Faithful usage questions were addressed to "management in our software developer group" while political relative advantage questions focused on image "within my organization". It is possible that developers answering this question might have used different referent groups such as peers for "within my organization" questions and actual management for faithful usage questions. Thus, the relationship to faithful SDPM practice use is not necessarily assured - especially if a software developer views social norms as more important than organizational expectations for use.

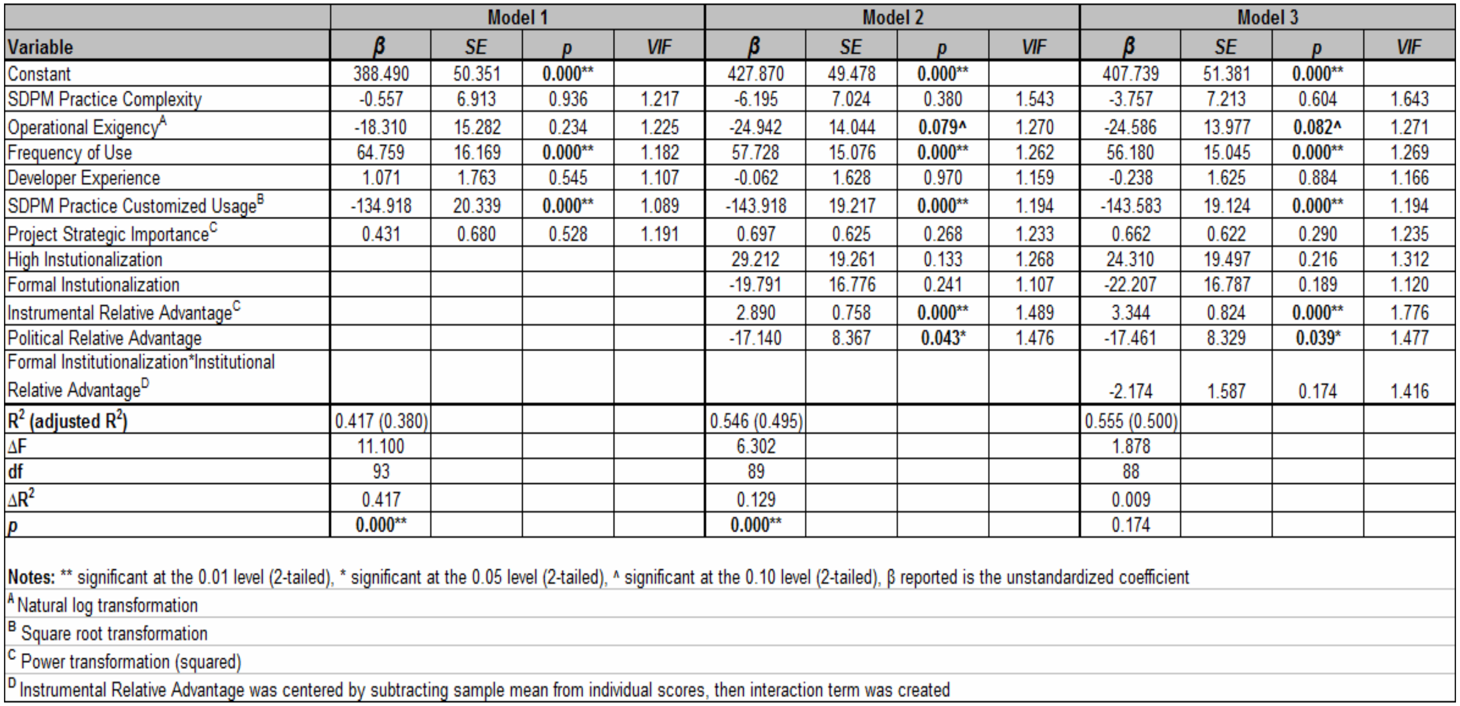
For RH5, both High (β=-29.212, p=0.133) and Formal (β=-19.791, p=0.241) Institutionalization environments were not found to contribute significantly to a developer's faithful use of the SDPM practice. As such, further comparison of the mean levels of faithful use between categories was not meaningful. The assertion in RH5 was not supported.

RH6 addressed the impact of Formal Institutionalization on faithful use, accounting for instrumental relative advantage perceptions. The interaction with instrumental relative advantage did not reveal a significant difference when introduced into the regression model ($\Delta R^2=0.009$, $p=0.174$), thus failing to provide support for RH6.

The lack of impact of strong practice regulation on faithful use is quite surprising since formal control has frequently been demonstrated as a way to promote compliant employee behavior within an organization (Jaworski, Stathakopoulos and Krishnan 1993; Kirsch 2004). A possible explanation for the lack of relationship for both environment types might be that developers, working within the context of a project are often pushed to alter behavior based on the short-term needs (Fitzgerald 1997). As such, the institutional environment will not be as salient in predicting faithful use as project-driven circumstances. Another possible explanation is that the small sample size (as reported in Figure 6.4-8) does not provide enough power to appropriately evaluate the institutional environment relationships.

Table 6.5-5 provides a summary of hypotheses results.

Table 6.5-4: Regression Results for Faithful Use



	Model 1				Model 2				Model 3			
Variable	**β**	***SE***	***p***	***VIF***	**β**	***SE***	***p***	***VIF***	**β**	***SE***	***p***	***VIF***
Constant	388.490	50.351	**0.000****		427.870	49.478	**0.000****		407.739	51.381	**0.000****	
SDPM Practice Complexity	-0.557	6.913	0.936	1.217	-6.195	7.024	0.380	1.543	-3.757	7.213	0.604	1.643
Operational Exigency[A]	-18.310	15.282	0.234	1.225	-24.942	14.044	**0.079^**	1.270	-24.586	13.977	**0.082^**	1.271
Frequency of Use	64.759	16.169	**0.000****	1.182	57.728	15.076	**0.000****	1.262	56.180	15.045	**0.000****	1.269
Developer Experience	1.071	1.763	0.545	1.107	-0.062	1.628	0.970	1.159	-0.238	1.625	0.884	1.166
SDPM Practice Customized Usage[B]	-134.918	20.339	**0.000****	1.089	-143.918	19.217	**0.000****	1.194	-143.583	19.124	**0.000****	1.194
Project Strategic Importance[C]	0.431	0.680	0.528	1.191	0.697	0.625	0.268	1.233	0.662	0.622	0.290	1.235
High Instutionalization					29.212	19.261	0.133	1.268	24.310	19.497	0.216	1.312
Formal Instutionalization					-19.791	16.776	0.241	1.107	-22.207	16.787	0.189	1.120
Instrumental Relative Advantage[C]					2.890	0.758	**0.000****	1.489	3.344	0.824	**0.000****	1.776
Political Relative Advantage					-17.140	8.367	**0.043***	1.476	-17.461	8.329	**0.039***	1.477
Formal Institutionalization*Institutional Relative Advantage[D]									-2.174	1.587	0.174	1.416
R^2 (adjusted R^2)	0.417 (0.380)				0.546 (0.495)				0.555 (0.500)			
ΔF	11.100				6.302				1.878			
df	93				89				88			
ΔR^2	0.417				0.129				0.009			
p	**0.000****				**0.000****				0.174			

Notes: ** significant at the 0.01 level (2-tailed), * significant at the 0.05 level (2-tailed), ^ significant at the 0.10 level (2-tailed), β reported is the unstandardized coefficient

[A] Natural log transformation

[B] Square root transformation

[C] Power transformation (squared)

[D] Instrumental Relative Advantage was centered by subtracting sample mean from individual scores, then interaction term was created

Table 6.5-5: Summary of Hypotheses Tests

Revised Hypothesis		Supported?
RH1	A software developer’s instrumental relative advantage perceptions of the SDPM practice will increase their customized use of the practice.	Supported
RH2	SDPM practice customization will be highest in environments of High Institutionalization, less in Informal Institutionalization, and least in Formal Institutionalization.	Partially Supported
RH3	SDPM practice customization within Formal Institutionalization environments will be enhanced when a software developer perceives the SDPM practice to be instrumentally valuable.	Not Supported
RH4	A software developer’s political relative advantage perceptions of the SDPM practice will increase their faithful use of the practice.	Not Supported
RH5	SDPM practice faithful use will be more likely in High Institutionalization environments than in Formal Institutionalization environments.	Not Supported
RH6	The positive relationship between Formal Institutionalization environments and SDPM practice faithful use will be attenuated by a software developer’s instrumental relative advantage perceptions of the SDPM practice.	Not Supported

Chapter VII: Discussion

7.1. Summary of Findings

While the majority of hypothesized relationships were found to be non-significant, there are several important findings that surfaced in this research. First, software developers were shown to customize the SDPM practice and perceive its use as being faithful to expectations when task-related value perceptions were strong. This finding supports previous software development research which has stressed the importance of instrumental relative advantage in understanding developer behavior (Hardgrave et al. 2003), but provides an extension by assessing the nature of usage that is encouraged through these types of value perceptions. Specifically, a belief that the SDPM practice would help the developer complete project tasks seemed to encourage mindful utilization of the practice which encouraged (a) a deeper and more substantive use of the SDPM practice and (b) a consistency in belief that the practice is used as expected by the organization.

Second, practice regulation seems to be an important means of encouraging customized use of the SPDM practice. Rather than forcing simple compliance with existing methods for using the SDPM practice, a strong regulative environment seemed to provide developers with stability such that they were not burdened with knowing if or how to use the practice, but rather were permitted to focus on how to use the practice effectively within a given project context. This is an important finding since conventional wisdom often suggests that highly controlled environments inhibit individual creativity (Ford and Gioia 2000). For software development environments, this suggests that the mandate of structured practices can actually encourage a more meaningful use within development projects. An important caveat to this assertion is that perceptions of SPDM practice complexity were shown to reduce the likelihood of practice customization. It is possible that complex practices require a developer to invest resources in understanding and implementing the practice, and as such detract from their ability to adapt it within a given project.

Third, factors encouraging a developer to consider the task in greater detail, such as perceptions of instrumental relative advantage and operational exigencies, seemed to have fostered a developer's perceptions of practice faithfulness. On the other hand, perceptions that focused on factors outside the current project such as pressure exerted by the institutional environment or political value, had very little impact on faithfulness perceptions. For example, while a strong regulative environment encouraged customization, it did not seem to help motivate a developer to view usage behavior as being more consistent with organizational expectations. As such, findings suggest that developers were most motivated to reconcile their usage behavior with organizational expectations when encountering beliefs that forced greater consideration of the task itself. This assertion is consistent with cognitive dissonance theories positing that individuals will seek consonance in beliefs when conflicting ideas are strong enough to create mental tension (Festinger 1962).

7.1.1. Implications for Theory

In addition to the findings mentioned above, several implications for theory arise from this research. Most significantly, utilizing an "organization-as-institution" perspective to develop an institutional environment categorization scheme provides a novel means of evaluating individual behavior in light of both the formal and informal institutional environments within an organization. Using this scheme, future research could draw on Oliver's work on strategic responses to institutional processes (1991) to better understand how a developer will use the SDPM practice for a given project. For example, acquiescence usage behavior (pictured as habit, imitation or conscious compliance) could be predicted by theorizing about the formal and informal dimensions of the institutional environment (i.e., most likely in environments where formal norms for usage are quite strong but informal norms are weak). Building on this classification scheme could provide a rich means of understanding post-adoptive behavior and also extends institutional literature by formalizing the characterization of both formal and informal pressure.

This research also offers a unique perspective in post-adoptive research by looking beyond simple usage and delving into the nature of usage. Specifically, future research can benefit from evaluating ***how*** usage is exhibited through concepts similar to customization and faithfulness. In addition, addressing relative advantage perceptions through both instrumental and political manifestations provides an essential perspective when investigating factors that motivate organizational workers. Researchers can draw on the bifurcated view of relative advantage utilized in this research to increase the viability of adoption research in organizations.

Further, the unexpectedly strong negative correlation demonstrated between customized and faithful use presents some important opportunities for research examining the nature of post-adoptive use. Three possible explanations exist that might work to explain the divergent movement of customized and faithful SDPM practice usage. First, a causal relationship might exist between the two variables, where one is expected to directly influence the other. A second possible explanation is that both variables underlie some higher-level construct, an example being where faithfulness and customization represent two ends on a continuum. A final explanation for the strong negative correlation between customized and faithful usage is that some third variable is influencing both and forcing them in opposite directions.

As a means of understanding the divergent relationship, a post-hoc analysis was conducted that considered the correlation between customization and faithfulness in greater detail. To achieve this, the sample was divided into high, medium and low customization categories where cutoff points were specified by using ½ standard deviations from the sample mean value. Descriptives for this classification scheme are included in Table 7.1-1. In order to verify that resulting customization levels were distinct, a univariate GLM procedure was run and contrasts between levels were evaluated. The overall model demonstrated a significant difference in mean customization between levels (F=222.009, $p \leq 0.01$) and contrasts showed a significant difference between each level ($p < 0.01$).

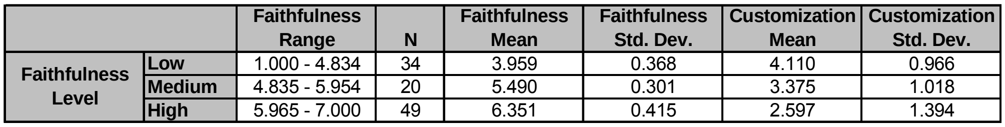
Table 7.1-1: Customization Classification Details

		Customization Range	N	Customization Mean	Customization Std. Dev.	Faithfulness Mean	Faithfulness Std. Dev.
Customization Level	Low	1.000 - 2.564	43	1.901	0.427	6.088	0.813
	Medium	2.565 - 3.934	17	3.265	0.348	5.424	0.897
	High	3.935 - 7.000	43	4.587	0.777	4.688	1.055

Consistent with the negative correlation between customization and faithfulness evidenced in this research, Table 7.1-1illustrates that faithfulness becomes less prevalent as the customization level moves from low to high. However, evaluating the correlation between customization and faithfulness within each level reveals that something more complex is occurring. Within environments of low customization, a significantly negative correlation exists between the two variables (r=-0.492, $p \leq 0.01$) and within environments of medium customization, a weak and non-significant negative correlation exists (r=-0.031, p=0.905). However, within the high customization group, a significantly positive correlation exists between customization and faithfulness (r=0.369, $p \leq 0.05$). So, while the lowest level of faithfulness exists within the high customization group, increases in customization within this group are likely to evidence increases in usage faithfulness.

A similar process of classification was used for faithfulness (as reported in Table 7.1-2), but correlation values between customization and faithfulness within each resulting level were found to be non-significant.

Table 7.1-2: Faithfulness Classification Details



		Faithfulness Range	N	Faithfulness Mean	Faithfulness Std. Dev.	Customization Mean	Customization Std. Dev.
Faithfulness Level	Low	1.000 - 4.834	34	3.959	0.368	4.110	0.966
	Medium	4.835 - 5.954	20	5.490	0.301	3.375	1.018
	High	5.965 - 7.000	49	6.351	0.415	2.597	1.394

Revisiting the three possible explanations underlying a negative relationship between customized and faithful use, customization classification results provide some preliminary guidance. First, results suggest that there might be a negative causal relationship between customization and faithfulness on the whole, but the nature of the relationship could change when one considers the high versus low customization groups. So while a causal relationship is plausible, the nature of the relationship would seem to be contingent on additional factors (i.e., level of customization). The second suggestion of a higher-order construct is also still a possibility, although the suggestion that faithfulness and customization are two ends on a continuum is unlikely given the positive correlation that was demonstrated between faithful and customized use within the high customization group. The final possibility, where a spurious relationship exists between the two variables, can also not be ruled out either based on the existing data. Future research can extend this dissertation by further examining these three possible explanations, most promising of which seems to be a complex causal relationship between customized and faithful use.

7.1.2. Implications for Practice

There are also several important lessons within the findings that can be applied to practice. First, this research has illustrated the importance of promoting the task-related value of practices within a software development context. In order for a SDPM practice to be used meaningfully, a developer must be convinced of its value with regards to the task at hand. As illustrated in the results, developers who perceived the practice to be more valuable for use within a practice were more likely to customize the practice to meet project-related needs. Without a belief that the practice is important for the project, developers are unlikely to enact a deep and effective use of the practice. Second, environments where practice use is defined and mandated can provide developers with a structure that encourages use of the practice which fits specific project needs. Rather than constraining developer behavior, strongly regulated environments can serve to free developers so that their focus can be on completing projects effectively. Third, organizations can encourage practice customization and faithfulness by simply encouraging developers to be mindful regarding the practice. Within the study, this included the induction of operational exigencies and instrumental relative advantage which both seemed to encourage a focus on the task itself. However, other mechanisms such as limiting developer work to one project at a time or requiring frequent structured interaction within the project team might promote a task focus and in turn induce usage that is both faithful and adapted to meet project needs.

Finally, the negative relationship between customized and faithful SDPM practice use presents an interesting situation for organizations. Software development groups within an organization have a strong incentive to promote faithful use so that they can ensure collective action. Collective action is especially important in situations where large development efforts are being undertaken. In an environment where multiple developers are focused on creating one product, unity is essential. However, organizations are also interested in ensuring that developers are working as effectively as possible within each project they have been assigned. Customization of the SDPM practice allows for a targeted use of the practice and provides an essential opportunity for innovation within the development process. Results of a negative relationship between customization and faithfulness suggest that organizations must be careful that the promotion of a specific outcome (i.e., collective action) does not drain the existence of the other outcome (i.e., innovation) and as a consequence impede the group's long-term ability to deliver quality software.

7.2. Limitations of the Study and Directions for Future Research

While this dissertation has provided some important perspectives for SDPM practice use, there are several limitations which must be noted. First and foremost, the small number of software developer responses prevents a robust evaluation of the research hypotheses. Specifically, a limited sample size attenuates the statistical power required to properly test proposed hypotheses within a multiple regression procedure, and prevents the use of potentially valuable statistical procedures such as Hierarchical Linear Modeling (Raudenbush and Bryk 2002).

Further, analyzing data that incorporates developer responses for both SDPM practices creates a situation where independence, a major assumption of multiple regression analysis, is violated. This is particularly problematic since results might be a function of dependencies within the data rather than being derived through theorized relationships. Despite efforts to minimize the harmful effects of non-independence of error terms, the problem clearly persists within the data analyzed in this research and has not been dealt with in an ideal fashion. Data collection efforts are still ongoing which will enable the use of methods, such as between subjects analysis, that provide a more appropriate and robust evaluation of hypothesized relationships.

Additionally, data was not collected regarding whether a software developer needed to use the SDPM practice during the month period between surveys. The research design employed within this dissertation failed to account for the frequent situations where a developer might not need to use the practice, such as when a change to project scope is not encountered during the early stages of a project. Future research will directly request information from the project manager (not directly from the software developer) regarding the degree to which SDPM practice use was actually needed during the timeframe. This data will allow for the difference in practice usage between situations where use was deemed necessary by management and situations where it was not needed.

Further, the research design employed in this research failed to collect information regarding how peers expected the software developer to use the SDPM practice. Focusing only on peer influence through social norms for usage prevented the examination of faithful usage with respect to peer expectations. Specifically, hypotheses could not be made regarding the impact of Informal Institutionalized environments on faithful usage since data did not provide information regarding the type of usage expected from peers. Future research will develop a measure of expectations for use so that the model can be extended to include the impact of Informal Institutionalization on a developer's faithful use of the SDPM practice.

In addition, the role of operational exigencies in the usage process failed to yield very meaningful results. Future research will expand on the exigency construct to understand ways in which external pressures influence developer behavior within the context of a software development project.

Further, the conceptualization of political relative advantage was stunted in this research and deserves more focused attention. Future work must address the positioning of political relative advantage with respect to management *and* co-workers. Specifically, future research must strive to understand developer perceptions of the relative importance of meeting management versus co-workers expectations for use in order to unravel the effects of political relative advantage.

Finally, a study of practice use to some degree implies that the use of the SDPM practice is meaningful for project success. However, SDPM practice use will not guarantee project success, and as such is only an entry point into the more important issue of factors that influence software development project success. Issues relating to this include identifying SDPM practices which can encourage project success as well as contextual factors which enable the success of specific practices. Software development project work is extremely complicated and varies substantially from project to project and organization to organization. While this study examined general developer usage behavior across a number of organizations, future research on SD project success would benefit greatly by pursuing a more in-depth, qualitative approach within a software development environment where contextual issues such as organizational politics (i.e., external pressure, project legitimacy within the organization, impact of a senior management project champion), group dynamics (i.e., co-located and dispersed project teams, interpersonal relationships, power distribution) and structural factors (i.e., development methodology, programming platform, utilization of organizational employees and contract workers) were considered. In-depth qualitative methods are more apt to capture the fine grained issues that arise during a software development project and work to influence the success of a project.

7.3. Conclusion

The focus of this dissertation has been on understanding factors which influence how a software developer uses a SDPM practice within the context of a software development project. Findings illustrate that software developer customization of a SDPM practice is strongly impacted by their perceptions of the task-related value of the practice and encouraged by an institutional environment where both regulative and normative pressures encourage use. Further, findings demonstrated that faithful use of the SDPM practice was most impacted by both instrumental and political value perceptions of the practice. These findings offer an essential first step in addressing the importance of SDPM practice use in software development project work.

References

Adams, D.A., Nelson, R.R., and Todd, P.A. "Perceived Usefulness, Ease of Use, and Usage of Information Technology: A Replication," *MIS Quarterly* (16:2), June 1992, pp 227-248.

Agarwal, R. "Individual Acceptance of Information Technologies," in: *Framing the Domains of IT Management: Projecting the Future Through the Past,* R.W. Zmud (ed.), Pinnaflex Educational Resources, Inc., Cincinnati, OH, 2000, pp. 85-104.

Agarwal, R., and Prasad, J. "A Field Study of the Adoption of Software Process Innovations by Information Systems Professionals," *IEEE Transactions on Engineering Management* (47:3), August 2000, pp 295-308.

Aiken, L.S., and West, S.G. *Multiple Regression: Testing and Interpreting Interactions* Sage Publications, Newbury Park, CA, 1991.

Ajzen, I. (ed.) *From Intentions to Actions: A Theory of Planned Behavior*. Springer-Verlag, New York, 1985.

Ajzen, I. "The Theory of Planned Behavior," *Organizational Behavior and Human Decision Processes* (50:2), December 1991, pp 179-211.

Ajzen, I., and Madden, T.J. "Prediction of Goal-Directed Behavior: Attitudes, Intentions and Perceived Behavioral Control," *Journal of Experimental Social Psychology* (22:5) 1986, pp 453-474.

Austin, R.D. "The effects of time pressure on quality in software development: An agency model," *Information Systems Research* (12:2) 2001, p 195.

Baer, M., and Frese, M. "Innovation is not Enough: Climates for Initiative and Psychological Safety, Process Innovations, and Firm Performance," *Journal of Organizational Behavior* (24:1), February 2003, pp 45-68.

Baron, R.M., and Kenny, D.A. "The Moderator-Mediator Variable Distinction in Social Psychological Research: Conceptual, Strategic, and Statistical Considerations," *Journal of Personality and Social Psychology* (51:6), December 1986, pp 1173-1182.

Beaudry, A., and Pinsonneault, A. "Understanding User Responses to Information Technology: A Coping Model of User Adaptation," *MIS Quarterly* (29:3), September 2005, pp 493-524.

Bock, G.-W., Zmud, R.W., Kim, Y.-G., and Lee, J.-N. "Behavioral Intention Formation in Knowledge Sharing: Examining the Roles of Extrinsic Motivators, Social-Psychological Forces, and Organizational Climate," *MIS Quarterly* (29:1), March 2005, pp 87-111.

Brooks, F.P. *The Mythical Man-Month: Essays on Software Engineering,* (2nd ed.) Addison-Wesley Publishing Company, New York, NY, 1995.

