Explaining Developer Attitude Toward Using Formalized Commercial Methodologies: Decomposing Perceived Usefulness

David L. Henderson III

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
In
General Business, Accounting

France Belanger, Co-Chair
Steven D. Sheetz, Co-Chair
Loren P. Rees
David P. Tegarden
Linda G. Wallace

September 7, 2007
Blacksburg, Virginia

Keywords: systems development methodologies, developer, attitude, perceived usefulness

Explaining Developer Attitude Toward Using Formalized Commercial Methodologies: Decomposing Perceived Usefulness

David L. Henderson III

ABSTRACT

Although methodology use generally leads to fewer software defects and reductions in development time, the introduction of a formalized systems development methodology is often met with substantial resistance. Motivated by the purported benefits of methodology use, yet resistance to the introduction of a methodology, this study explains developer attitude toward using a formalized commercial methodology.

An important variable for explaining attitude is perceived usefulness, defined as the degree to which using a methodology will enhance a developer's job performance. If, however, a benefit of using a methodology is different than increased job performance, then limiting the definition of perceived usefulness to beliefs surrounding job performance may provide an incomplete representation of what makes a methodology useful to developers. A methodology may be perceived as a rational process, used to achieve objectives such as increasing job performance or as a political process used to achieve objectives particular to one person or group. In order to determine what makes a methodology useful to developers, the perceived usefulness construct was expanded to include benefits of methodology use related toward achieving political objectives. In addition to broadening the perceived usefulness construct, this research also broke down perceived usefulness into its referent dimensions. Decomposing perceived usefulness provided a deeper understanding of what makes a methodology useful to developers and revealed the relative importance of each dimension of perceived usefulness.

The study surveyed 120 developers. Partial least squares regression was used to test the antecedents of developer attitude as well as the hypothesized structure of perceived usefulness. Results indicate that developers will have more favorable attitudes toward methodologies they perceive as useful, easy to use, and consistent with the way they like to develop systems. Additionally, findings suggest that developers may find methodologies not only useful for achieving rational goals such as increasing system quality, raising productivity, and enhancing communication, but also useful for achieving political goals such as increasing career opportunities, showing others that professional development practices are being used, reducing anxiety, and defending against unreasonable user demands.

"Never, never, never give up."
Winston Churchill

Dedication

To my wife, Jen, with love. Your love, support, understanding, encouragement, and editing skills helped me to get through the tough times and achieve this goal.

Acknowledgments

This endeavor could not have been completed without the help and support of my family and friends. I would first like to thank my wife, Jen, and my entire family, especially my mom and dad, brother, sister, step dad, step mom, grandparents, step sister, mother- and father in-law, brother in-law, and sister in-law for encouraging me to pursue this goal. I also appreciate the support from our friends, especially our Northern Virginia group and Jen's friends from college who were a big support to her while I was away at school.

I would like to thank my fellow doctoral students, especially Robin Romanus and Damon Fleming, for all of their support throughout the entire program. I'm glad we all went through the program together and I'm especially happy that we all made it through!!

I would also like to acknowledge my dissertation committee for their help and guidance throughout the dissertation process: Steve Sheetz (Co-Chair), France Belanger (Co-Chair), Loren Rees, Dave Tegarden, and Linda Wallace. I am especially grateful to Steve Sheetz for his advice, time, and encouragement throughout the entire four-year program.

I would also like to thank the following family members and friends who pre-tested my survey and listened to all of my dissertation ideas: Paul Henderson, Kira Semerjian, Barbara Henderson, Dave Henderson, Ray Weber, Bob Payne, Adam Brake, and Forrest Shull.

I would also like to thank the following friends and colleagues for their help with data collection: Geoff Garnhart, John Abdo, Dan Ehrlich, Sheryl Tolbert, Mark Ziegler, Dan Lantner, Duy Vu, Scott Ambler, Scott Gaydos, Barry Keiser, Pat Horner, Mike Mauer, Matt Ernst, Brian Lyons, Bernie Clark, Hach Vu, Rich Egan, John Mutarelli, Rob Strassheim, Ben Sutton, Courtney Akers, Vicente Ferguson, Sam Hupart, Anne Kennedy, William Money, and John Long.

I would like to thank the following colleagues for their help with Partial Least Squares regression: Patrick Fan, Jörg Henseler, Hui Lin, and John Sailors. I would like to thank Phyllis Neece, Arnita Perfater, Dinah Akers, and Anna Hawthorne for their administrative assistance. Finally, I would like to thank the Accounting and Information Systems department for its financial assistance over the last four years.

Table of Contents

List of Tables

List of Figures

Chapter One

Introduction

The high rate of software project failure has been extensively documented (Ewusi- Mensah, 1997). A staggering 53% of software projects are either late, over budget, or are implemented with fewer features than originally intended, and 18% of all software projects are canceled before completion (The Standish Group, 2004). Formalized information systems development methodologies have been touted as