Butler, B.S., and Gray, P.H. "Reliability, Mindfulness, and Information Systems," *MIS Quarterly* (30:2), June 2006, pp 211-224.

Cardinal, L.B. "Technological Innovation in the Pharmaceutical Industry: The Use of Organizational Control in Managing Research and

Development," *Organization Science* (12:1), January-February 2001, pp 19-36.

Cattell, R.B. *The Scientific Use of Factor Analysis in Behavioral and Life Sciences* Springer, New York, 1978.

Chang, M.K. "Predicting Unethical Behavior: A Comparison of the Theory of Reasoned Action and the Theory of Planned Behavior," *Journal of Business Ethics* (17:16), December 1998, pp 1825-1834.

Checkland, P. *Systems Thinking, Systems Practice* John Wiley & Sons Ltd., Great Britain, 1981, p. 330.

Chin, W.W., Gopal, A., and Salisbury, W.D. "Advancing the Theory of Adaptive Structuration: The Development of a Scale to Measure Faithfulness of Appropriation," *Information Systems Research* (8:4), December 1997, pp 342-367.

Cohen, J., and Cohen, P. *Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences* (2nd ed.) Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, 1983, p. 545.

Compeau, D., and Higgins, C.A. "Computer Self-Efficacy: Development of a Measure and Initial Test," *MIS Quarterly* (19:2), June 1995, pp 189-211.

Compeau, D., Higgins, C.A., and Huff, S. "Social Cognitive Theory and Individual Reactions to Computing Technology: A Longitudinal Study," *MIS Quarterly* (23:2), June 1999, pp 145-158.

Cooper, R.B., and Bhattacherjee, A. "Preliminary Evidence for the Effect of Automatic Responses to Authority on Information Technology Diffusion," *The DATA BASE for Advances in Information Systems* (32:3), Summer 2001, pp 36-50.

Davenport, T.H. "The Coming Commoditization of Processes," *Harvard Business Review* (83:6), June 2005, pp 100-108.

Davis, F.D. "Perceived Usefulness, Perceived Ease of Use, and User Acceptance in Information Technology," *MIS Quarterly* (13:3), September 1989, pp 319-339.

Davis, F.D., Bagozzi, R., and Warshaw, P. "User Acceptance of Computer Technology: A Comparison of Two Theoretical Models," *Management Science* (35:8) 1989, pp 982-1003.

Dean, J.W., Jr., and Sharfman, M.P. "The Relationship Between Procedural Rationality and Political Behavior in Strategic Decision Making," *Decision Sciences* (24:6), November/December 1993, pp 1069-1083.

Dean, J.W., Jr., and Sharfman, M.P. "Does Decision Process Matter? A Study of Strategic Decision-Making Effectiveness," *Academy of Management Journal* (39:2), April 1996, pp 368-396.

DeGrace, P., and Stahl, L.H. *Wicked Problems, Righteous Solutions: A Catalogue of Modern Software Engineering Paradigms* Prentice Hall, Englewood Cliffs, NJ, 1990.

DeMarco, T., and Lister, T. *Peopleware: Productive Projects and Teams*, (2nd ed.) Dorset House Publishing Company, Inc., New York, NY, 1999.

Dennis, A.R., and Reinicke, B.A. "Beta Versus VHS and the Acceptance of Electronic Brainstorming Technology," *MIS Quarterly* (28:1), March 2004, pp 1-20.

Dennis, A.R., Wixom, B.H., and Vandenberg, R.J. "Understanding Fit and Appropriation Effects in Group Support Systems via Meta-Analysis," *MIS Quarterly* (25:2), June 2001, pp 167-193.

DeSanctis, G., and Poole, M.S. "Capturing the Complexity in Advanced Technology Use: Adaptive Structuration Theory," *Organization Science* (5:2), May 1994, pp 121-147.

Dillman, D.A. *Mail and Internet Surveys: The Tailored Design Method*, (2nd ed.) John Wiley & Sons, Inc., New York, NY, 2000, p. 480.

DiMaggio, P.J., and Powell, W.W. "The Iron Cage Revisited: Institutional Isomorphism and Collective Rationality in Organizational Fields," *American Sociological Review* (48:2), April 1983, pp 147-160.

Doty, D.H., and Glick, W.H. "Common Methods Bias: Does Common Methods Variance Really Bias Results," *Organizational Research Methods* (1:4) 1998, pp 374-406.

Dutton, J.E., and Dukerich, J.M. "Keeping an Eye on the Mirror: Image and Identity in Organizational Adaptation," *Academy of Management Journal* (34:3), September 1991, pp 517-554.

Edland, A. "Time Pressure and the Application of Decision Rules: Choices and Judgments among Multi-attribute Alternatives," *Scandinavian Journal of Psychology* (35) 1994, pp 281-291.

Eisenhardt, K.M., and Bourgeois, L.J., III "Politics Of Strategic Decision Making In High-Velocity Environments: Toward a Midrange Theory," *Academy of Management Journal* (31:4), December 1988, pp 737-770.

Eisenhardt, K.M., and Zbaracki, M.J. "Strategic Decision Making," *Strategic Management Journal* (13) 1992, p 17.

Fabrigar, L.R., Wegener, D.T., MacCallum, R.C., and Strahan, E.J. "Evaluating the Use of Exploratory Factor Analysis in Psychological Research," *Psychological Methods* (4:3) 1999, pp 272-299.

Feldman, M.S., and Pentland, B.T. "Reconceptualizing Organizational Routines as a Source of Flexibility and Change," *Administrative Science Quarterly* (48:1), March 2003, pp 94-118.

Festinger, L. *A Theory of Cognitive Dissonance* Stanford University Press, Stanford, CA, 1962, p. 291.

Fiol, C.M., and O'Connor, E.J. "Waking Up! Mindfulness in the Face of Bandwagons," *Academy of Management Review* (28:1), January 2003, pp 54-70.

Fishbein, M., and Ajzen, I. *Belief, Attitude, Intention and Behavior: An Introduction to Theory and Research* Addison-Wesley, Reading, MA, 1975.

Fitzgerald, B. "The Use of Systems Development Methodologies in Practice: A Field Study," *Information Systems Journal* (7:3), July 1997, pp 201-212.

Fitzgerald, B. "An Empirical Investigation into the Adoption of Systems Development Methodologies," *Information & Management* (34:6), December 1998, pp 317-328.

Ford, C.M., and Gioia, D.A. "Factors Influencing Creativity in the Domain of Managerial Decision Making," *Journal of Management* (26:4), August 2000, pp 705-732.

Ford, J.K., MacCallum, R.C., and Tait, M. "The Application of Exploratory Factor Analysis in Applied Psychology: A Critical Review and Analysis," *Personnel Psychology* (39) 1986, pp 291-314.

Fornell, C., and Larcker, D.F. "Evaluating Structural Equation Models with Unobservable Variables and Measurement Error," *Journal of Marketing Research* (18:1), February 1981, pp 39-50.

Fredrickson, J.W., and Iaquinto, A.L. "Inertia and Creeping Rationality in Strategic Decision Processes," *Academy of Management Journal* (32:3), September 1989, pp 516-542.

Fulk, J. "Social Construction of Communication Technology," *Academy of Management Journal* (36:5), October 1993, pp 921-950.

Gerbing, D.W., and Anderson, J.C. "An Updated Paradigm for Scale Development Incorporating Unidimensionality and its Assessment," *Journal of Marketing Research* (25:2), May 1988, pp 186-192.

Gilb, T., and Finzi, S. *Principles of Software Engineering Management* Addison-Wesley Publishing Company, Workingham, England, 1988.

Goodrick, E., and Salancik, G.R. "Organizational discretion in responding to institutional practices: Hospitals and cesarean births," *Administrative Science Quarterly* (41:1), March 1996, pp 1-28.

Goodstein, J.D. "Institutional pressures and strategic responsiveness: Employer involvement in work-family issues," *Academy of Management Journal* (37:2), April 1994, pp 350-382.

Gorsuch, R.L. *Factor Analysis*, (2nd ed.) Lawrence Erlbaum, Hillsdale, NJ, 1983.

Goulielmos, M. "Systems Development Approach: Transcending Methodology," *Information Systems Journal* (14:4), October 2004, pp 363-386.

Grossman, D. *On Killing : The Psychological Cost of Learning to Kill in War and Society* Back Bay Books Boston, MA, 1995, p. 400.

Hair, J.F., Anderson, R.E., Tatham, R.L., and Black, W.C. *Multivariate Data Analysis*, (5th ed.) Prentice-Hall, Inc., Upper Saddle River, NJ, 1998, p. 730.

Hardgrave, B.C., Davis, F.D., and Riemenschneider, C.K. "Investigating Determinants of Software Developers' Intentions to Follow

Methodologies," *Journal of Management Information Systems* (20:1), Summer 2003, pp 123-151.

Hartman, F., and Ashrafi, R.A. "Project Management in the Information Systems and Information Technologies Industries," *Project Management Journal* (33:3), September 2002, pp 5-15.

Hartwick, J., and Barki, H. "Explaining the role of user participation in information system use," *Management Science* (40:4), April 1994, pp 440-465.

Igbaria, M., Guimaraes, T., and Davis, G.B. "Testing the Determinants of Microcomputer Usage via a Structural Equation Model," *Journal of Management Information Systems* (11:4), Spring 1995, pp 87-114.

Igbaria, M., Zinatelli, N., Cragg, P., and Cavaye, A.L.M. "Personal Computing Acceptance Factors in Small Firms: A Structural Equation Model," *MIS Quarterly* (21:3), September 1997, pp 279-305.

Iivari, J., and Huisman, M. "The Relationship Between Organizational Culture and the Deployment of Systems Development Methodologies," *MIS Quarterly* (31:1), March 2007, pp 35-58.

Jasperson, J.S., Carter, P.E., and Zmud, R.W. "A Comprehensive Conceptualization of Post-Adoptive Behaviors Associated with Information Technology Enabled Work Systems," *MIS Quarterly* (29:3), September 2005, pp 525-557.

Jaworski, B.J., Stathakopoulos, V., and Krishnan, H.S. "Control Combinations in Marketing: Conceptual Framework and Empirical Evidence," *Journal of Marketing* (57:1), January 1993, pp 57-69.

Jeyaraj, A., Rottman, J.W., and Lacity, M.C. "A Review of the Predictors, Linkages, and Biases in IT Innovation Adoption Research," *Journal of Information Technology* (21:1), February 2006, pp 1-23.

Johnson, R., Hardgrave, B.C., and Doke, E.R. "An Industry Analysis of Developer Beliefs About Object-oriented Systems Development," *DATA BASE for Advances in Information Systems* (30:1) 1999, pp 47-64.

Karahanna, E., Straub, D.W., and Chervany, N.L. "Information Technology Adoption Across Time: A Cross-Sectional Comparison of Pre-Adoption and Post-Adoption Beliefs," *MIS Quarterly* (23:2), June 1999, pp 183-213.

Katz, M.L., and Shapiro, C. "Technology Adoption in the Presence of Network Externalities," *Journal of Political Economy* (94:4) 1986, pp 822-841.

Kerlinger, F.N., and Lee, H.B. *Foundations of Behavioral Research*, (Fourth ed.) Harcourt College Publishers, Fort Worth, TX, 2000, p. 890.

Khalifa, M., and Verner, J.M. "Drivers for Software Development Method Usage," *IEEE Transactions on Engineering Management* (47:3), August 2000, pp 360-369.

Kirsch, L.J. "Deploying Common Systems Globally: The Dynamics of Control," *Information Systems Research* (15:4), December 2004, pp 374-395.

Kostova, T., and Roth, K. "Adoption of an organizational practice by subsidiaries of multinational corporations: Institutional and relational effects," *Academy of Management Journal* (45:1), February 2002, pp 215-233.

Kutner, M.H., Nachtsheim, C.J., Neter, J., and Li, W. *Applied Linear Statistical Models*, (Fifth ed.) McGraw Hill Higher Education, New York, 2004.

Langer, E.J., and Moldoveanu, M. "The Construct of Mindfulness," *Journal of Social Issues* (56:1) 2000, pp 1-9.

Leary, M.R., and Kowalski, R.M. "Impression Management: A Literature Review and Two-Component Model," *Psychological Bulletin* (107:1), January 1990, pp 34-47.

Leonard-Barton, D., and Deschamps, I. "Managerial Influence in the Implementation of a New Technology," *Management Science* (34:10), October 1988, pp 1252-1265.

Levinthal, D., and Rerup, C. "Crossing an Apparent Chasm: Bridging Mindful and Less-Mindful Perspectives on Organizational Learning," *Organization Science* (17:4), July-August 2006, pp 502-513.

Lewis, W., Agarwal, R., and Sambamurthy, V. "Sources of Influence on Beliefs About Information Technology Use: An Empirical Study of Knowledge Workers," *MIS Quarterly* (27:4), December 2003, pp 657-678.

Majchrzak, A., Rice, R.E., Malhotra, A., King, N., and Ba, S. "Technology Adaptation: The Case of a Computer-Supported Inter-Organizational Virtual Team," *MIS Quarterly* (24:4) 2000, pp 569-600.

Mathieson, K. "Predicting User Intentions: Comparing the Technology Acceptance Model with the Theory of Planned Behavior," *Information Systems Research* (2:3), September 1991, pp 173-191.

McCarthy, J. *Dynamics of Software Development* Microsoft Press, Redmond, WA, 1995.

McConnell, S. *Code Complete*, (2nd ed.) Microsoft Press, Redmond, WA, 2004.

McConnell, S.C. *Software Project Survival Guide* Microsoft Press, Redmond, WA, 1998.

Meyer, J.W., and Rowan, B. "Institutionalized Organizations: Formal Structure as Myth and Ceremony," *American Journal of Sociology* (83:2), September 1977, pp 340-363.

Mintzberg, H. *The Structuring of Organizations* Prentice-Hall, Englewood Cliffs, NJ, 1978, p. 512.

Moore, G.C., and Benbasat, I. "Development of an Instrument to Measure the Perceptions of Adopting an Information Technology Innovation," *Information Systems Research* (2:3) 1991, pp 192-222.

Nicolaou, A.I. "Social Control in Information Systems Development," *Information Technology & People* (12:2) 1999, pp 130-147.

NIST "The Economic Impacts of Inadequate Infrastructure for Software Testing," Planning Report 02-3, U.S. National Institute of Standards & Technology, Gaithersburg, MD, pp. 1-309.

Nonaka, I. "A Dynamic Theory of Organizational Knowledge Creation," *Organization Science* (5:1), February 1994, pp 14-37.

Nunnally, J.C., and Bernstein, I. *Psychometric Theory*, (3rd ed.) McGraw Hill, New York, 1994, p. 736.

O'Keefe, D.J. *Persuasion: Theory & Research*, (2nd ed.) Sage Publications, Inc., Thousand Oaks, CA, 2002, p. 365.

Ohly, S., Sonnentag, S., and Pluntke, F. "Routinization, Work Characteristics and their Relationships with Creative and Proactive Behaviors," *Journal of Organizational Behavior* (27:3), May 2006, pp 257-279.

Oliver, C. "Strategic Responses to Institutional Processes," *Academy of Management Review* (16:1), January 1991, pp 145-179.

Orlikowski, W.J., and Barley, S.R. "Technology and Institutions: What Can Research on Information Technology and Research on Organizations Learn from Each Other?," *MIS Quarterly* (25:2), June 2001, pp 145-165.

Orlikowski, W.J., Yates, J., Okamura, K., and Fujimoto, M. "Shaping Electronic Communication: The Metastructuring of Technology in the Context of Use," *Organization Science* (6:4), Jul/Aug 1995, pp 423-444.

Ouchi, W.G. "A Conceptual Framework for the Design of Organizational Control Mechanisms," *Management Science* (25:9), September 1979, pp 833-848.

Pascale, R.T., and Sternin, J. "Best Practices are Hard to Copy," *Harvard Business Review* (83:5), May 2005, p 79.

Paulk, M.C., Weber, C.V., Curtis, B., and Chrissis, M.B. (eds.) *The Capability Maturity Model: Guidelines for Improving the Software Process*. Addison-Wesley Publishing Co., Reading, MA, 1995.

Payne, J.W., Bettman, J.R., and Johnson, E.J. "Adaptive Strategy Selection in Decision Making," *Journal of Experimental Psychology: Learning, Memory and Cognition* (14) 1988, pp 534-522.

Payne, J.W., Bettman, J.R., and Luce, M.F. "When Time Is Money: Decision Behavior Under Opportunity-Cost Time Pressure," *Organizational Behavior and Human Decision Processes* (66:2), May 1996, pp 131-152.

Perlow, L.A., Okhuysen, G.A., and Repenning, N.P. "The Speed Trap: Exploring the Relationship Between Decision Making and Temporal

Context," *Academy of Management Journal* (45:5), October 2002, pp 931-955.

Petty, R.E., and Cacioppo, J.T. "The Effects of Involvement on Responses to Argument Quantity and Quality: Central and Peripheral Routes to Persuasion," *Journal of Personality and Social Psychology* (46:1), January 1984, pp 69-81.

PMI *A Guide to the Project Management Body of Knowledge (PMBOK© Guide)*, (Third ed.) Project Management Institute, Inc., Newton Square, PA, 2003.

Purvis, R.L., Sambamurthy, V., and Zmud, R.W. "The Assimilation of Knowledge Platforms in Organizations: An Empirical Investigation," *Organization Science* (12:2), Mar/Apr 2001, pp 117-135.

Raudenbush, S.W., and Bryk, A.S. *Hierarchical Linear Models: Applications and Data Analysis Methods*, (2nd ed.) Sage Publications, Inc., Thousand Oaks, CA, 2002, p. 504.

Repenning, N.P. "Understanding Fire Fighting in New Product Development," *The Journal of Product Innovation Management* (18:5), September 2001, pp 285-300.

Riemenschneider, C.K., Hardgrave, B.C., and Davis, F.D. "Explaining Software Developer Acceptance of Methodologies: A Comparison of Five Theoretical Models," *IEEE Transactions on Software Engineering* (28:12), December 2002, pp 1135-1145.

Robey, D., and Markus, M.L. "Rituals in Information Systems Design," *MIS Quarterly* (8:1), March 1984, pp 5-15.

Rogers, E.M. *Diffusion of Innovations*, (4th ed.) The Free Press, New York, NY, 1995, p. 519.

Rosenfeld, P.R., Giacalone, R.A., and Riordan, C.A. *Impression Management: Building and Enhancing Reputations at Work* International Thomson Business Press, New York, 2001, p. 288.

Royce, W. *Software Project Management: A Unified Framework* Addison-Wesley Publishing Company, Reading, MA, 1998.

Salancik, G.R., and Pfeffer, J. "A Social Information Processing Approach to Job Attitudes and Task Design," *Administrative Science Quarterly* (23:2), June 1978, pp 224-253.

Sanders, W.G., and Tuschke, A. "The Adoption of Institutionally Contested Organizational Practices: The Emergence of Stock Option Pay in Germany," *Academy of Management Journal* (50:1), February 2007, pp 33-56.

Scott, W.R. *Institutions and Organizations*, (2nd ed.) Sage Publications, Inc., Thousand Oaks, CA, 2001, p. 280.

Scott, W.R. *Organizations: Rational, Natural, and Open Systems*, (5th ed.) Prentice Hall, Upper Saddle River, NJ, 2003, p. 430.

Selznick, P. "Institutionalism 'Old' and 'New'," *Administrative Science Quarterly* (41:2) 1996, pp 270-277.

Shadish, W.R., Campbell, T.D., and Cook, D.T. *Experimental and Quasi-Experimental Designs for Generalized Causal Inference* Houghton Mifflin Company, Boston, MA, 2002, p. 623.

Simon, H.A. *Administrative Behavior*, (4th ed.) The Free Press, New York, NY, 1997.

Staw, B.M., and Epstein, L.D. "What Bandwagons Bring: Effects of Popular Management Techniques on Corporate Performance, Reputation, and CEO Pay," *Administrative Science Quarterly* (45:3), September 2000, pp 523-556.

Sternberg, R.J. "Images of Mindfulness," *Journal of Social Issues* (56:1) 2000, pp 11-26.

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

Thompson, R.L., Higgins, C.A., and Howell, J.M. "Personal Computing: Toward a Conceptual Model of Utilization," *MIS Quarterly* (15:1), March 1991, pp 125-143.

Tolbert, P.S., and Zucker, L.G. "Institutional Sources of Change in the Formal Structure of Organizations: The Diffusion of Civil Service Reform, 1880-1935," *Administrative Science Quarterly* (28:1), March 1983, pp 22-39.

Triandis, H.C. "Values, Attitudes and Interpersonal Behavior," Nebraska Symp. Motivation, 1979: Beliefs, attitudes and values, Lincoln, NE, 1980, pp. 159-295.

Tyre, M.J., and Orlikowski, W.J. "Windows of Opportunity: Temporal Patterns of Technological Adaptation in Organizations," *Organization Science* (5:1), February 1994, pp 98-118.

Van de Ven, A. "Central Problems in the Management of Innovation," *Management Science* (32:5), May 1986, pp 590-607.

van der Heijden, H. "User Acceptance of Hedonic Information Systems," *MIS Quarterly* (28:4), December 2004, pp 695-704.

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

Venkatesh, V., Morris, M.G., Davis, G.B., and Davis, F.D. "User Acceptance of Information Technology: Toward a Unified View," *MIS Quarterly* (27:3), Sep 2003, pp 425-478.

Warren, D.E. "Constructive and Destructive Deviance in Organizations," *Academy of Management Review* (28:4), October 2003, pp 622-632.

Wastell, D.G. "Learning dysfunctions in information systems development: Overcoming the social defenses with transitional objects," *MIS Quarterly* (23:4) 1999, p 581.

Weick, K.E., Sutcliffe, K.M., and Obstfeld, D. "Organizing for High Reliability: Processes of Collective Mindfulness," *Research in Organizational Behavior* (21) 1999, pp 81-123.

Westphal, J.D., Gulati, R., and Shortell, S.M. "Customization or Conformity? An Institutional and Network Perspective on the Content and Consequences of TQM Adoption," *Administrative Science Quarterly* (42:2), June 1997, pp 366-394.

Wheeler, B.C., and Valacich, J.S. "Facilitation, GSS, and Training as Sources of Process Restrictiveness and Guidance for Structured Group Decision Making: An Empirical Assessment," *Information Systems Research* (7:4), December 1996, pp 429-450.

Xiao, B., and Benbasat, I. "E-commerce Product Recommendation Agents: Use, Characteristics, and Impact," *MIS Quarterly* (31:1), March 2007, pp 137-209.

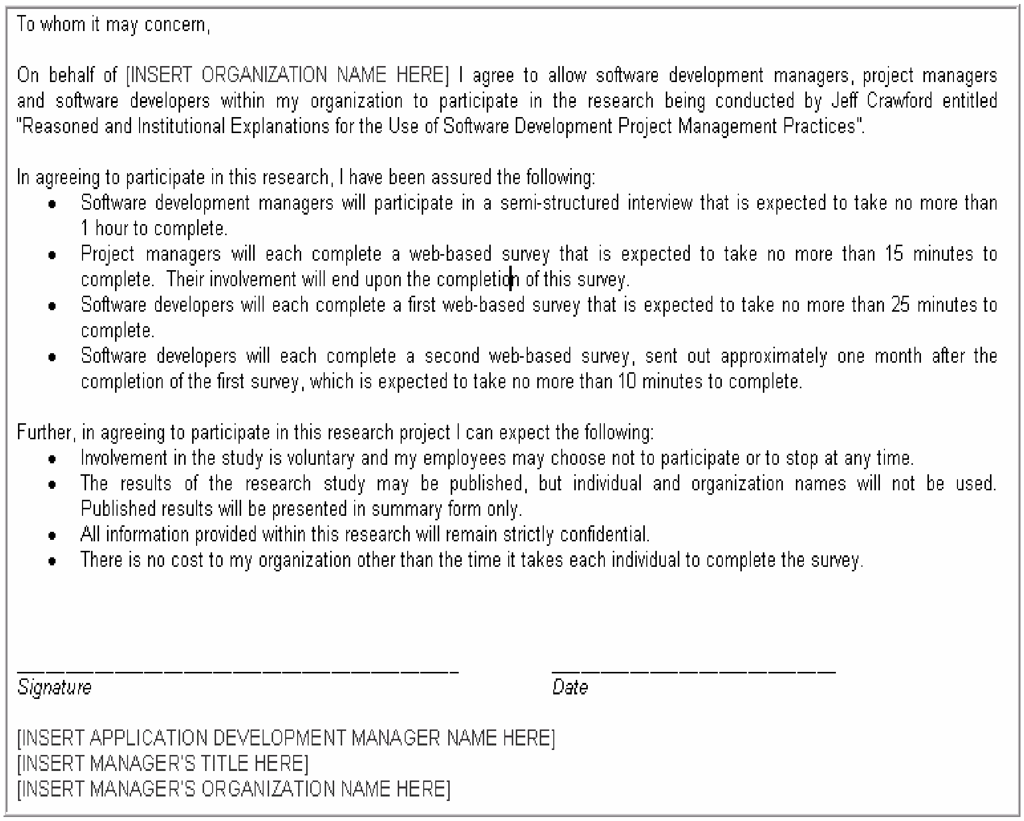
Yourdon, E. *Structured Walkthroughs*, (4th ed.) Yourdon Press, Englewood Cliffs, N.J., 1989.

Zakay, D. "The Impact of Time Perception Processes on Decision Making Under Time Stress," in: *Time Pressure and Stress in Human Judgment and Decision Making*, O. Svenson and A.J. Maule (eds.), Plenum Press, New York, NY, 1993, pp. 59-72.

Zigurs, I., and Buckland, B.K. "A Theory of Task/Technology Fit and Group Support Systems Effectiveness," *MIS Quarterly* (22:3), September 1998, pp 313-334.

Zucker, L.G. "Institutional Theories of Organization," *Annual Review of Sociology* (13:1) 1987, pp 443-464.

Appendix I - Organizational Informed Consent Form



To whom it may concern,

On behalf of [INSERT ORGANIZATION NAME HERE] I agree to allow software development managers, project managers and software developers within my organization to participate in the research being conducted by Jeff Crawford entitled "Reasoned and Institutional Explanations for the Use of Software Development Project Management Practices".

In agreeing to participate in this research, I have been assured the following:

- Software development managers will participate in a semi-structured interview that is expected to take no more than 1 hour to complete.
- Project managers will each complete a web-based survey that is expected to take no more than 15 minutes to complete. Their involvement will end upon the completion of this survey.
- Software developers will each complete a first web-based survey that is expected to take no more than 25 minutes to complete.
- Software developers will each complete a second web-based survey, sent out approximately one month after the completion of the first survey, which is expected to take no more than 10 minutes to complete.

Further, in agreeing to participate in this research project I can expect the following:

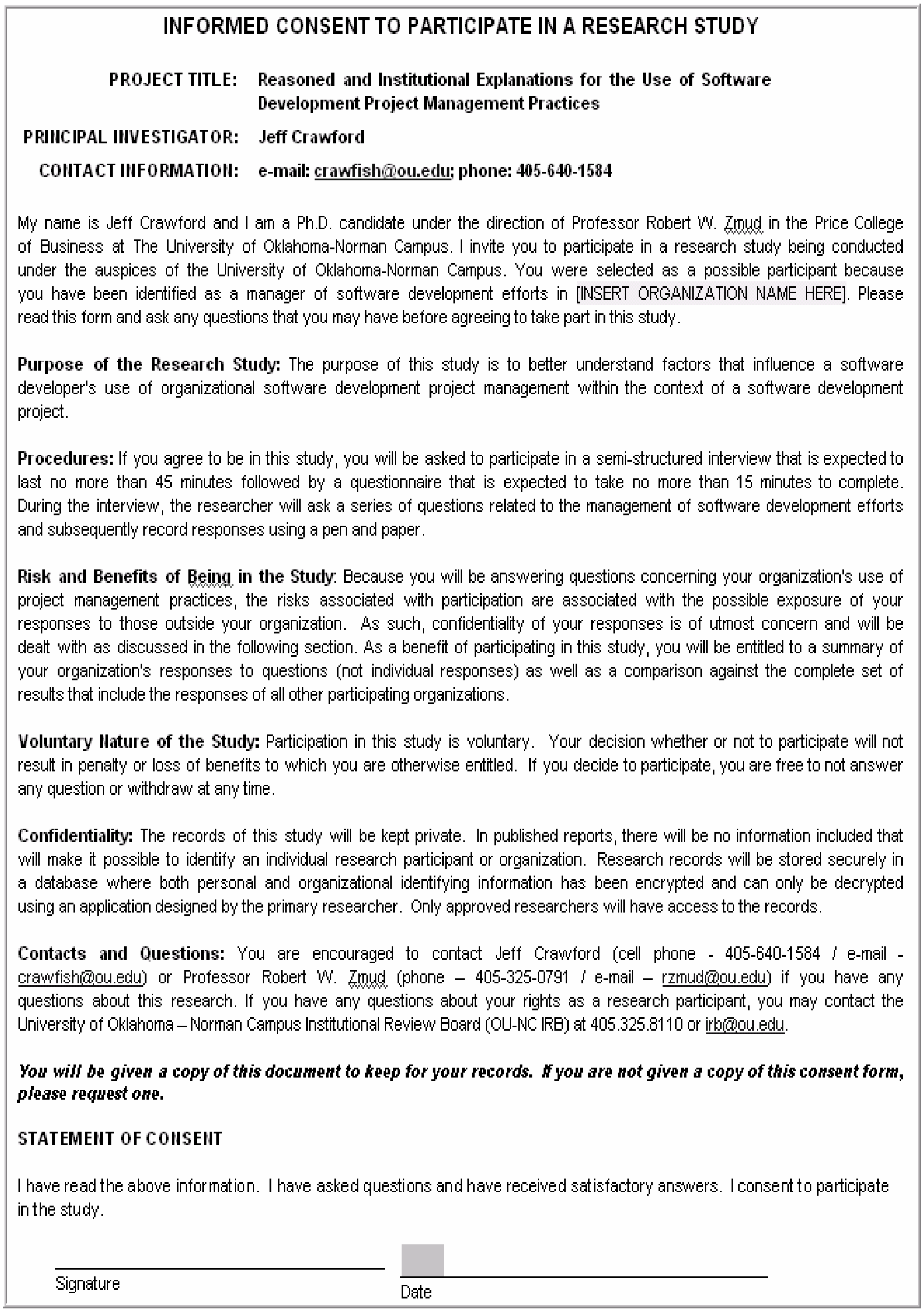
- Involvement in the study is voluntary and my employees may choose not to participate or to stop at any time.
- The results of the research study may be published, but individual and organization names will not be used. Published results will be presented in summary form only.
- All information provided within this research will remain strictly confidential.
- There is no cost to my organization other than the time it takes each individual to complete the survey.

______________________________________ ________________________

Signature *Date*

[INSERT APPLICATION DEVELOPMENT MANAGER NAME HERE]
[INSERT MANAGER'S TITLE HERE]
[INSERT MANAGER'S ORGANIZATION NAME HERE]

Appendix II - Software Development Manager Informed Consent Form



INFORMED CONSENT TO PARTICIPATE IN A RESEARCH STUDY

PROJECT TITLE:	Reasoned and Institutional Explanations for the Use of Software Development Project Management Practices
PRINCIPAL INVESTIGATOR:	Jeff Crawford
CONTACT INFORMATION:	e-mail: crawfish@ou.edu; phone: 405-640-1584

My name is Jeff Crawford and I am a Ph.D. candidate under the direction of Professor Robert W. Zmud in the Price College of Business at The University of Oklahoma-Norman Campus. I invite you to participate in a research study being conducted under the auspices of the University of Oklahoma-Norman Campus. You were selected as a possible participant because you have been identified as a manager of software development efforts in [INSERT ORGANIZATION NAME HERE]. Please read this form and ask any questions that you may have before agreeing to take part in this study.

Purpose of the Research Study: The purpose of this study is to better understand factors that influence a software developer's use of organizational software development project management within the context of a software development project.

Procedures: If you agree to be in this study, you will be asked to participate in a semi-structured interview that is expected to last no more than 45 minutes followed by a questionnaire that is expected to take no more than 15 minutes to complete. During the interview, the researcher will ask a series of questions related to the management of software development efforts and subsequently record responses using a pen and paper.

Risk and Benefits of Being in the Study: Because you will be answering questions concerning your organization's use of project management practices, the risks associated with participation are associated with the possible exposure of your responses to those outside your organization. As such, confidentiality of your responses is of utmost concern and will be dealt with as discussed in the following section. As a benefit of participating in this study, you will be entitled to a summary of your organization's responses to questions (not individual responses) as well as a comparison against the complete set of results that include the responses of all other participating organizations.

Voluntary Nature of the Study: Participation in this study is voluntary. Your decision whether or not to participate will not result in penalty or loss of benefits to which you are otherwise entitled. If you decide to participate, you are free to not answer any question or withdraw at any time.

Confidentiality: The records of this study will be kept private. In published reports, there will be no information included that will make it possible to identify an individual research participant or organization. Research records will be stored securely in a database where both personal and organizational identifying information has been encrypted and can only be decrypted using an application designed by the primary researcher. Only approved researchers will have access to the records.

Contacts and Questions: You are encouraged to contact Jeff Crawford (cell phone - 405-640-1584 / e-mail - crawfish@ou.edu) or Professor Robert W. Zmud (phone – 405-325-0791 / e-mail – rzmud@ou.edu) if you have any questions about this research. If you have any questions about your rights as a research participant, you may contact the University of Oklahoma – Norman Campus Institutional Review Board (OU-NC IRB) at 405.325.8110 or irb@ou.edu.

You will be given a copy of this document to keep for your records. If you are not given a copy of this consent form, please request one.

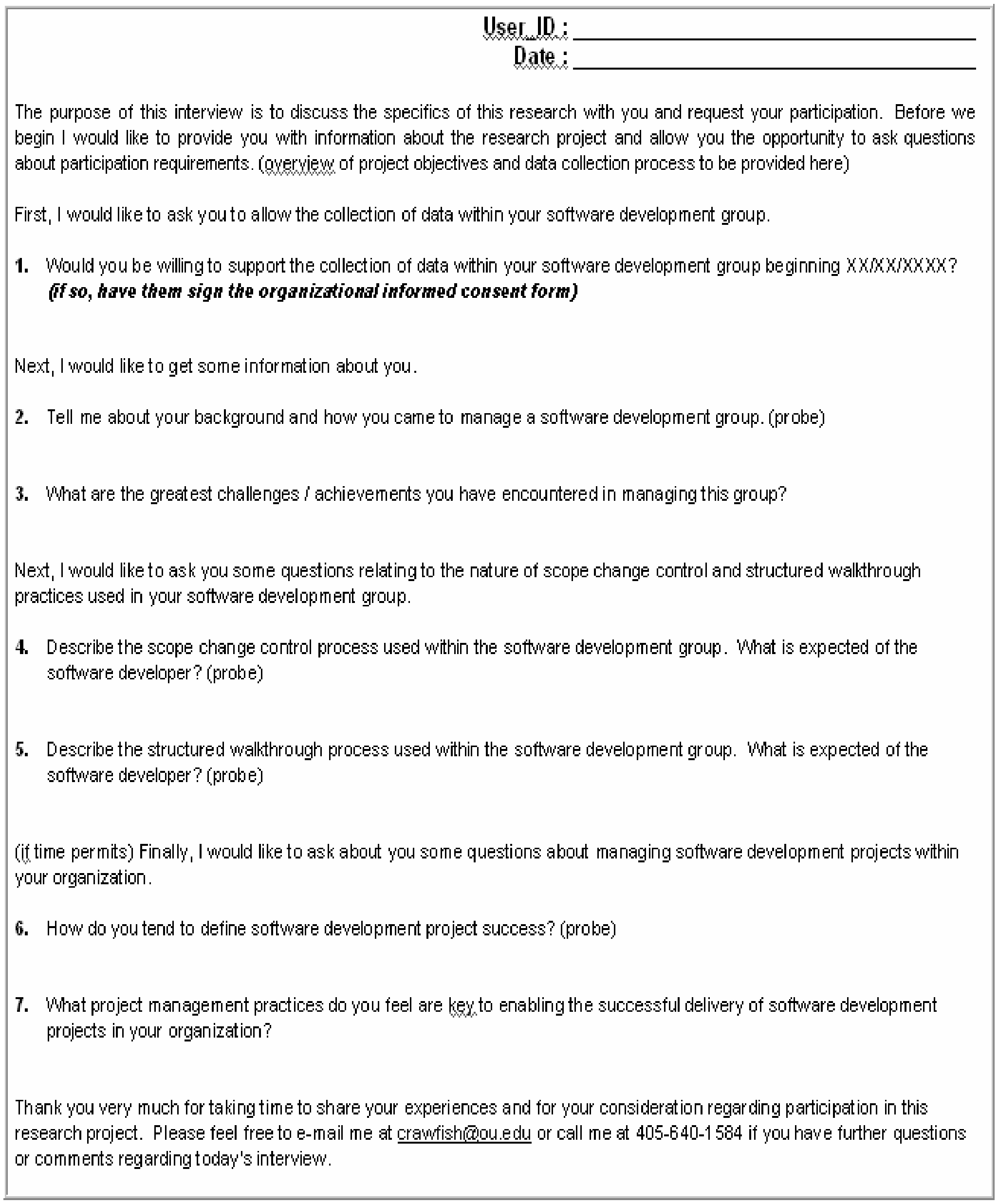
STATEMENT OF CONSENT

I have read the above information. I have asked questions and have received satisfactory answers. I consent to participate in the study.

____________________________ ____________________________

Signature Date

Appendix III - Software Development Manager Interview Schedule



User ID : ______________________

Date : ______________________

The purpose of this interview is to discuss the specifics of this research with you and request your participation. Before we begin I would like to provide you with information about the research project and allow you the opportunity to ask questions about participation requirements. (overview of project objectives and data collection process to be provided here)

First, I would like to ask you to allow the collection of data within your software development group.

1. Would you be willing to support the collection of data within your software development group beginning XX/XX/XXXX?
(if so, have them sign the organizational informed consent form)

Next, I would like to get some information about you.

2. Tell me about your background and how you came to manage a software development group. (probe)

3. What are the greatest challenges / achievements you have encountered in managing this group?

Next, I would like to ask you some questions relating to the nature of scope change control and structured walkthrough practices used in your software development group.

4. Describe the scope change control process used within the software development group. What is expected of the software developer? (probe)

5. Describe the structured walkthrough process used within the software development group. What is expected of the software developer? (probe)

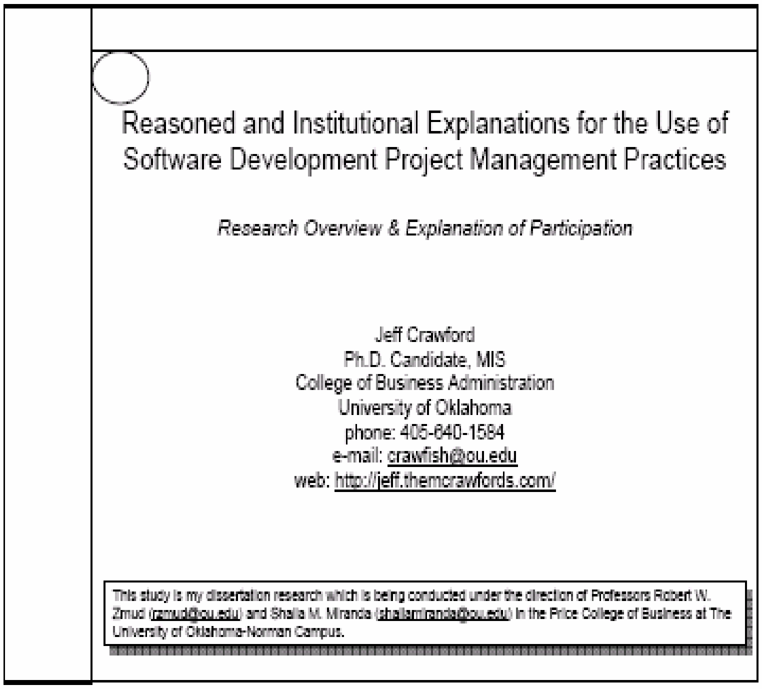
(if time permits) Finally, I would like to ask about you some questions about managing software development projects within your organization.

6. How do you tend to define software development project success? (probe)

7. What project management practices do you feel are key to enabling the successful delivery of software development projects in your organization?

Thank you very much for taking time to share your experiences and for your consideration regarding participation in this research project. Please feel free to e-mail me at crawfish@ou.edu or call me at 405-640-1584 if you have further questions or comments regarding today's interview.

Appendix IV – Participation Sales Presentation

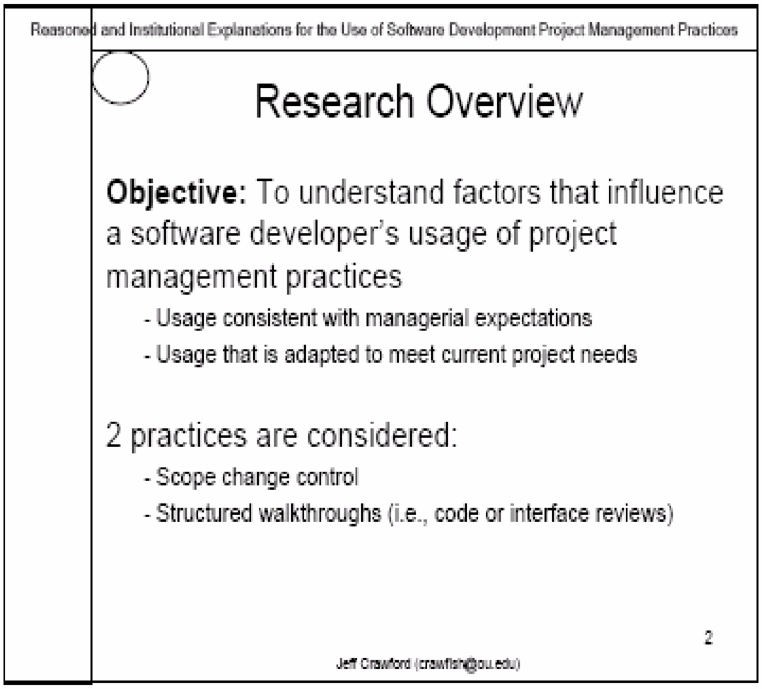


Reasoned and Institutional Explanations for the Use of Software Development Project Management Practices

Research Overview & Explanation of Participation

Jeff Crawford
Ph.D. Candidate, MIS
College of Business Administration
University of Oklahoma
phone: 405-640-1584
e-mail: crawfish@ou.edu
web: http://jeff.themcrawfords.com/

This study is my dissertation research which is being conducted under the direction of Professors Robert W. Zmud (rzmud@ou.edu) and Shaila M. Miranda (shailamiranda@ou.edu) in the Price College of Business at The University of Oklahoma-Norman Campus.



Reasoned and Institutional Explanations for the Use of Software Development Project Management Practices

Research Overview

Objective: To understand factors that influence a software developer's usage of project management practices

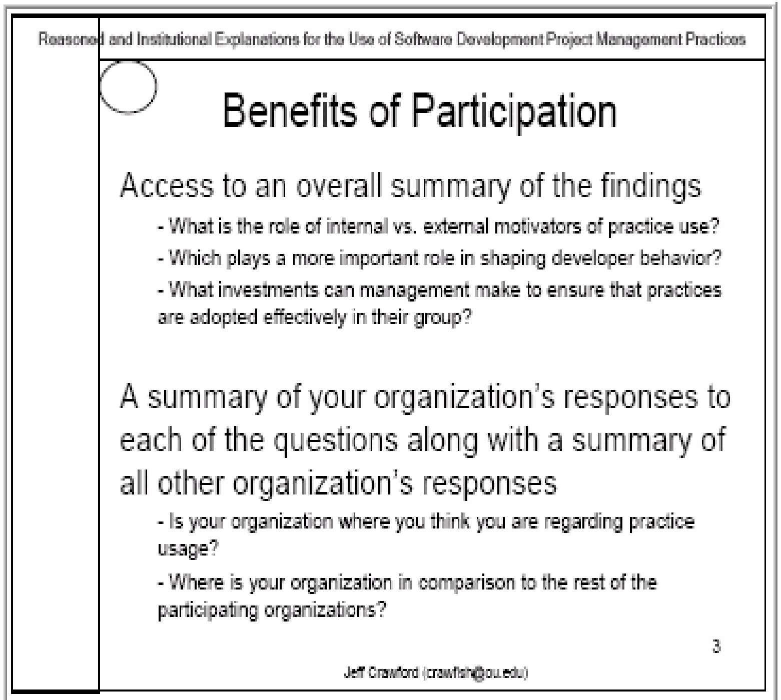
- Usage consistent with managerial expectations
- Usage that is adapted to meet current project needs

2 practices are considered:

- Scope change control
- Structured walkthroughs (i.e., code or interface reviews)

2

Jeff Crawford (crawfish@ou.edu)



Reasoned and Institutional Explanations for the Use of Software Development Project Management Practices

Benefits of Participation

Access to an overall summary of the findings

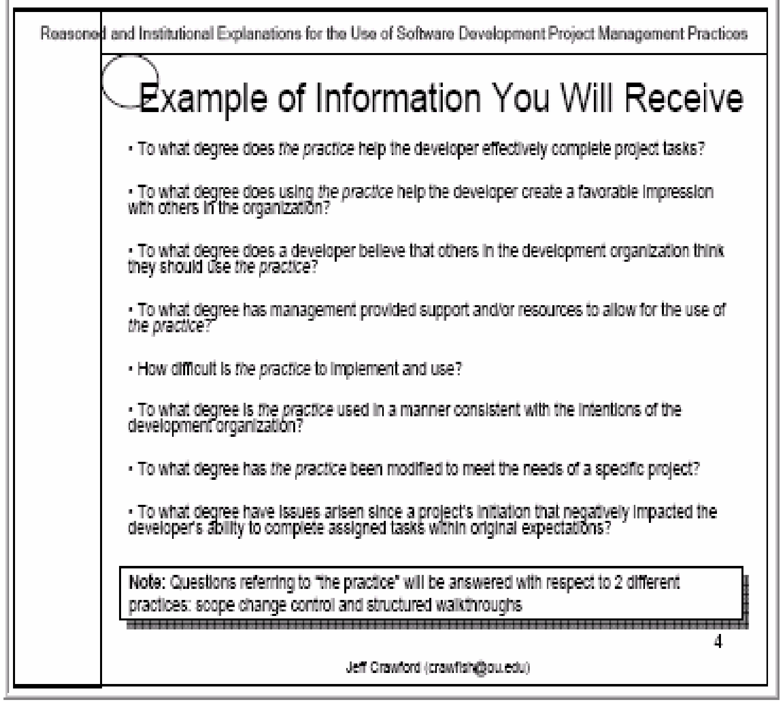
- What is the role of internal vs. external motivators of practice use?
- Which plays a more important role in shaping developer behavior?
- What investments can management make to ensure that practices are adopted effectively in their group?

A summary of your organization's responses to each of the questions along with a summary of all other organization's responses

- Is your organization where you think you are regarding practice usage?
- Where is your organization in comparison to the rest of the participating organizations?

3

Jeff Crawford (crawfish@ou.edu)



Reasoned and Institutional Explanations for the Use of Software Development Project Management Practices

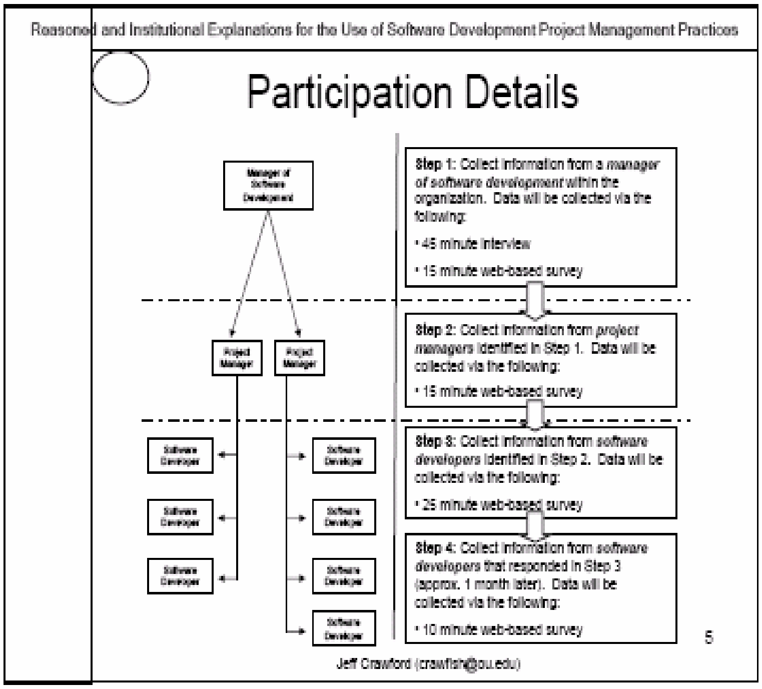
Example of Information You Will Receive

- To what degree does *the practice* help the developer effectively complete project tasks?
- To what degree does using *the practice* help the developer create a favorable impression with others in the organization?
- To what degree does a developer believe that others in the development organization think they should use *the practice*?
- To what degree has management provided support and/or resources to allow for the use of *the practice*?
- How difficult is *the practice* to implement and use?
- To what degree is *the practice* used in a manner consistent with the intentions of the development organization?
- To what degree has *the practice* been modified to meet the needs of a specific project?
- To what degree have issues arisen since a project's initiation that negatively impacted the developer's ability to complete assigned tasks within original expectations?

Note: Questions referring to "the practice" will be answered with respect to 2 different practices: scope change control and structured walkthroughs

4

Jeff Crawford (crawfish@ou.edu)


Manager of Software Development
Project Manager
Project Manager
Software Developer
Software Developer
Software Developer
Software Developer
Software Developer
Software Developer
Software Developer

Reasoned and Institutional Explanations for the Use of Software Development Project Management Practices

Participation Details

Step 1: Collect information from a *manager of software development* within the organization. Data will be collected via the following:

• 45 minute interview

• 15 minute web-based survey

Step 2: Collect information from *project managers* identified in Step 1. Data will be collected via the following:

• 15 minute web-based survey

Step 3: Collect information from *software developers* identified in Step 2. Data will be collected via the following:

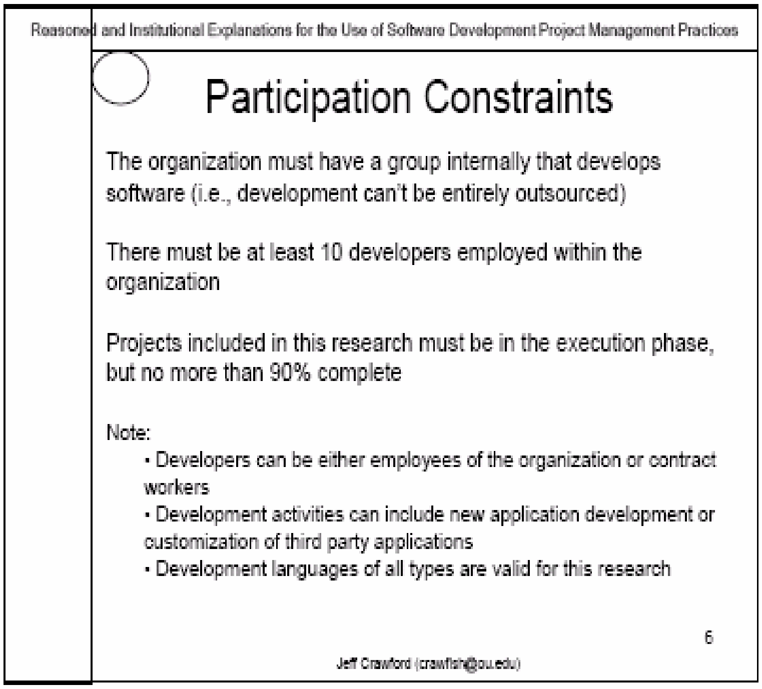
• 25 minute web-based survey

Step 4: Collect information from *software developers* that responded in Step 3 (approx. 1 month later). Data will be collected via the following:

• 10 minute web-based survey

5

Jeff Crawford (crawfish@ou.edu)



Reasoned and Institutional Explanations for the Use of Software Development Project Management Practices

Participation Constraints

The organization must have a group internally that develops software (i.e., development can't be entirely outsourced)

There must be at least 10 developers employed within the organization

Projects included in this research must be in the execution phase, but no more than 90% complete

Note:

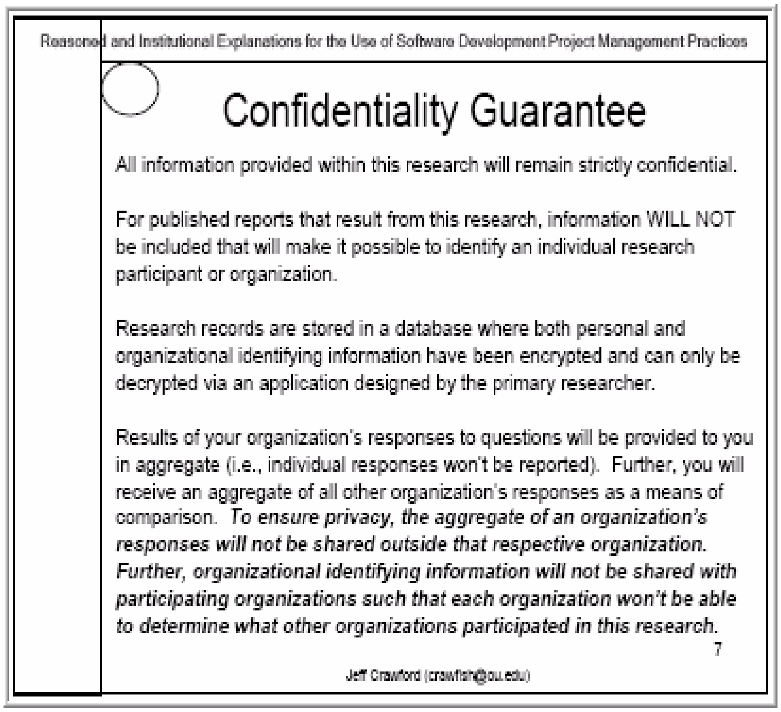
• Developers can be either employees of the organization or contract workers

• Development activities can include new application development or customization of third party applications

• Development languages of all types are valid for this research

6

Jeff Crawford (crawfish@ou.edu)



Reasoned and Institutional Explanations for the Use of Software Development Project Management Practices

Confidentiality Guarantee

All information provided within this research will remain strictly confidential.

For published reports that result from this research, information WILL NOT be included that will make it possible to identify an individual research participant or organization.

Research records are stored in a database where both personal and organizational identifying information have been encrypted and can only be decrypted via an application designed by the primary researcher.

Results of your organization's responses to questions will be provided to you in aggregate (i.e., individual responses won't be reported). Further, you will receive an aggregate of all other organization's responses as a means of comparison. ***To ensure privacy, the aggregate of an organization's responses will not be shared outside that respective organization. Further, organizational identifying information will not be shared with participating organizations such that each organization won't be able to determine what other organizations participated in this research.***

7

Jeff Crawford (crawfish@ou.edu)

Appendix V – Software Development Manager

Interview Schedule

User_ID : ____
Date : ____

The purpose of this interview is to discuss the specifics of this research with you and request your participation. Before we begin I would like to provide you with information about the research project and allow you the opportunity to ask questions about participation requirements. (overview of project objectives and data collection process to be provided here)

First, I would like to ask you to allow the collection of data within your software development group.

Would you be willing to support the collection of data within your software development group beginning XX/XX/XXXX? *(if so, have them sign the organizational informed consent form)*

Next, I would like to get some information about you.

Tell me about your background and how you came to manage a software development group. (probe)

What are the greatest challenges / achievements you have encountered in managing this group?

Next, I would like to ask you some questions relating to the nature of scope change control and structured walkthrough practices used in your software development group.

Describe the scope change control process used within the software development group. What is expected of the software developer? (probe)

Describe the structured walkthrough process used within the software development group. What is expected of the software developer? (probe)

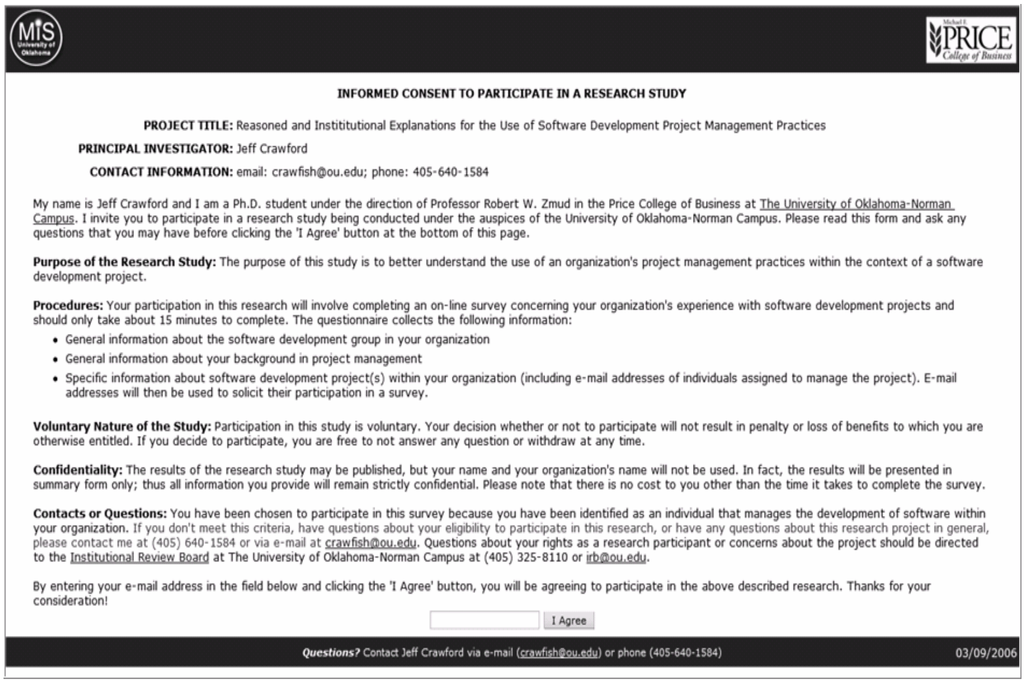
(if time permits) Finally, I would like to ask about you some questions about managing software development projects within your organization.

How do you tend to define software development project success? (probe)

What project management practices do you feel are key to enabling the successful delivery of software development projects in your organization?

Thank you very much for taking time to share your experiences and for your consideration regarding participation in this research project. Please feel free to e-mail me at crawfish@ou.edu or call me at 405-640-1584 if you have further questions or comments regarding today's interview.

Appendix VI – Software Development Manager Survey



INFORMED CONSENT TO PARTICIPATE IN A RESEARCH STUDY

PROJECT TITLE: Reasoned and Instititutional Explanations for the Use of Software Development Project Management Practices

PRINCIPAL INVESTIGATOR: Jeff Crawford

CONTACT INFORMATION: email: crawfish@ou.edu; phone: 405-640-1584

My name is Jeff Crawford and I am a Ph.D. student under the direction of Professor Robert W. Zmud in the Price College of Business at The University of Oklahoma-Norman Campus. I invite you to participate in a research study being conducted under the auspices of the University of Oklahoma-Norman Campus. Please read this form and ask any questions that you may have before clicking the 'I Agree' button at the bottom of this page.

Purpose of the Research Study: The purpose of this study is to better understand the use of an organization's project management practices within the context of a software development project.

Procedures: Your participation in this research will involve completing an on-line survey concerning your organization's experience with software development projects and should only take about 15 minutes to complete. The questionnaire collects the following information:

- General information about the software development group in your organization
- General information about your background in project management
- Specific information about software development project(s) within your organization (including e-mail addresses of individuals assigned to manage the project). E-mail addresses will then be used to solicit their participation in a survey.

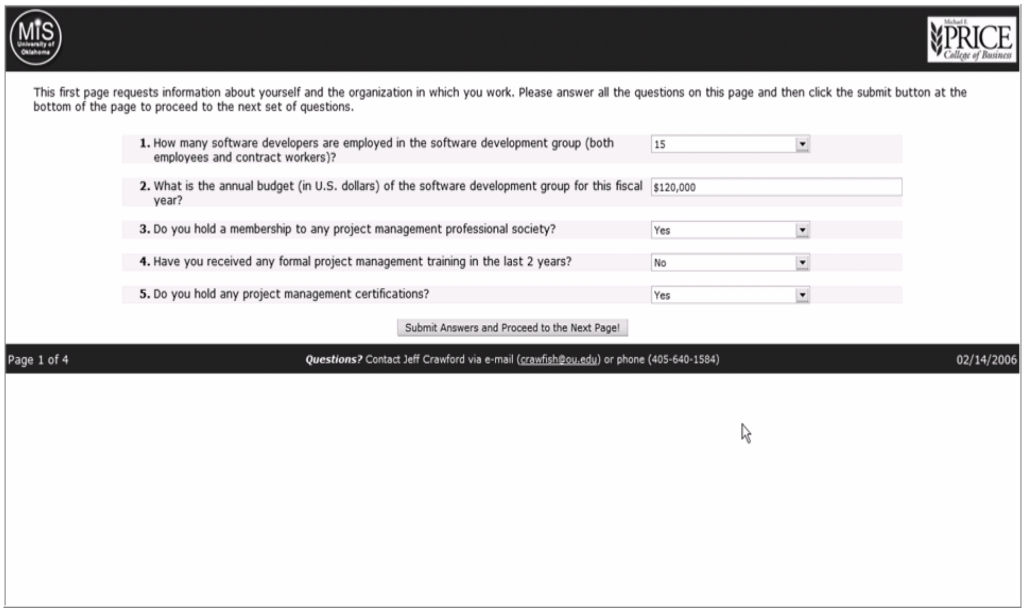
Voluntary Nature of the Study: Participation in this study is voluntary. Your decision whether or not to participate will not result in penalty or loss of benefits to which you are otherwise entitled. If you decide to participate, you are free to not answer any question or withdraw at any time.

Confidentiality: The results of the research study may be published, but your name and your organization's name will not be used. In fact, the results will be presented in summary form only; thus all information you provide will remain strictly confidential. Please note that there is no cost to you other than the time it takes to complete the survey.

Contacts or Questions: You have been chosen to participate in this survey because you have been identified as an individual that manages the development of software within your organization. If you don't meet this criteria, have questions about your eligibility to participate in this research, or have any questions about this research project in general, please contact me at (405) 640-1584 or via e-mail at crawfish@ou.edu. Questions about your rights as a research participant or concerns about the project should be directed to the Institutional Review Board at The University of Oklahoma-Norman Campus at (405) 325-8110 or irb@ou.edu.

By entering your e-mail address in the field below and clicking the 'I Agree' button, you will be agreeing to participate in the above described research. Thanks for your consideration!

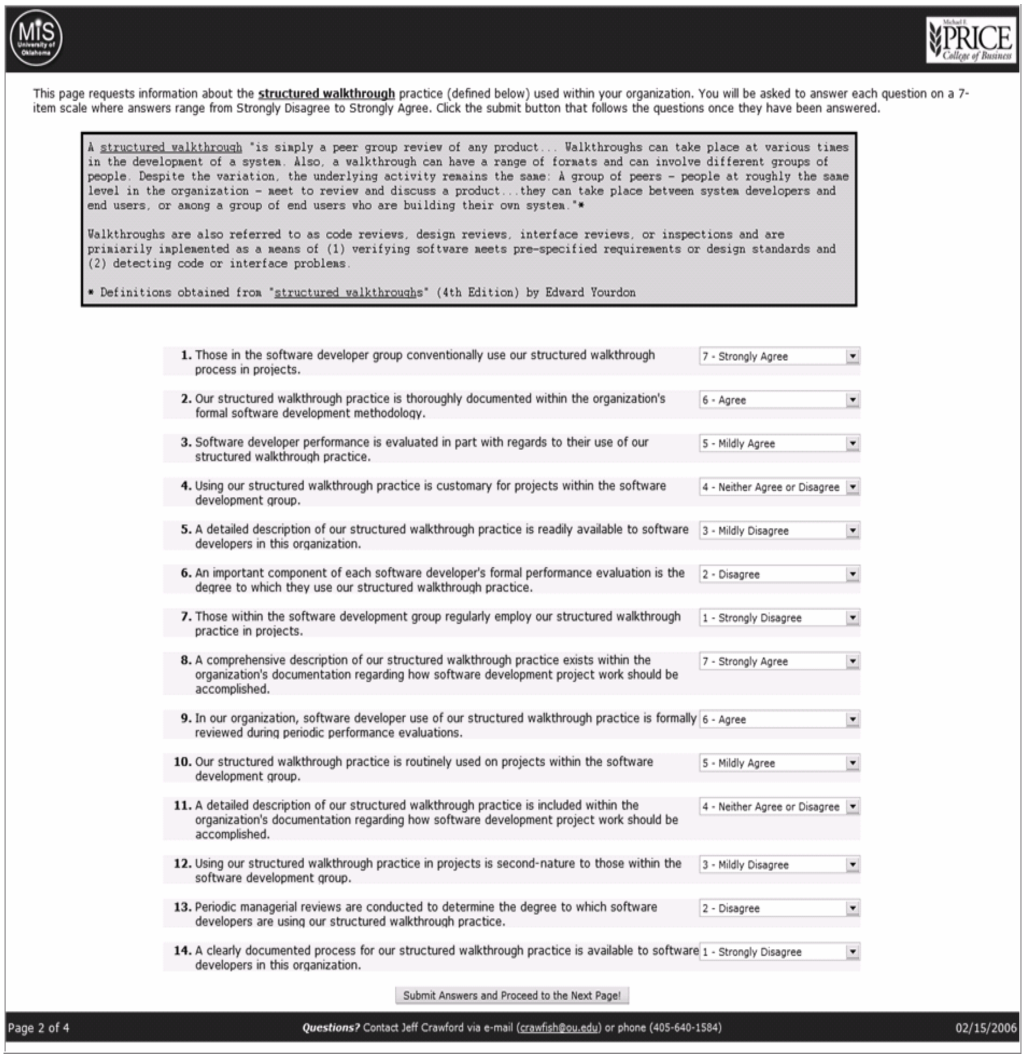
Questions? Contact Jeff Crawford via e-mail (crawfish@ou.edu) or phone (405-640-1584) 03/09/2006



This first page requests information about yourself and the organization in which you work. Please answer all the questions on this page and then click the submit button at the bottom of the page to proceed to the next set of questions.

1. How many software developers are employed in the software development group (both employees and contract workers)?	15
2. What is the annual budget (in U.S. dollars) of the software development group for this fiscal year?	$120,000
3. Do you hold a membership to any project management professional society?	Yes
4. Have you received any formal project management training in the last 2 years?	No
5. Do you hold any project management certifications?	Yes

Page 1 of 4 ***Questions?*** Contact Jeff Crawford via e-mail (crawfish@ou.edu) or phone (405-640-1584) 02/14/2006

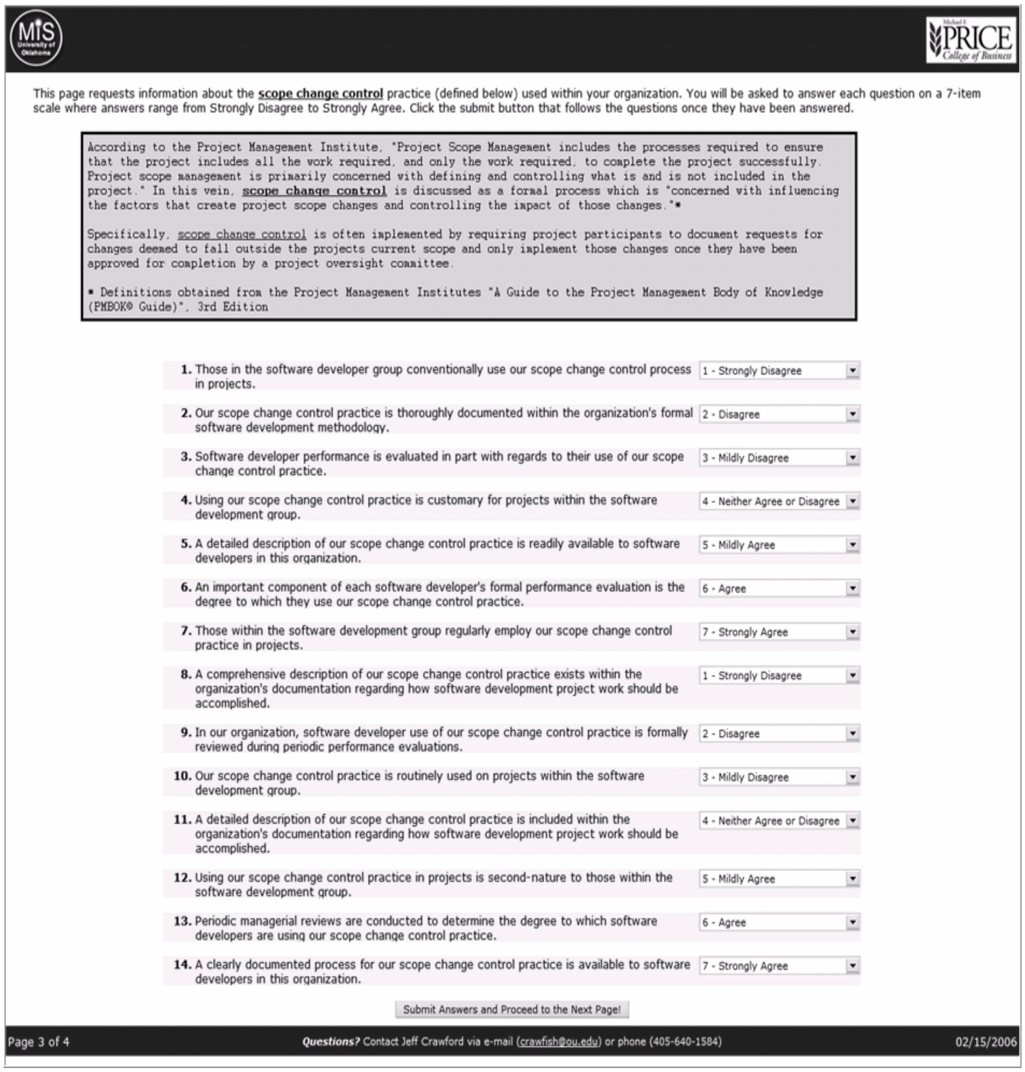

MIS University of Oklahoma
Michael F. PRICE College of Business
Submit Answers and Proceed to the Next Page!

This page requests information about the **structured walkthrough** practice (defined below) used within your organization. You will be asked to answer each question on a 7-item scale where answers range from Strongly Disagree to Strongly Agree. Click the submit button that follows the questions once they have been answered.

> A structured walkthrough "is simply a peer group review of any product... Walkthroughs can take place at various times in the development of a system. Also, a walkthrough can have a range of formats and can involve different groups of people. Despite the variation, the underlying activity remains the same: A group of peers – people at roughly the same level in the organization – meet to review and discuss a product...they can take place between system developers and end users, or among a group of end users who are building their own system."*
>
> Walkthroughs are also referred to as code reviews, design reviews, interface reviews, or inspections and are priniarily implemented as a means of (1) verifying software meets pre-specified requirements or design standards and (2) detecting code or interface problems.
>
> * Definitions obtained from "structured walkthroughs" (4th Edition) by Edward Yourdon

1.	Those in the software developer group conventionally use our structured walkthrough process in projects.	7 - Strongly Agree ▾
2.	Our structured walkthrough practice is thoroughly documented within the organization's formal software development methodology.	6 - Agree ▾
3.	Software developer performance is evaluated in part with regards to their use of our structured walkthrough practice.	5 - Mildly Agree ▾
4.	Using our structured walkthrough practice is customary for projects within the software development group.	4 - Neither Agree or Disagree ▾
5.	A detailed description of our structured walkthrough practice is readily available to software developers in this organization.	3 - Mildly Disagree ▾
6.	An important component of each software developer's formal performance evaluation is the degree to which they use our structured walkthrough practice.	2 - Disagree ▾
7.	Those within the software development group regularly employ our structured walkthrough practice in projects.	1 - Strongly Disagree ▾
8.	A comprehensive description of our structured walkthrough practice exists within the organization's documentation regarding how software development project work should be accomplished.	7 - Strongly Agree ▾
9.	In our organization, software developer use of our structured walkthrough practice is formally reviewed during periodic performance evaluations.	6 - Agree ▾
10.	Our structured walkthrough practice is routinely used on projects within the software development group.	5 - Mildly Agree ▾
11.	A detailed description of our structured walkthrough practice is included within the organization's documentation regarding how software development project work should be accomplished.	4 - Neither Agree or Disagree ▾
12.	Using our structured walkthrough practice in projects is second-nature to those within the software development group.	3 - Mildly Disagree ▾
13.	Periodic managerial reviews are conducted to determine the degree to which software developers are using our structured walkthrough practice.	2 - Disagree ▾
14.	A clearly documented process for our structured walkthrough practice is available to software developers in this organization.	1 - Strongly Disagree ▾

Page 2 of 4 **Questions?** Contact Jeff Crawford via e-mail (crawfish@ou.edu) or phone (405-640-1584) 02/15/2006



This page requests information about the **scope change control** practice (defined below) used within your organization. You will be asked to answer each question on a 7-item scale where answers range from Strongly Disagree to Strongly Agree. Click the submit button that follows the questions once they have been answered.

According to the Project Management Institute, "Project Scope Management includes the processes required to ensure that the project includes all the work required, and only the work required, to complete the project successfully. Project scope management is primarily concerned with defining and controlling what is and is not included in the project." In this vein, **scope change control** is discussed as a formal process which is "concerned with influencing the factors that create project scope changes and controlling the impact of those changes."*

Specifically, scope change control is often implemented by requiring project participants to document requests for changes deemed to fall outside the projects current scope and only implement those changes once they have been approved for completion by a project oversight committee.

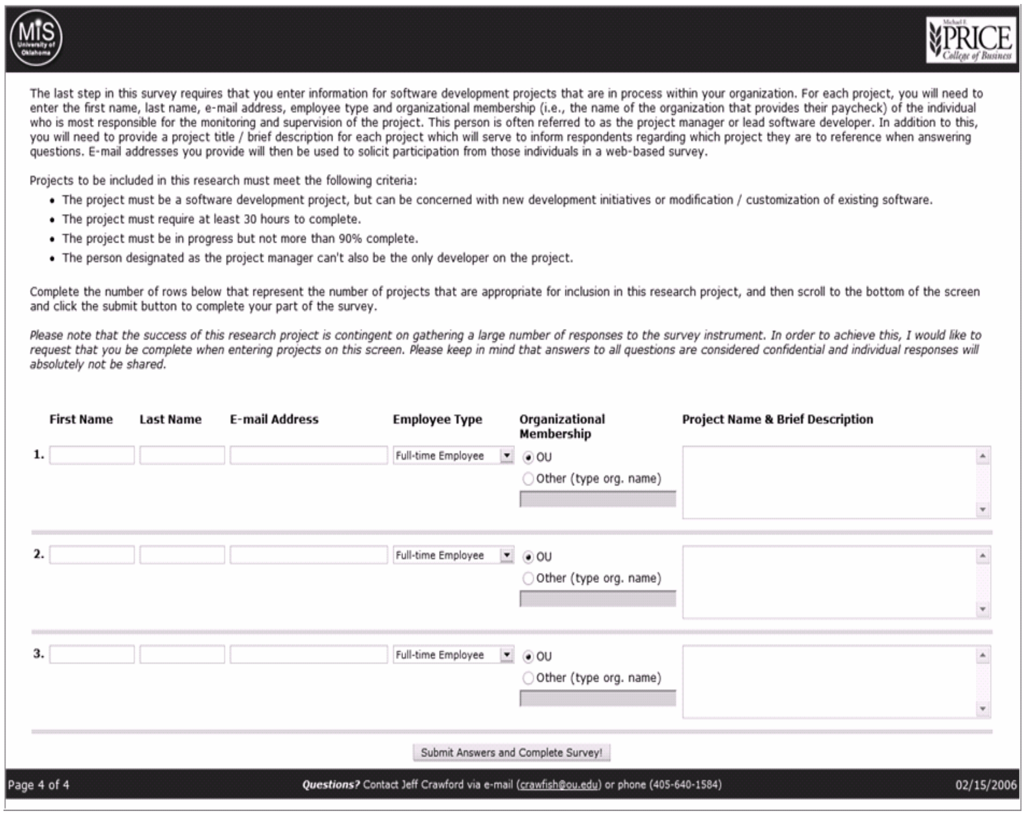
* Definitions obtained from the Project Management Institutes "A Guide to the Project Management Body of Knowledge (PMBOK® Guide)", 3rd Edition

1. Those in the software developer group conventionally use our scope change control process in projects.	1 - Strongly Disagree
2. Our scope change control practice is thoroughly documented within the organization's formal software development methodology.	2 - Disagree
3. Software developer performance is evaluated in part with regards to their use of our scope change control practice.	3 - Mildly Disagree
4. Using our scope change control practice is customary for projects within the software development group.	4 - Neither Agree or Disagree
5. A detailed description of our scope change control practice is readily available to software developers in this organization.	5 - Mildly Agree
6. An important component of each software developer's formal performance evaluation is the degree to which they use our scope change control practice.	6 - Agree
7. Those within the software development group regularly employ our scope change control practice in projects.	7 - Strongly Agree
8. A comprehensive description of our scope change control practice exists within the organization's documentation regarding how software development project work should be accomplished.	1 - Strongly Disagree
9. In our organization, software developer use of our scope change control practice is formally reviewed during periodic performance evaluations.	2 - Disagree
10. Our scope change control practice is routinely used on projects within the software development group.	3 - Mildly Disagree
11. A detailed description of our scope change control practice is included within the organization's documentation regarding how software development project work should be accomplished.	4 - Neither Agree or Disagree
12. Using our scope change control practice in projects is second-nature to those within the software development group.	5 - Mildly Agree
13. Periodic managerial reviews are conducted to determine the degree to which software developers are using our scope change control practice.	6 - Agree
14. A clearly documented process for our scope change control practice is available to software developers in this organization.	7 - Strongly Agree

Page 3 of 4

Questions? Contact Jeff Crawford via e-mail (crawfish@ou.edu) or phone (405-640-1584)

02/15/2006



The last step in this survey requires that you enter information for software development projects that are in process within your organization. For each project, you will need to enter the first name, last name, e-mail address, employee type and organizational membership (i.e., the name of the organization that provides their paycheck) of the individual who is most responsible for the monitoring and supervision of the project. This person is often referred to as the project manager or lead software developer. In addition to this, you will need to provide a project title / brief description for each project which will serve to inform respondents regarding which project they are to reference when answering questions. E-mail addresses you provide will then be used to solicit participation from those individuals in a web-based survey.

Projects to be included in this research must meet the following criteria:

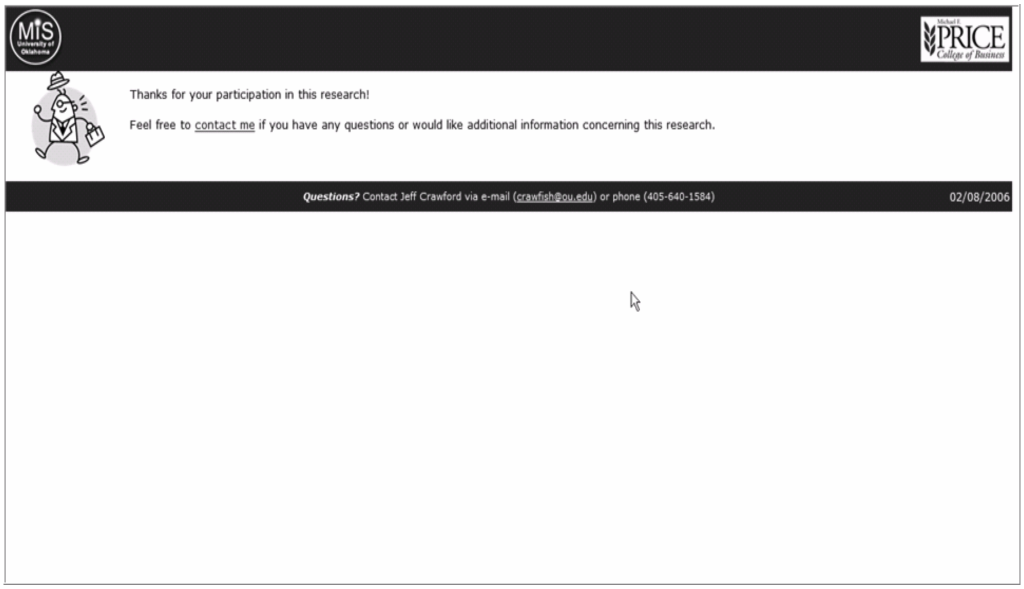
- The project must be a software development project, but can be concerned with new development initiatives or modification / customization of existing software.
- The project must require at least 30 hours to complete.
- The project must be in progress but not more than 90% complete.
- The person designated as the project manager can't also be the only developer on the project.

Complete the number of rows below that represent the number of projects that are appropriate for inclusion in this research project, and then scroll to the bottom of the screen and click the submit button to complete your part of the survey.

Please note that the success of this research project is contingent on gathering a large number of responses to the survey instrument. In order to achieve this, I would like to request that you be complete when entering projects on this screen. Please keep in mind that answers to all questions are considered confidential and individual responses will absolutely not be shared.

	First Name	Last Name	E-mail Address	Employee Type	Organizational Membership	Project Name & Brief Description
1.				Full-time Employee ▾	● OU ○ Other (type org. name)	
2.				Full-time Employee ▾	● OU ○ Other (type org. name)	
3.				Full-time Employee ▾	● OU ○ Other (type org. name)	

Page 4 of 4 **Questions?** Contact Jeff Crawford via e-mail (crawfish@ou.edu) or phone (405-640-1584) 02/15/2006

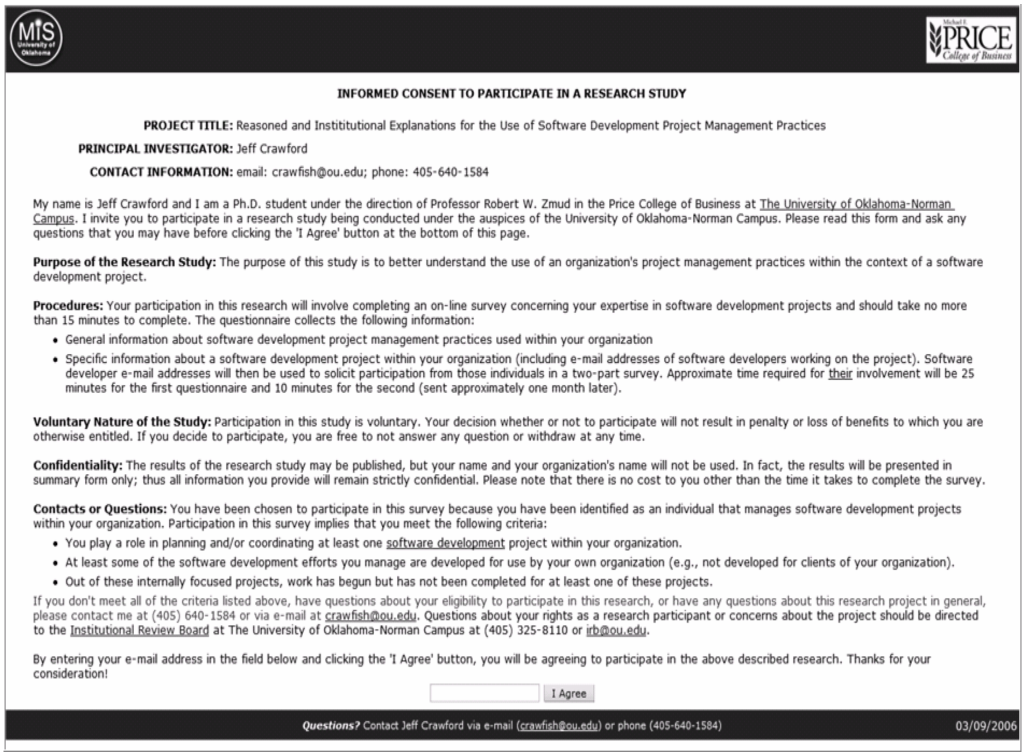


Thanks for your participation in this research!

Feel free to contact me if you have any questions or would like additional information concerning this research.

Questions? Contact Jeff Crawford via e-mail (crawfish@ou.edu) or phone (405-640-1584) 02/08/2006

Appendix VII – Project Manager Survey



INFORMED CONSENT TO PARTICIPATE IN A RESEARCH STUDY

PROJECT TITLE: Reasoned and Instititutional Explanations for the Use of Software Development Project Management Practices

PRINCIPAL INVESTIGATOR: Jeff Crawford

CONTACT INFORMATION: email: crawfish@ou.edu; phone: 405-640-1584

My name is Jeff Crawford and I am a Ph.D. student under the direction of Professor Robert W. Zmud in the Price College of Business at The University of Oklahoma-Norman Campus. I invite you to participate in a research study being conducted under the auspices of the University of Oklahoma-Norman Campus. Please read this form and ask any questions that you may have before clicking the 'I Agree' button at the bottom of this page.

Purpose of the Research Study: The purpose of this study is to better understand the use of an organization's project management practices within the context of a software development project.

Procedures: Your participation in this research will involve completing an on-line survey concerning your expertise in software development projects and should take no more than 15 minutes to complete. The questionnaire collects the following information:

- General information about software development project management practices used within your organization
- Specific information about a software development project within your organization (including e-mail addresses of software developers working on the project). Software developer e-mail addresses will then be used to solicit participation from those individuals in a two-part survey. Approximate time required for their involvement will be 25 minutes for the first questionnaire and 10 minutes for the second (sent approximately one month later).

Voluntary Nature of the Study: Participation in this study is voluntary. Your decision whether or not to participate will not result in penalty or loss of benefits to which you are otherwise entitled. If you decide to participate, you are free to not answer any question or withdraw at any time.

Confidentiality: The results of the research study may be published, but your name and your organization's name will not be used. In fact, the results will be presented in summary form only; thus all information you provide will remain strictly confidential. Please note that there is no cost to you other than the time it takes to complete the survey.

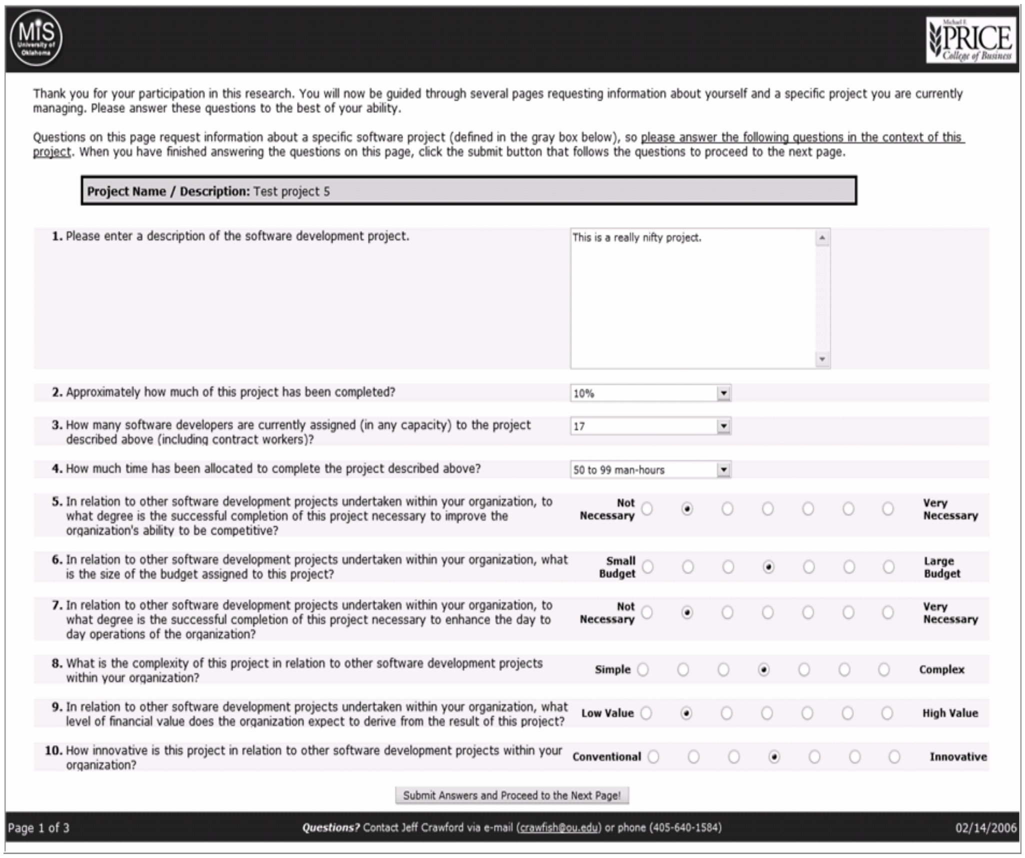
Contacts or Questions: You have been chosen to participate in this survey because you have been identified as an individual that manages software development projects within your organization. Participation in this survey implies that you meet the following criteria:

- You play a role in planning and/or coordinating at least one software development project within your organization.
- At least some of the software development efforts you manage are developed for use by your own organization (e.g., not developed for clients of your organization).
- Out of these internally focused projects, work has begun but has not been completed for at least one of these projects.

If you don't meet all of the criteria listed above, have questions about your eligibility to participate in this research, or have any questions about this research project in general, please contact me at (405) 640-1584 or via e-mail at crawfish@ou.edu. Questions about your rights as a research participant or concerns about the project should be directed to the Institutional Review Board at The University of Oklahoma-Norman Campus at (405) 325-8110 or irb@ou.edu.

By entering your e-mail address in the field below and clicking the 'I Agree' button, you will be agreeing to participate in the above described research. Thanks for your consideration!

Questions? Contact Jeff Crawford via e-mail (crawfish@ou.edu) or phone (405-640-1584) 03/09/2006



Thank you for your participation in this research. You will now be guided through several pages requesting information about yourself and a specific project you are currently managing. Please answer these questions to the best of your ability.

Questions on this page request information about a specific software project (defined in the gray box below), so please answer the following questions in the context of this project. When you have finished answering the questions on this page, click the submit button that follows the questions to proceed to the next page.

Project Name / Description: Test project 5

1. Please enter a description of the software development project. — This is a really nifty project.

2. Approximately how much of this project has been completed? — 10%

3. How many software developers are currently assigned (in any capacity) to the project described above (including contract workers)? — 17

4. How much time has been allocated to complete the project described above? — 50 to 99 man-hours

5. In relation to other software development projects undertaken within your organization, to what degree is the successful completion of this project necessary to improve the organization's ability to be competitive? — **Not Necessary** ○ ● ○ ○ ○ ○ ○ **Very Necessary**

6. In relation to other software development projects undertaken within your organization, what is the size of the budget assigned to this project? — **Small Budget** ○ ○ ○ ● ○ ○ ○ **Large Budget**

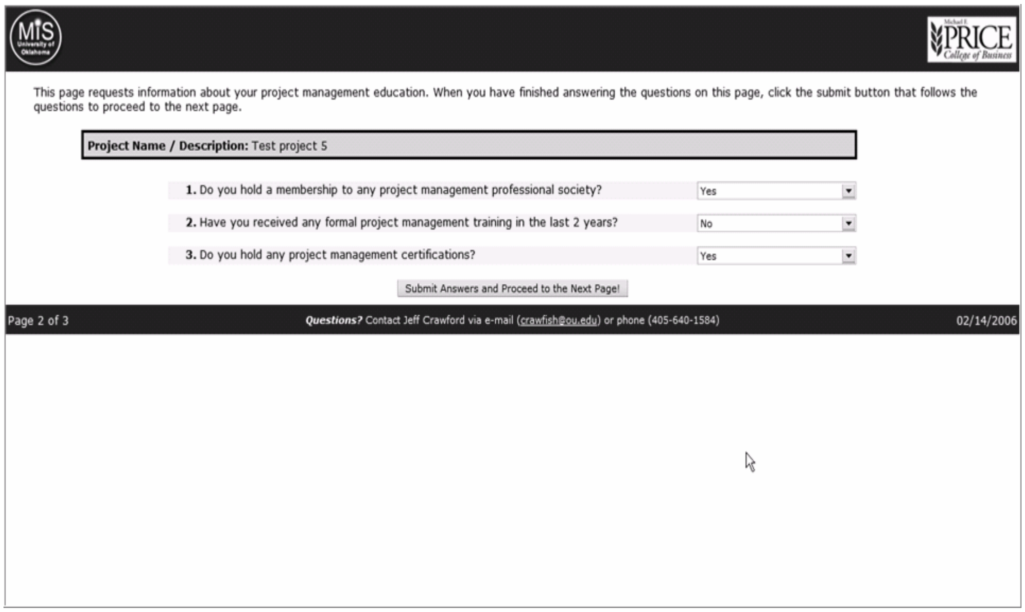
7. In relation to other software development projects undertaken within your organization, to what degree is the successful completion of this project necessary to enhance the day to day operations of the organization? — **Not Necessary** ○ ● ○ ○ ○ ○ ○ **Very Necessary**

8. What is the complexity of this project in relation to other software development projects within your organization? — **Simple** ○ ○ ○ ● ○ ○ ○ **Complex**

9. In relation to other software development projects undertaken within your organization, what level of financial value does the organization expect to derive from the result of this project? — **Low Value** ○ ● ○ ○ ○ ○ ○ **High Value**

10. How innovative is this project in relation to other software development projects within your organization? — **Conventional** ○ ○ ○ ● ○ ○ ○ **Innovative**

Page 1 of 3 — ***Questions?*** Contact Jeff Crawford via e-mail (crawfish@ou.edu) or phone (405-640-1584) — 02/14/2006



This page requests information about your project management education. When you have finished answering the questions on this page, click the submit button that follows the questions to proceed to the next page.

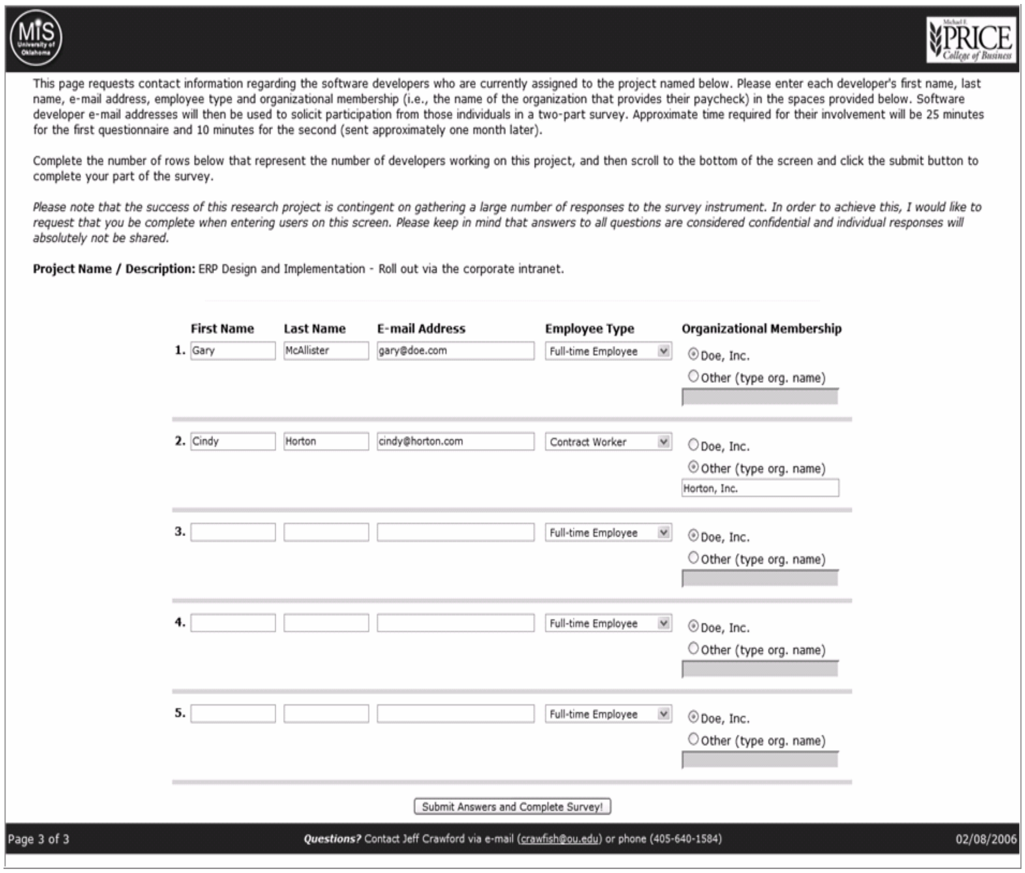
Project Name / Description: Test project 5

1. Do you hold a membership to any project management professional society? — Yes

2. Have you received any formal project management training in the last 2 years? — No

3. Do you hold any project management certifications? — Yes

Page 2 of 3 — ***Questions?*** Contact Jeff Crawford via e-mail (crawfish@ou.edu) or phone (405-640-1584) — 02/14/2006



This page requests contact information regarding the software developers who are currently assigned to the project named below. Please enter each developer's first name, last name, e-mail address, employee type and organizational membership (i.e., the name of the organization that provides their paycheck) in the spaces provided below. Software developer e-mail addresses will then be used to solicit participation from those individuals in a two-part survey. Approximate time required for their involvement will be 25 minutes for the first questionnaire and 10 minutes for the second (sent approximately one month later).

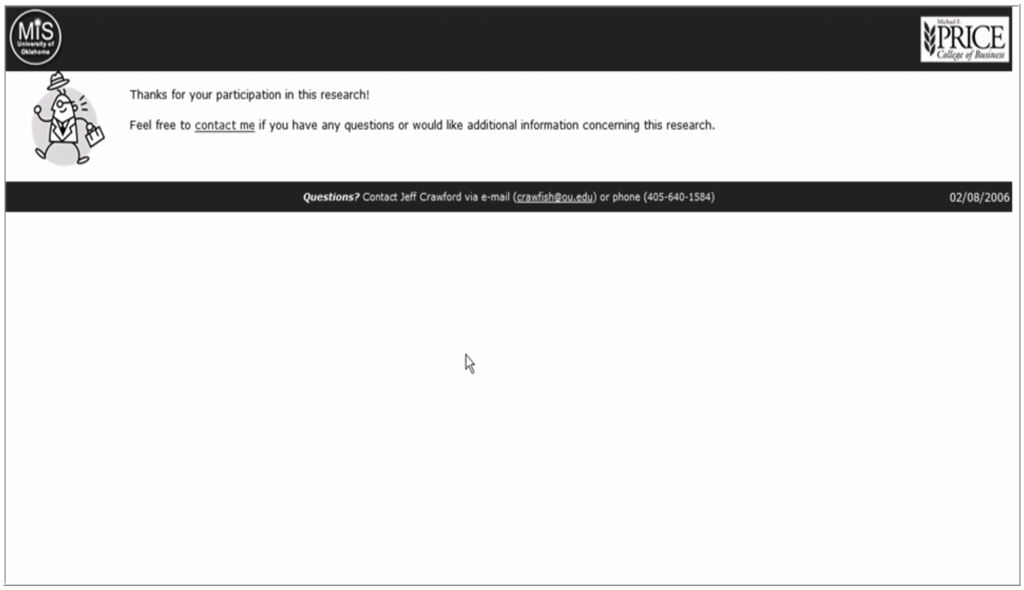
Complete the number of rows below that represent the number of developers working on this project, and then scroll to the bottom of the screen and click the submit button to complete your part of the survey.

Please note that the success of this research project is contingent on gathering a large number of responses to the survey instrument. In order to achieve this, I would like to request that you be complete when entering users on this screen. Please keep in mind that answers to all questions are considered confidential and individual responses will absolutely not be shared.

Project Name / Description: ERP Design and Implementation - Roll out via the corporate intranet.

	First Name	Last Name	E-mail Address	Employee Type	Organizational Membership
1.	Gary	McAllister	gary@doe.com	Full-time Employee	● Doe, Inc. ○ Other (type org. name)
2.	Cindy	Horton	cindy@horton.com	Contract Worker	○ Doe, Inc. ● Other (type org. name) Horton, Inc.
3.				Full-time Employee	● Doe, Inc. ○ Other (type org. name)
4.				Full-time Employee	● Doe, Inc. ○ Other (type org. name)
5.				Full-time Employee	● Doe, Inc. ○ Other (type org. name)

Page 3 of 3 ***Questions?*** Contact Jeff Crawford via e-mail (crawfish@ou.edu) or phone (405-640-1584) 02/08/2006

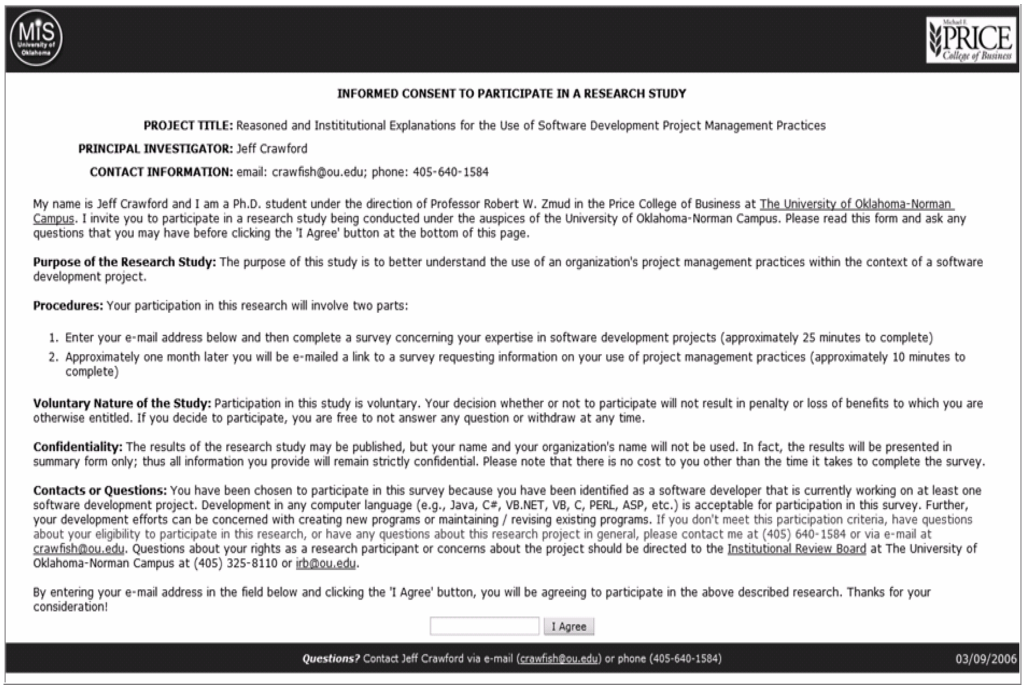


Thanks for your participation in this research!

Feel free to contact me if you have any questions or would like additional information concerning this research.

Questions? Contact Jeff Crawford via e-mail (crawfish@ou.edu) or phone (405-640-1584) 02/08/2006

Appendix VIII – Software Developer Survey (Time 1)



INFORMED CONSENT TO PARTICIPATE IN A RESEARCH STUDY

PROJECT TITLE: Reasoned and Instititutional Explanations for the Use of Software Development Project Management Practices

PRINCIPAL INVESTIGATOR: Jeff Crawford

CONTACT INFORMATION: email: crawfish@ou.edu; phone: 405-640-1584

My name is Jeff Crawford and I am a Ph.D. student under the direction of Professor Robert W. Zmud in the Price College of Business at The University of Oklahoma-Norman Campus. I invite you to participate in a research study being conducted under the auspices of the University of Oklahoma-Norman Campus. Please read this form and ask any questions that you may have before clicking the 'I Agree' button at the bottom of this page.

Purpose of the Research Study: The purpose of this study is to better understand the use of an organization's project management practices within the context of a software development project.

Procedures: Your participation in this research will involve two parts:

1. Enter your e-mail address below and then complete a survey concerning your expertise in software development projects (approximately 25 minutes to complete)
2. Approximately one month later you will be e-mailed a link to a survey requesting information on your use of project management practices (approximately 10 minutes to complete)

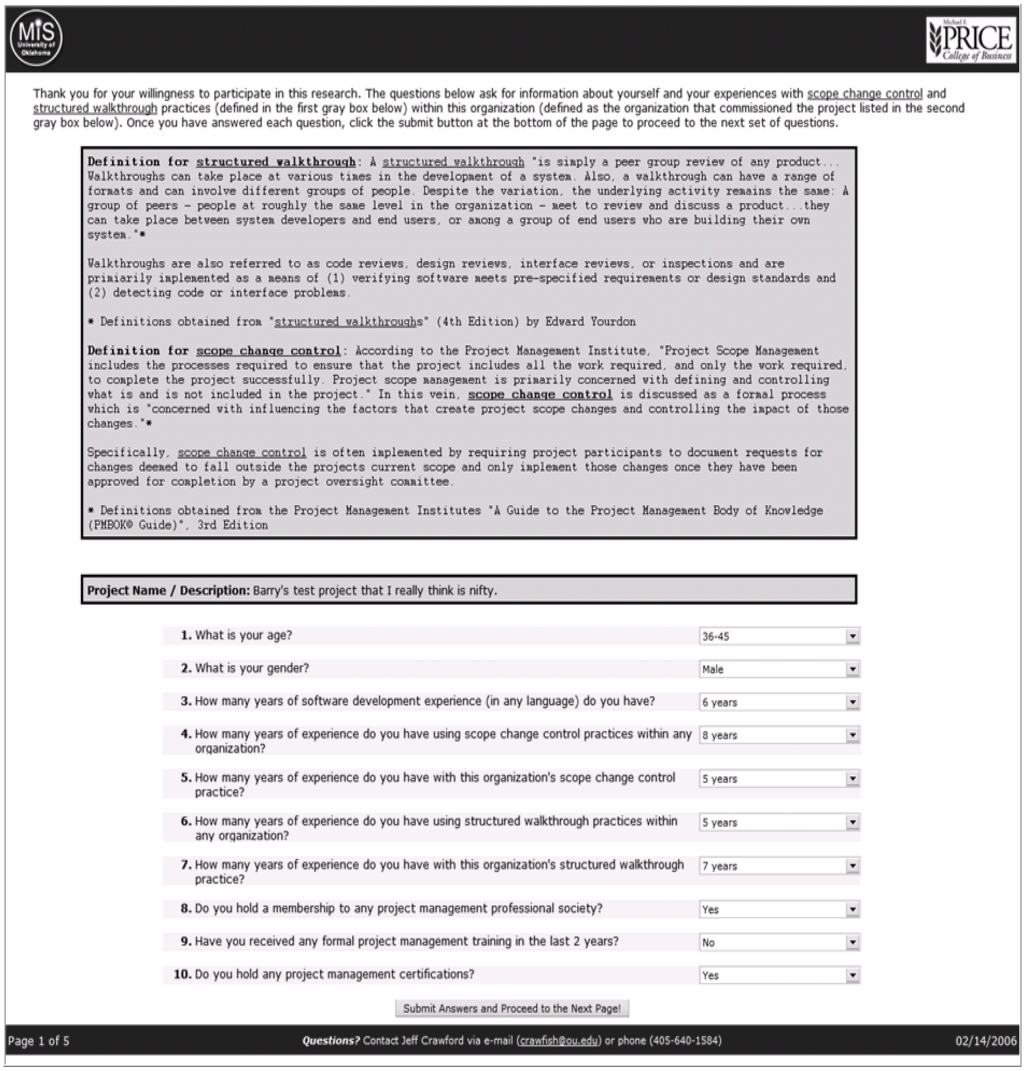
Voluntary Nature of the Study: Participation in this study is voluntary. Your decision whether or not to participate will not result in penalty or loss of benefits to which you are otherwise entitled. If you decide to participate, you are free to not answer any question or withdraw at any time.

Confidentiality: The results of the research study may be published, but your name and your organization's name will not be used. In fact, the results will be presented in summary form only; thus all information you provide will remain strictly confidential. Please note that there is no cost to you other than the time it takes to complete the survey.

Contacts or Questions: You have been chosen to participate in this survey because you have been identified as a software developer that is currently working on at least one software development project. Development in any computer language (e.g., Java, C#, VB.NET, VB, C, PERL, ASP, etc.) is acceptable for participation in this survey. Further, your development efforts can be concerned with creating new programs or maintaining / revising existing programs. If you don't meet this participation criteria, have questions about your eligibility to participate in this research, or have any questions about this research project in general, please contact me at (405) 640-1584 or via e-mail at crawfish@ou.edu. Questions about your rights as a research participant or concerns about the project should be directed to the Institutional Review Board at The University of Oklahoma-Norman Campus at (405) 325-8110 or irb@ou.edu.

By entering your e-mail address in the field below and clicking the 'I Agree' button, you will be agreeing to participate in the above described research. Thanks for your consideration!

Questions? Contact Jeff Crawford via e-mail (crawfish@ou.edu) or phone (405-640-1584) 03/09/2006



Thank you for your willingness to participate in this research. The questions below ask for information about yourself and your experiences with scope change control and structured walkthrough practices (defined in the first gray box below) within this organization (defined as the organization that commissioned the project listed in the second gray box below). Once you have answered each question, click the submit button at the bottom of the page to proceed to the next set of questions.

Definition for structured walkthrough: A structured walkthrough "is simply a peer group review of any product... Walkthroughs can take place at various times in the development of a system. Also, a walkthrough can have a range of formats and can involve different groups of people. Despite the variation, the underlying activity remains the same: A group of peers - people at roughly the same level in the organization - meet to review and discuss a product...they can take place between system developers and end users, or among a group of end users who are building their own system."*

Walkthroughs are also referred to as code reviews, design reviews, interface reviews, or inspections and are priniarily implemented as a means of (1) verifying software meets pre-specified requirements or design standards and (2) detecting code or interface problems.

* Definitions obtained from "structured walkthroughs" (4th Edition) by Edward Yourdon

Definition for scope change control: According to the Project Management Institute, "Project Scope Management includes the processes required to ensure that the project includes all the work required, and only the work required, to complete the project successfully. Project scope management is primarily concerned with defining and controlling what is and is not included in the project." In this vein, **scope change control** is discussed as a formal process which is "concerned with influencing the factors that create project scope changes and controlling the impact of those changes."*

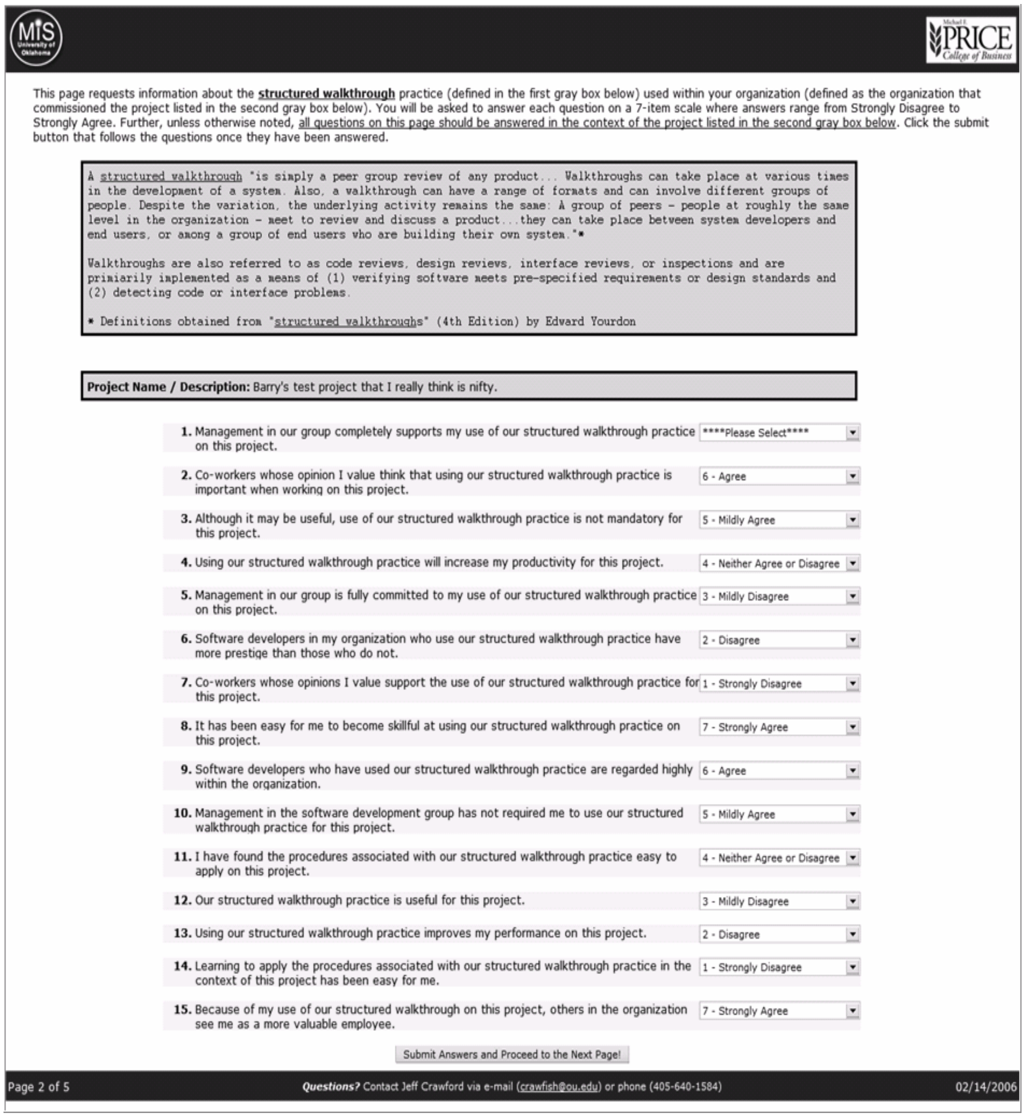
Specifically, scope change control is often implemented by requiring project participants to document requests for changes deemed to fall outside the projects current scope and only implement those changes once they have been approved for completion by a project oversight committee.

* Definitions obtained from the Project Management Institutes "A Guide to the Project Management Body of Knowledge (PMBOK® Guide)", 3rd Edition

Project Name / Description: Barry's test project that I really think is nifty.

1. What is your age?	36-45 ▾
2. What is your gender?	Male ▾
3. How many years of software development experience (in any language) do you have?	6 years ▾
4. How many years of experience do you have using scope change control practices within any organization?	8 years ▾
5. How many years of experience do you have with this organization's scope change control practice?	5 years ▾
6. How many years of experience do you have using structured walkthrough practices within any organization?	5 years ▾
7. How many years of experience do you have with this organization's structured walkthrough practice?	7 years ▾
8. Do you hold a membership to any project management professional society?	Yes ▾
9. Have you received any formal project management training in the last 2 years?	No ▾
10. Do you hold any project management certifications?	Yes ▾

Page 1 of 5 *Questions?* Contact Jeff Crawford via e-mail (crawfish@ou.edu) or phone (405-640-1584) 02/14/2006



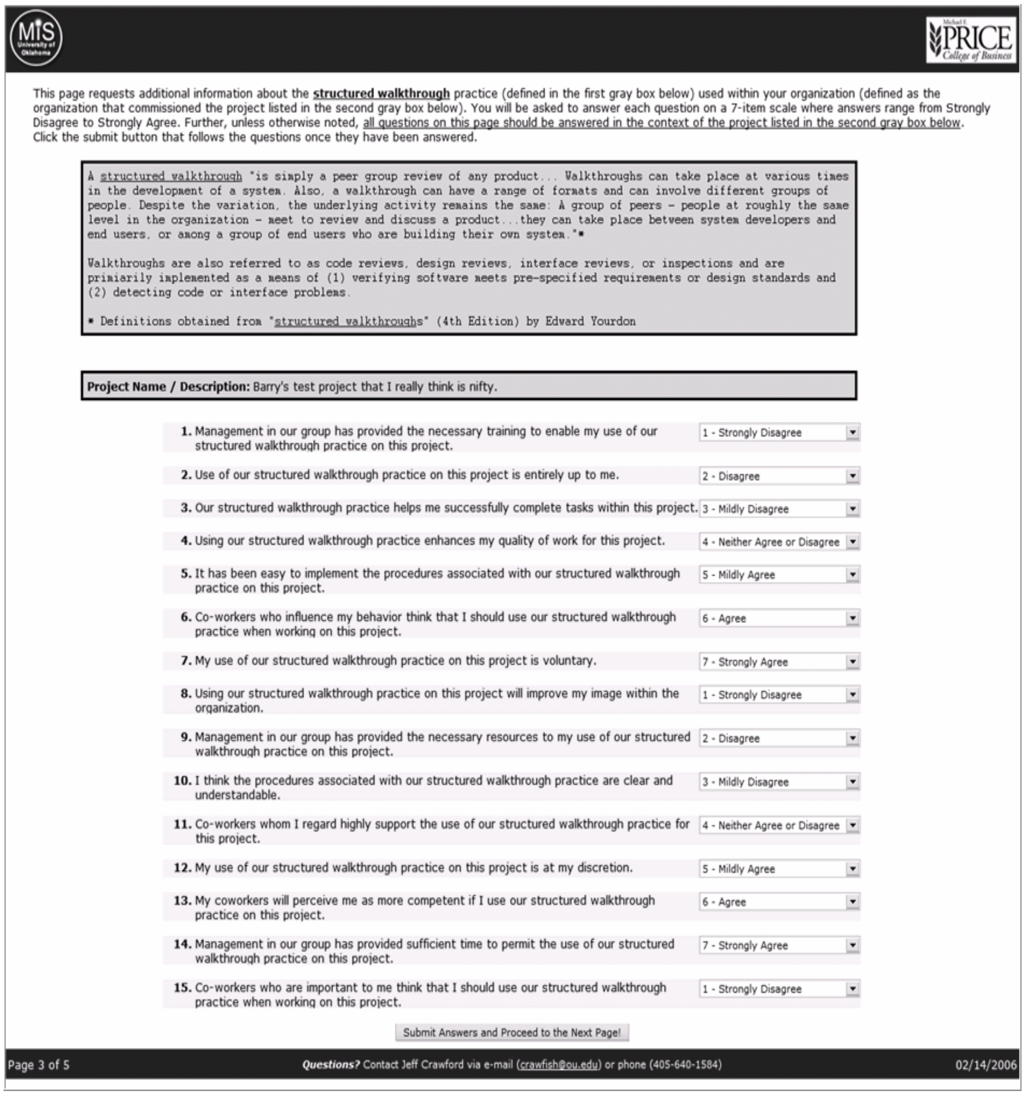
This page requests information about the **structured walkthrough** practice (defined in the first gray box below) used within your organization (defined as the organization that commissioned the project listed in the second gray box below). You will be asked to answer each question on a 7-item scale where answers range from Strongly Disagree to Strongly Agree. Further, unless otherwise noted, all questions on this page should be answered in the context of the project listed in the second gray box below. Click the submit button that follows the questions once they have been answered.

> A structured walkthrough "is simply a peer group review of any product... Walkthroughs can take place at various times in the development of a system. Also, a walkthrough can have a range of formats and can involve different groups of people. Despite the variation, the underlying activity remains the same: A group of peers – people at roughly the same level in the organization – meet to review and discuss a product...they can take place between system developers and end users, or among a group of end users who are building their own system."*
>
> Walkthroughs are also referred to as code reviews, design reviews, interface reviews, or inspections and are priniarily implemented as a means of (1) verifying software meets pre-specified requirements or design standards and (2) detecting code or interface problems.
>
> * Definitions obtained from "structured walkthroughs" (4th Edition) by Edward Yourdon

Project Name / Description: Barry's test project that I really think is nifty.

1. Management in our group completely supports my use of our structured walkthrough practice on this project.	****Please Select**** ▾
2. Co-workers whose opinion I value think that using our structured walkthrough practice is important when working on this project.	6 - Agree ▾
3. Although it may be useful, use of our structured walkthrough practice is not mandatory for this project.	5 - Mildly Agree ▾
4. Using our structured walkthrough practice will increase my productivity for this project.	4 - Neither Agree or Disagree ▾
5. Management in our group is fully committed to my use of our structured walkthrough practice on this project.	3 - Mildly Disagree ▾
6. Software developers in my organization who use our structured walkthrough practice have more prestige than those who do not.	2 - Disagree ▾
7. Co-workers whose opinions I value support the use of our structured walkthrough practice for this project.	1 - Strongly Disagree ▾
8. It has been easy for me to become skillful at using our structured walkthrough practice on this project.	7 - Strongly Agree ▾
9. Software developers who have used our structured walkthrough practice are regarded highly within the organization.	6 - Agree ▾
10. Management in the software development group has not required me to use our structured walkthrough practice for this project.	5 - Mildly Agree ▾
11. I have found the procedures associated with our structured walkthrough practice easy to apply on this project.	4 - Neither Agree or Disagree ▾
12. Our structured walkthrough practice is useful for this project.	3 - Mildly Disagree ▾
13. Using our structured walkthrough practice improves my performance on this project.	2 - Disagree ▾
14. Learning to apply the procedures associated with our structured walkthrough practice in the context of this project has been easy for me.	1 - Strongly Disagree ▾
15. Because of my use of our structured walkthrough on this project, others in the organization see me as a more valuable employee.	7 - Strongly Agree ▾

Page 2 of 5 ***Questions?*** Contact Jeff Crawford via e-mail (crawfish@ou.edu) or phone (405-640-1584) 02/14/2006



MIS University of Oklahoma
Michael F. PRICE College of Business

This page requests additional information about the **structured walkthrough** practice (defined in the first gray box below) used within your organization (defined as the organization that commissioned the project listed in the second gray box below). You will be asked to answer each question on a 7-item scale where answers range from Strongly Disagree to Strongly Agree. Further, unless otherwise noted, all questions on this page should be answered in the context of the project listed in the second gray box below. Click the submit button that follows the questions once they have been answered.

> A structured walkthrough "is simply a peer group review of any product... Walkthroughs can take place at various times in the development of a system. Also, a walkthrough can have a range of formats and can involve different groups of people. Despite the variation, the underlying activity remains the same: A group of peers – people at roughly the same level in the organization – meet to review and discuss a product...they can take place between system developers and end users, or among a group of end users who are building their own system."*
>
> Walkthroughs are also referred to as code reviews, design reviews, interface reviews, or inspections and are priniarily implemented as a means of (1) verifying software meets pre-specified requirements or design standards and (2) detecting code or interface problems.
>
> * Definitions obtained from "structured walkthroughs" (4th Edition) by Edward Yourdon

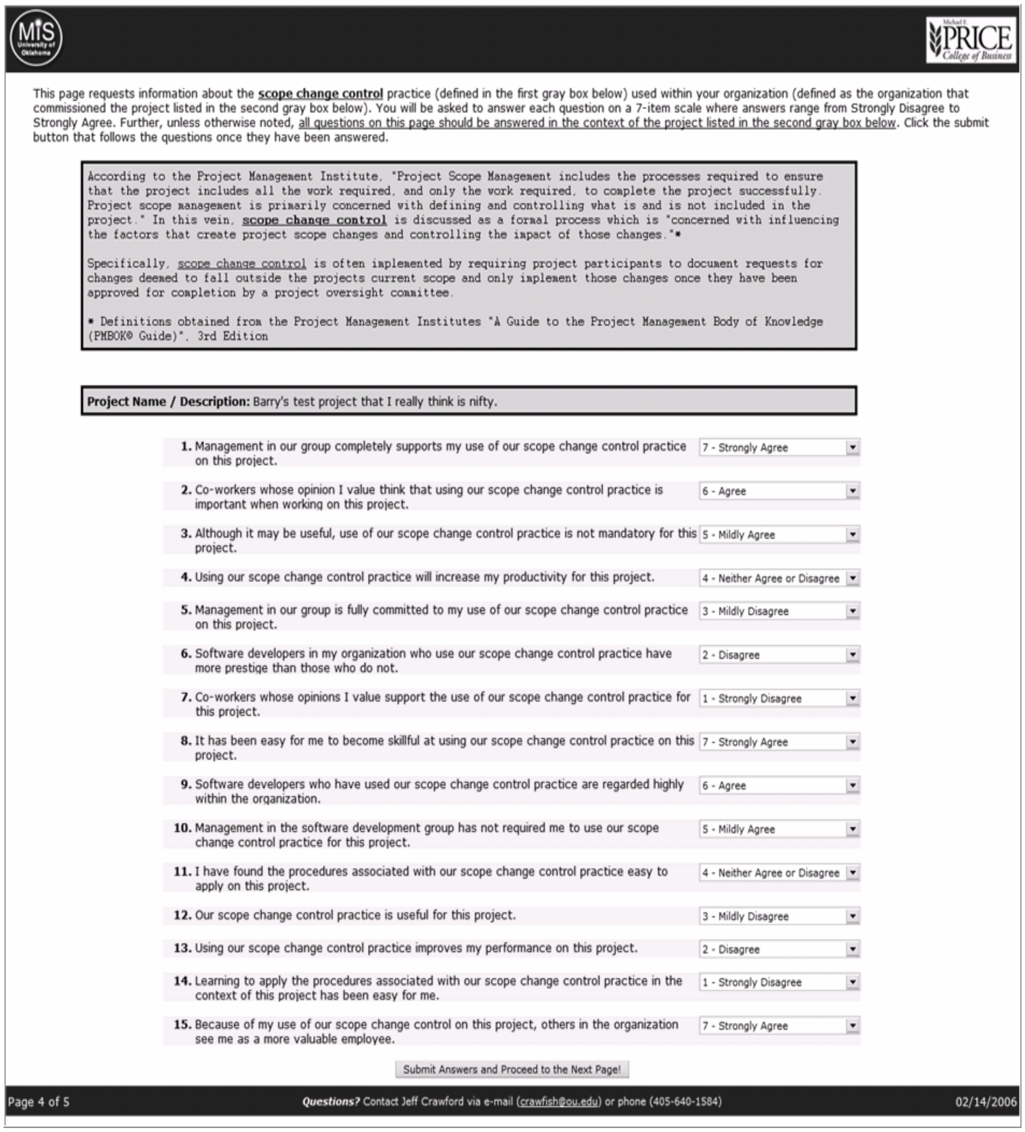
Project Name / Description: Barry's test project that I really think is nifty.

1. Management in our group has provided the necessary training to enable my use of our structured walkthrough practice on this project.	1 - Strongly Disagree
2. Use of our structured walkthrough practice on this project is entirely up to me.	2 - Disagree
3. Our structured walkthrough practice helps me successfully complete tasks within this project.	3 - Mildly Disagree
4. Using our structured walkthrough practice enhances my quality of work for this project.	4 - Neither Agree or Disagree
5. It has been easy to implement the procedures associated with our structured walkthrough practice on this project.	5 - Mildly Agree
6. Co-workers who influence my behavior think that I should use our structured walkthrough practice when working on this project.	6 - Agree
7. My use of our structured walkthrough practice on this project is voluntary.	7 - Strongly Agree
8. Using our structured walkthrough practice on this project will improve my image within the organization.	1 - Strongly Disagree
9. Management in our group has provided the necessary resources to my use of our structured walkthrough practice on this project.	2 - Disagree
10. I think the procedures associated with our structured walkthrough practice are clear and understandable.	3 - Mildly Disagree
11. Co-workers whom I regard highly support the use of our structured walkthrough practice for this project.	4 - Neither Agree or Disagree
12. My use of our structured walkthrough practice on this project is at my discretion.	5 - Mildly Agree
13. My coworkers will perceive me as more competent if I use our structured walkthrough practice on this project.	6 - Agree
14. Management in our group has provided sufficient time to permit the use of our structured walkthrough practice on this project.	7 - Strongly Agree
15. Co-workers who are important to me think that I should use our structured walkthrough practice when working on this project.	1 - Strongly Disagree

Page 3 of 5

Questions? Contact Jeff Crawford via e-mail (crawfish@ou.edu) or phone (405-640-1584)

02/14/2006


MIS
University of Oklahoma
Michael F. PRICE College of Business
Submit Answers and Proceed to the Next Page!

This page requests information about the **scope change control** practice (defined in the first gray box below) used within your organization (defined as the organization that commissioned the project listed in the second gray box below). You will be asked to answer each question on a 7-item scale where answers range from Strongly Disagree to Strongly Agree. Further, unless otherwise noted, all questions on this page should be answered in the context of the project listed in the second gray box below. Click the submit button that follows the questions once they have been answered.

According to the Project Management Institute, "Project Scope Management includes the processes required to ensure that the project includes all the work required, and only the work required, to complete the project successfully. Project scope management is primarily concerned with defining and controlling what is and is not included in the project." In this vein, **scope change control** is discussed as a formal process which is "concerned with influencing the factors that create project scope changes and controlling the impact of those changes."*

Specifically, scope change control is often implemented by requiring project participants to document requests for changes deemed to fall outside the projects current scope and only implement those changes once they have been approved for completion by a project oversight committee.

* Definitions obtained from the Project Management Institutes "A Guide to the Project Management Body of Knowledge (PMBOK® Guide)", 3rd Edition

Project Name / Description: Barry's test project that I really think is nifty.

1. Management in our group completely supports my use of our scope change control practice on this project. 7 - Strongly Agree

2. Co-workers whose opinion I value think that using our scope change control practice is important when working on this project. 6 - Agree

3. Although it may be useful, use of our scope change control practice is not mandatory for this project. 5 - Mildly Agree

4. Using our scope change control practice will increase my productivity for this project. 4 - Neither Agree or Disagree

5. Management in our group is fully committed to my use of our scope change control practice on this project. 3 - Mildly Disagree

6. Software developers in my organization who use our scope change control practice have more prestige than those who do not. 2 - Disagree

7. Co-workers whose opinions I value support the use of our scope change control practice for this project. 1 - Strongly Disagree

8. It has been easy for me to become skillful at using our scope change control practice on this project. 7 - Strongly Agree

9. Software developers who have used our scope change control practice are regarded highly within the organization. 6 - Agree

10. Management in the software development group has not required me to use our scope change control practice for this project. 5 - Mildly Agree

11. I have found the procedures associated with our scope change control practice easy to apply on this project. 4 - Neither Agree or Disagree

12. Our scope change control practice is useful for this project. 3 - Mildly Disagree

13. Using our scope change control practice improves my performance on this project. 2 - Disagree

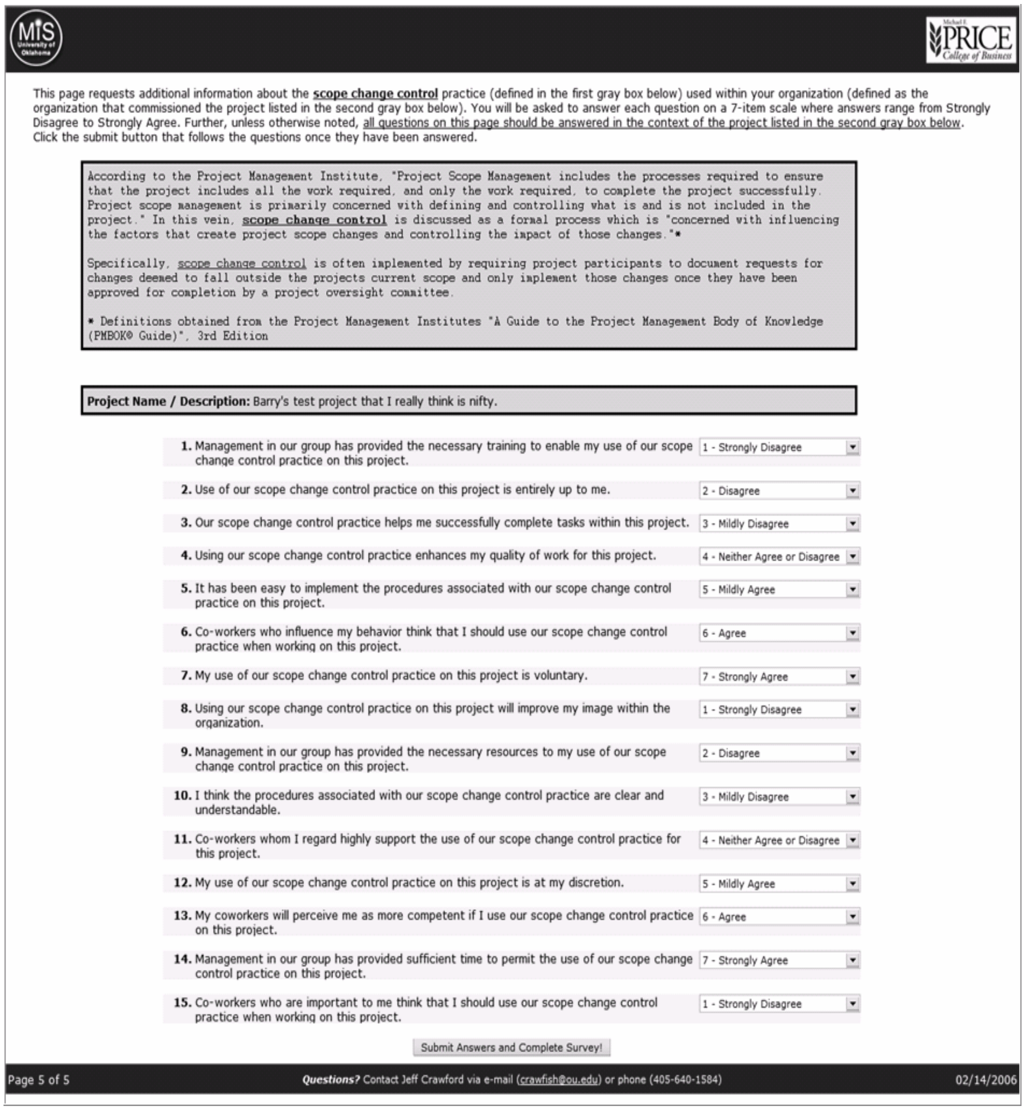
14. Learning to apply the procedures associated with our scope change control practice in the context of this project has been easy for me. 1 - Strongly Disagree

15. Because of my use of our scope change control on this project, others in the organization see me as a more valuable employee. 7 - Strongly Agree

Page 4 of 5

Questions? Contact Jeff Crawford via e-mail (crawfish@ou.edu) or phone (405-640-1584)

02/14/2006


MIS University of Oklahoma
Michael F. PRICE College of Business

This page requests additional information about the **scope change control** practice (defined in the first gray box below) used within your organization (defined as the organization that commissioned the project listed in the second gray box below). You will be asked to answer each question on a 7-item scale where answers range from Strongly Disagree to Strongly Agree. Further, unless otherwise noted, all questions on this page should be answered in the context of the project listed in the second gray box below. Click the submit button that follows the questions once they have been answered.

> According to the Project Management Institute, "Project Scope Management includes the processes required to ensure that the project includes all the work required, and only the work required, to complete the project successfully. Project scope management is primarily concerned with defining and controlling what is and is not included in the project." In this vein, **scope change control** is discussed as a formal process which is "concerned with influencing the factors that create project scope changes and controlling the impact of those changes."*
>
> Specifically, scope change control is often implemented by requiring project participants to document requests for changes deemed to fall outside the projects current scope and only implement those changes once they have been approved for completion by a project oversight committee.
>
> * Definitions obtained from the Project Management Institutes "A Guide to the Project Management Body of Knowledge (PMBOK® Guide)", 3rd Edition

Project Name / Description: Barry's test project that I really think is nifty.

1. Management in our group has provided the necessary training to enable my use of our scope change control practice on this project. 1 - Strongly Disagree

2. Use of our scope change control practice on this project is entirely up to me. 2 - Disagree

3. Our scope change control practice helps me successfully complete tasks within this project. 3 - Mildly Disagree

4. Using our scope change control practice enhances my quality of work for this project. 4 - Neither Agree or Disagree

5. It has been easy to implement the procedures associated with our scope change control practice on this project. 5 - Mildly Agree

6. Co-workers who influence my behavior think that I should use our scope change control practice when working on this project. 6 - Agree

7. My use of our scope change control practice on this project is voluntary. 7 - Strongly Agree

8. Using our scope change control practice on this project will improve my image within the organization. 1 - Strongly Disagree

9. Management in our group has provided the necessary resources to my use of our scope change control practice on this project. 2 - Disagree

10. I think the procedures associated with our scope change control practice are clear and understandable. 3 - Mildly Disagree

11. Co-workers whom I regard highly support the use of our scope change control practice for this project. 4 - Neither Agree or Disagree

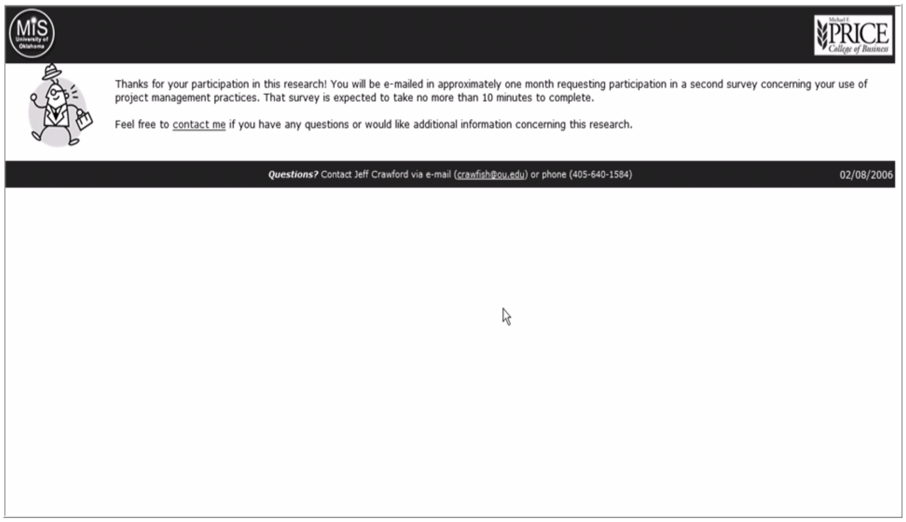
12. My use of our scope change control practice on this project is at my discretion. 5 - Mildly Agree

13. My coworkers will perceive me as more competent if I use our scope change control practice on this project. 6 - Agree

14. Management in our group has provided sufficient time to permit the use of our scope change control practice on this project. 7 - Strongly Agree

15. Co-workers who are important to me think that I should use our scope change control practice when working on this project. 1 - Strongly Disagree

Page 5 of 5 ***Questions?*** Contact Jeff Crawford via e-mail (crawfish@ou.edu) or phone (405-640-1584) 02/14/2006

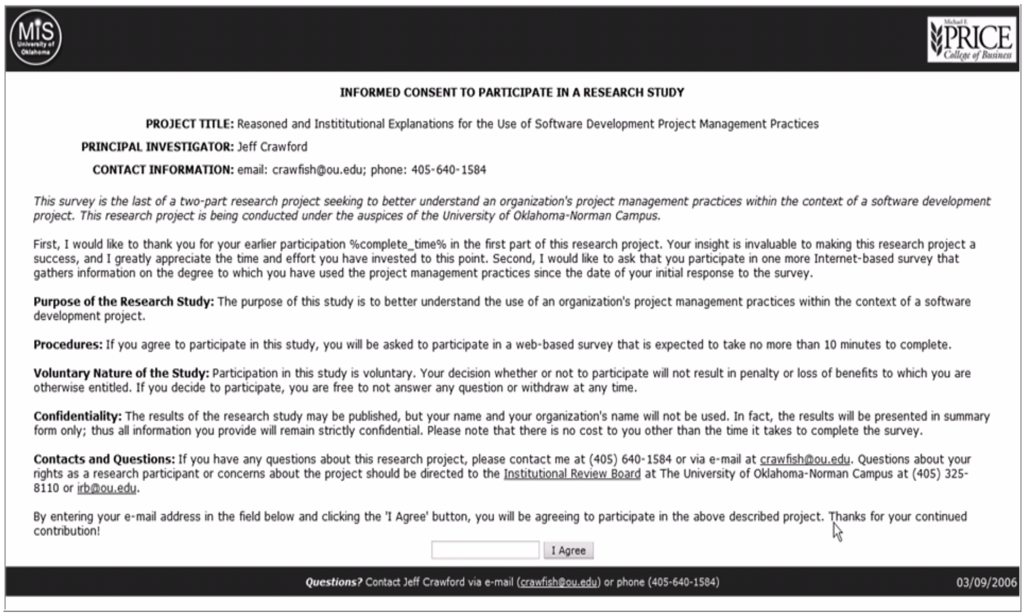

MIS University of Oklahoma
Michael F. PRICE College of Business

Thanks for your participation in this research! You will be e-mailed in approximately one month requesting participation in a second survey concerning your use of project management practices. That survey is expected to take no more than 10 minutes to complete.

Feel free to contact me if you have any questions or would like additional information concerning this research.

Questions? Contact Jeff Crawford via e-mail (crawfish@ou.edu) or phone (405-640-1584) 02/08/2006

Appendix IX – Software Developer Survey (Time 2)



MIS University of Oklahoma
Michael F. PRICE College of Business
I Agree

INFORMED CONSENT TO PARTICIPATE IN A RESEARCH STUDY

PROJECT TITLE: Reasoned and Instititutional Explanations for the Use of Software Development Project Management Practices

PRINCIPAL INVESTIGATOR: Jeff Crawford

CONTACT INFORMATION: email: crawfish@ou.edu; phone: 405-640-1584

This survey is the last of a two-part research project seeking to better understand an organization's project management practices within the context of a software development project. This research project is being conducted under the auspices of the University of Oklahoma-Norman Campus.

First, I would like to thank you for your earlier participation %complete_time% in the first part of this research project. Your insight is invaluable to making this research project a success, and I greatly appreciate the time and effort you have invested to this point. Second, I would like to ask that you participate in one more Internet-based survey that gathers information on the degree to which you have used the project management practices since the date of your initial response to the survey.

Purpose of the Research Study: The purpose of this study is to better understand the use of an organization's project management practices within the context of a software development project.

Procedures: If you agree to participate in this study, you will be asked to participate in a web-based survey that is expected to take no more than 10 minutes to complete.

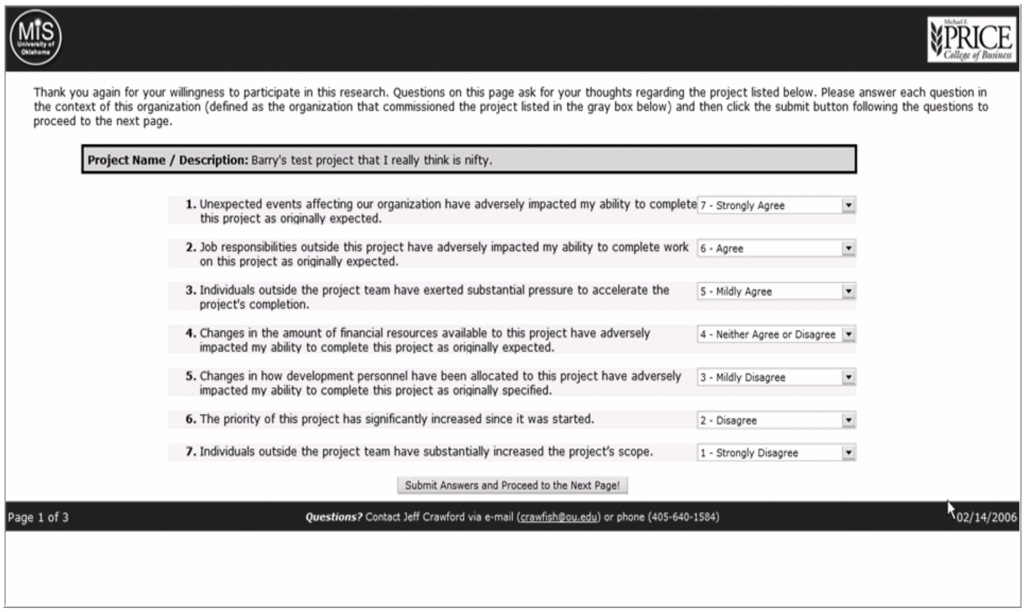
Voluntary Nature of the Study: Participation in this study is voluntary. Your decision whether or not to participate will not result in penalty or loss of benefits to which you are otherwise entitled. If you decide to participate, you are free to not answer any question or withdraw at any time.

Confidentiality: The results of the research study may be published, but your name and your organization's name will not be used. In fact, the results will be presented in summary form only; thus all information you provide will remain strictly confidential. Please note that there is no cost to you other than the time it takes to complete the survey.

Contacts and Questions: If you have any questions about this research project, please contact me at (405) 640-1584 or via e-mail at crawfish@ou.edu. Questions about your rights as a research participant or concerns about the project should be directed to the Institutional Review Board at The University of Oklahoma-Norman Campus at (405) 325-8110 or irb@ou.edu.

By entering your e-mail address in the field below and clicking the 'I Agree' button, you will be agreeing to participate in the above described project. Thanks for your continued contribution!

Questions? Contact Jeff Crawford via e-mail (crawfish@ou.edu) or phone (405-640-1584) 03/09/2006



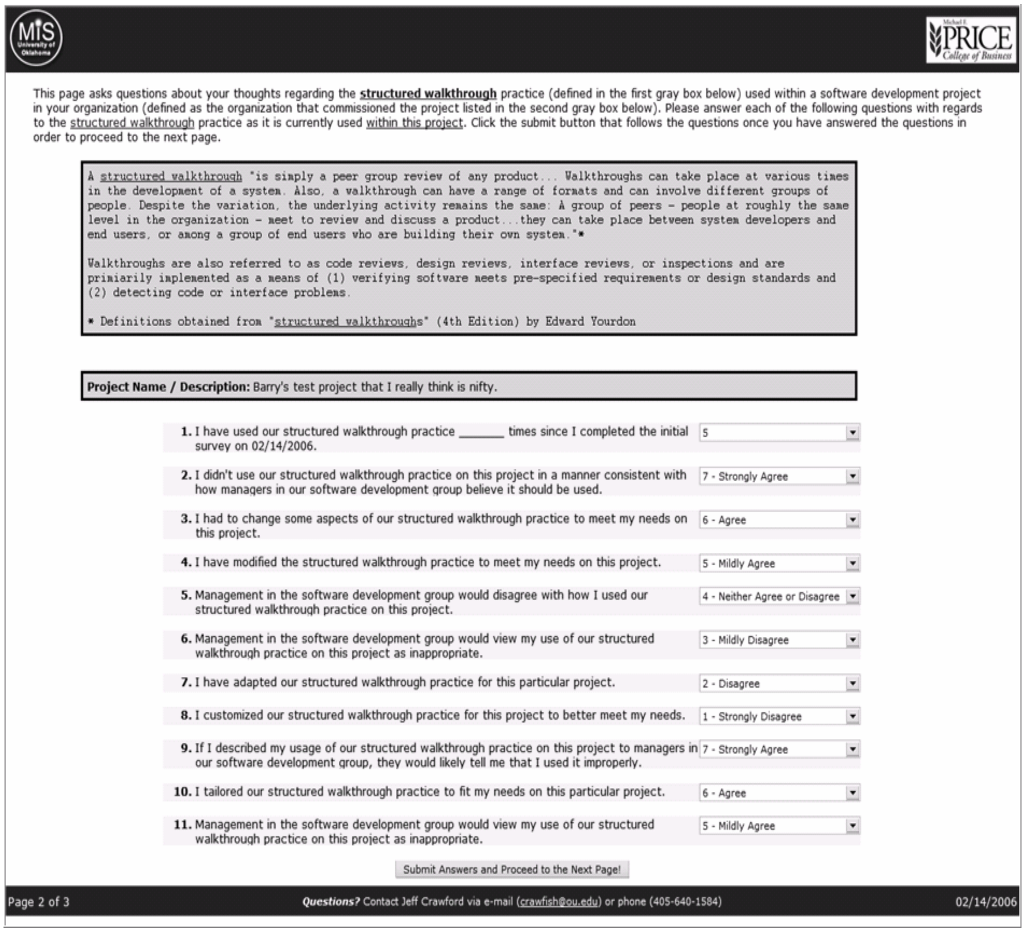
MIS University of Oklahoma
Michael F. PRICE College of Business
Submit Answers and Proceed to the Next Page!

Thank you again for your willingness to participate in this research. Questions on this page ask for your thoughts regarding the project listed below. Please answer each question in the context of this organization (defined as the organization that commissioned the project listed in the gray box below) and then click the submit button following the questions to proceed to the next page.

Project Name / Description: Barry's test project that I really think is nifty.

1. Unexpected events affecting our organization have adversely impacted my ability to complete this project as originally expected.	7 - Strongly Agree
2. Job responsibilities outside this project have adversely impacted my ability to complete work on this project as originally expected.	6 - Agree
3. Individuals outside the project team have exerted substantial pressure to accelerate the project's completion.	5 - Mildly Agree
4. Changes in the amount of financial resources available to this project have adversely impacted my ability to complete this project as originally expected.	4 - Neither Agree or Disagree
5. Changes in how development personnel have been allocated to this project have adversely impacted my ability to complete this project as originally specified.	3 - Mildly Disagree
6. The priority of this project has significantly increased since it was started.	2 - Disagree
7. Individuals outside the project team have substantially increased the project's scope.	1 - Strongly Disagree

Page 1 of 3 ***Questions?*** Contact Jeff Crawford via e-mail (crawfish@ou.edu) or phone (405-640-1584) 02/14/2006


MIS University of Oklahoma
Michael F. PRICE College of Business

This page asks questions about your thoughts regarding the **structured walkthrough** practice (defined in the first gray box below) used within a software development project in your organization (defined as the organization that commissioned the project listed in the second gray box below). Please answer each of the following questions with regards to the structured walkthrough practice as it is currently used within this project. Click the submit button that follows the questions once you have answered the questions in order to proceed to the next page.

A structured walkthrough "is simply a peer group review of any product... Walkthroughs can take place at various times in the development of a system. Also, a walkthrough can have a range of formats and can involve different groups of people. Despite the variation, the underlying activity remains the same: A group of peers - people at roughly the same level in the organization - meet to review and discuss a product...they can take place between system developers and end users, or among a group of end users who are building their own system."*

Walkthroughs are also referred to as code reviews, design reviews, interface reviews, or inspections and are priniarily implemented as a means of (1) verifying software meets pre-specified requirements or design standards and (2) detecting code or interface problems.

* Definitions obtained from "structured walkthroughs" (4th Edition) by Edward Yourdon

Project Name / Description: Barry's test project that I really think is nifty.

1. I have used our structured walkthrough practice _______ times since I completed the initial survey on 02/14/2006. 5

2. I didn't use our structured walkthrough practice on this project in a manner consistent with how managers in our software development group believe it should be used. 7 - Strongly Agree

3. I had to change some aspects of our structured walkthrough practice to meet my needs on this project. 6 - Agree

4. I have modified the structured walkthrough practice to meet my needs on this project. 5 - Mildly Agree

5. Management in the software development group would disagree with how I used our structured walkthrough practice on this project. 4 - Neither Agree or Disagree

6. Management in the software development group would view my use of our structured walkthrough practice on this project as inappropriate. 3 - Mildly Disagree

7. I have adapted our structured walkthrough practice for this particular project. 2 - Disagree

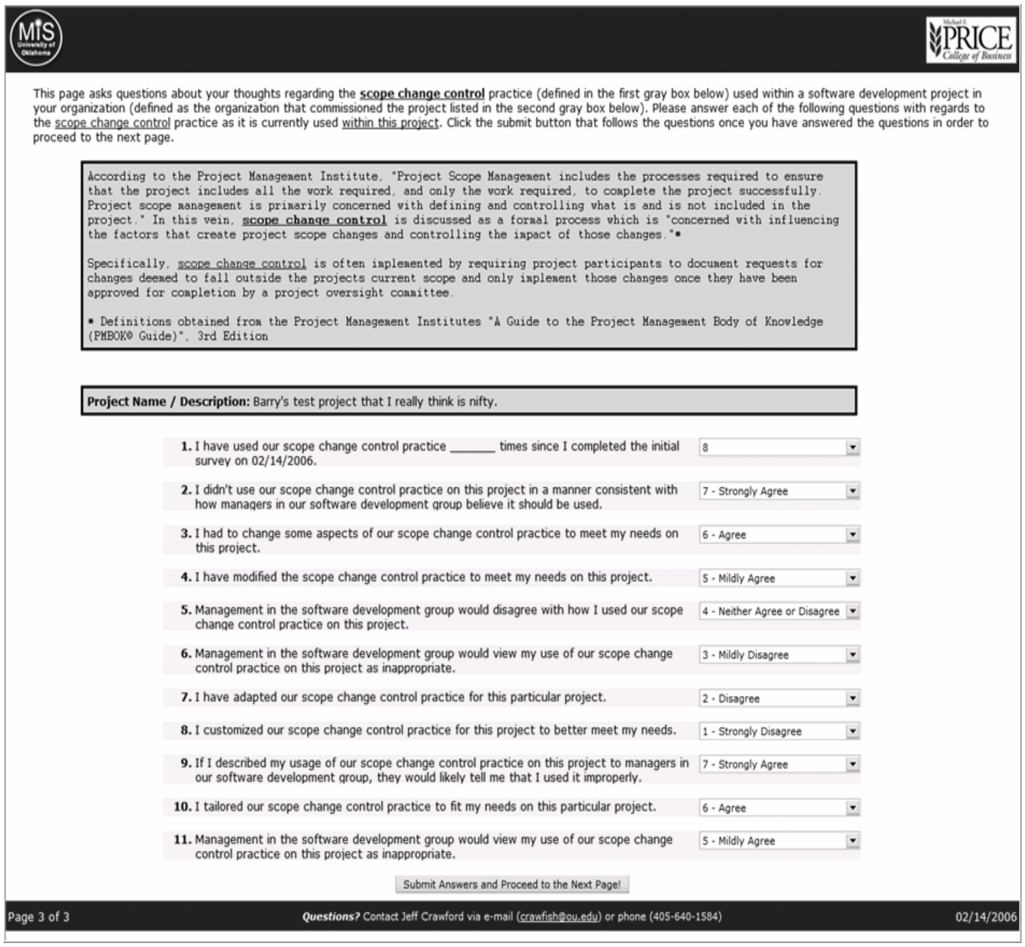
8. I customized our structured walkthrough practice for this project to better meet my needs. 1 - Strongly Disagree

9. If I described my usage of our structured walkthrough practice on this project to managers in our software development group, they would likely tell me that I used it improperly. 7 - Strongly Agree

10. I tailored our structured walkthrough practice to fit my needs on this particular project. 6 - Agree

11. Management in the software development group would view my use of our structured walkthrough practice on this project as inappropriate. 5 - Mildly Agree

Page 2 of 3 ***Questions?*** Contact Jeff Crawford via e-mail (crawfish@ou.edu) or phone (405-640-1584) 02/14/2006


MIS University of Oklahoma
Michael F. PRICE College of Business

This page asks questions about your thoughts regarding the **scope change control** practice (defined in the first gray box below) used within a software development project in your organization (defined as the organization that commissioned the project listed in the second gray box below). Please answer each of the following questions with regards to the scope change control practice as it is currently used within this project. Click the submit button that follows the questions once you have answered the questions in order to proceed to the next page.

According to the Project Management Institute, "Project Scope Management includes the processes required to ensure that the project includes all the work required, and only the work required, to complete the project successfully. Project scope management is primarily concerned with defining and controlling what is and is not included in the project." In this vein, **scope change control** is discussed as a formal process which is "concerned with influencing the factors that create project scope changes and controlling the impact of those changes."*

Specifically, scope change control is often implemented by requiring project participants to document requests for changes deemed to fall outside the projects current scope and only implement those changes once they have been approved for completion by a project oversight committee.

* Definitions obtained from the Project Management Institutes "A Guide to the Project Management Body of Knowledge (PMBOK® Guide)", 3rd Edition

Project Name / Description: Barry's test project that I really think is nifty.

1. I have used our scope change control practice _______ times since I completed the initial survey on 02/14/2006. 8

2. I didn't use our scope change control practice on this project in a manner consistent with how managers in our software development group believe it should be used. 7 - Strongly Agree

3. I had to change some aspects of our scope change control practice to meet my needs on this project. 6 - Agree

4. I have modified the scope change control practice to meet my needs on this project. 5 - Mildly Agree

5. Management in the software development group would disagree with how I used our scope change control practice on this project. 4 - Neither Agree or Disagree

6. Management in the software development group would view my use of our scope change control practice on this project as inappropriate. 3 - Mildly Disagree

7. I have adapted our scope change control practice for this particular project. 2 - Disagree

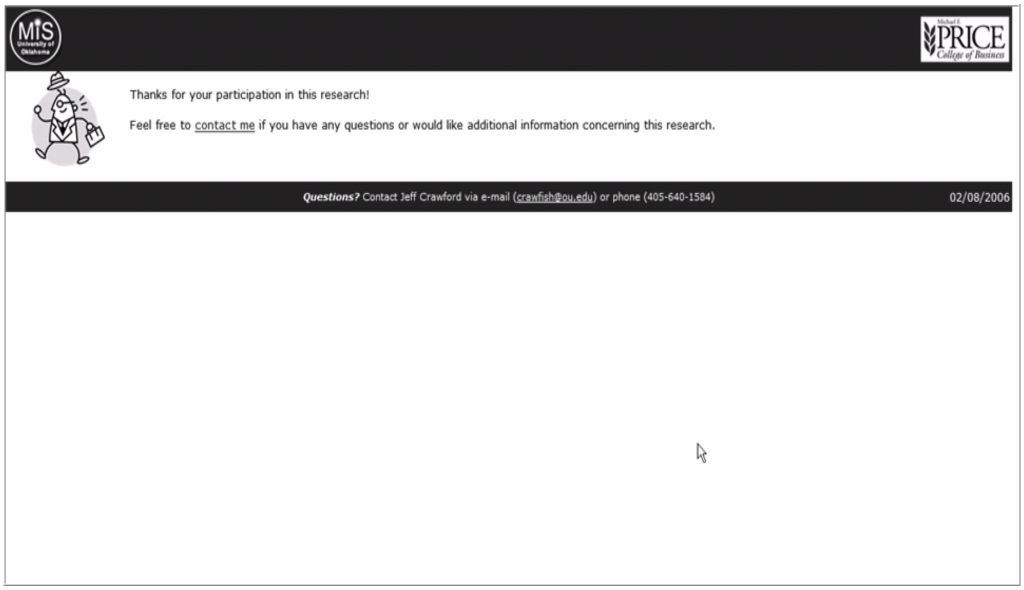
8. I customized our scope change control practice for this project to better meet my needs. 1 - Strongly Disagree

9. If I described my usage of our scope change control practice on this project to managers in our software development group, they would likely tell me that I used it improperly. 7 - Strongly Agree

10. I tailored our scope change control practice to fit my needs on this particular project. 6 - Agree

11. Management in the software development group would view my use of our scope change control practice on this project as inappropriate. 5 - Mildly Agree

Page 3 of 3 ***Questions?*** Contact Jeff Crawford via e-mail (crawfish@ou.edu) or phone (405-640-1584) 02/14/2006


MIS University of Oklahoma
Michael F. PRICE College of Business

Thanks for your participation in this research!

Feel free to contact me if you have any questions or would like additional information concerning this research.

Questions? Contact Jeff Crawford via e-mail (crawfish@ou.edu) or phone (405-640-1584) 02/08/2006

Appendix X – Customized Use Regression Results Controlling for Company

	Model 1				Model 2				Model 3			
Variable	***β***	*SE*	*p*	*VIF*	***β***	*SE*	*p*	*VIF*	***β***	*SE*	*p*	*VIF*
Constant	2.119	0.167	**0.000****		2.071	0.158	**0.000****		2.013	0.169	**0.000****	
SDPM Practice Complexity	-0.027	0.300	0.377	1.303	-0.058	0.030	**0.060^**	1.551	-0.048	0.032	0.129	1.695
Operational Exigency[A]	0.071	0.065	0.278	1.229	0.025	0.062	0.682	1.331	0.022	0.062	0.717	1.334
Frequency of Use	0.241	0.070	**0.001****	1.243	0.209	0.065	**0.002****	1.271	0.206	0.065	**0.002****	1.274
Developer Experience	-0.001	0.008	0.860	1.239	0.000	0.007	0.962	1.250	-0.001	0.007	0.908	1.256
SDPM Practice Faithful Usage[B]	-0.002	0.000	**0.000****	1.177	-0.003	0.000	**0.000****	1.255	-0.003	0.000	**0.000****	1.269
Company	0.004	0.012	0.764	1.181	0.007	0.011	0.515	1.215	0.007	0.011	0.556	1.220
Project Strategic Importance[C]	-0.001	0.003	0.797	1.197	0.001	0.003	0.751	1.247	0.001	0.003	0.776	1.248
High Institutionalization					0.161	0.089	**0.074^**	1.477	0.138	0.092	0.140	1.582
Informal Institutionalization					-0.103	0.089	0.249	1.462	-0.123	0.091	0.181	1.532
Formal Institutionalization					-0.079	0.077	0.313	1.275	-0.092	0.079	0.243	1.316
Instrumental Relative Advantage[C]					0.009	0.003	**0.009****	1.546	0.010	0.004	**0.006****	1.919
Formal Institutionalization*Institutional Relative Advantage[D]									-0.007	0.007	0.319	1.504
R^2 (adjusted R^2)	0.377 (0.330)				0.494 (0.434)				0.503 (0.434)			
ΔF	7.962				5.227				1.006			
df	92				88				87			
ΔR^2	0.377				0.120				0.006			
p	**0.000****				**0.001****				0.319			

Notes: ** significant at the 0.01 level (2-tailed), * significant at the 0.05 level (2-tailed), ^ significant at the 0.10 level (2-tailed), β reported is the unstandardized coefficient

[A] Natural Log Transformation

[B] Power transformation (cubed)

[C] Power transformation (squared)

[D] Instrumental Relative Advantage was centered by subtracting sample mean from individual scores, then interaction term was created

Appendix XI – Faithful Use Regression Results

Controlling for Company

	Model 1				Model 2				Model 3			
Variable	***β***	***SE***	***p***	***VIF***	***β***	***SE***	***p***	***VIF***	***β***	***SE***	***p***	***VIF***
Constant	394.620	51.457	**0.000****		432.690	51.131	**0.000****		413.022	52.089	**0.000****	
SDPM Practice Complexity	0.669	7.207	0.926	1.314	-5.125	7.541	0.498	1.762	-2.467	7.743	0.751	1.877
Operational Exigency^A	-18.492	15.335	0.231	1.225	-25.075	14.114	**0.079^**	1.271	-24.735	14.043	**0.082^**	1.271
Frequency of Use	63.221	16.408	**0.000****	1.209	57.202	15.204	**0.000****	1.271	55.539	15.172	**0.000****	1.279
Developer Experience	0.690	1.871	0.713	1.238	-0.306	1.744	0.861	1.318	-0.526	1.742	0.763	1.329
SDPM Practice Customized Usage^B	-133.859	20.476	**0.000****	1.096	-142.867	19.483	**0.000****	1.216	-142.351	19.386	**0.000****	1.216
Company	-1.769	2.827	0.533	1.177	-1.067	2.649	0.688	1.265	-1.245	2.639	0.638	1.268
Project Strategic Importance^C	0.444	0.683	0.517	1.192	0.695	0.628	0.271	1.233	0.660	0.625	0.294	1.235
High Instutionalization					29.296	19.353	0.263	1.119	24.326	16.943	0.218	1.312
Formal Instutionalization					-19.086	16.946	0.263	1.119	-21.426	16.943	0.209	1.130
Instrumental Relative Advantage^C					2.862	0.765	**0.000****	1.503	3.318	0.829	**0.000****	1.784
Political Relative Advantage					-17.803	8.566	**0.041***	1.532	-18.239	8.527	**0.035***	1.534
Formal Institutionalization*Institutional Relative Advantage^D									-2.211	1.596	0.169	1.419
R^2 (adjusted R^2)	0.420 (0.376)				0.547 (0.490)				0.557 (0.495)			
ΔF	9.508				6.164				1.920			
df	92				88				87			
ΔR^2	0.42				0.127				0.010			
p	**0.000****				**0.000****				0.169			

Notes: ** significant at the 0.01 level (2-tailed), * significant at the 0.05 level (2-tailed), ^ significant at the 0.10 level (2-tailed), β reported is the unstandardized coefficient

[A] Natural log transformation

[B] Square root transformation

[C] Power transformation (squared)

[D] Instrumental Relative Advantage was centered by subtracting sample mean from individual scores, then interaction term was created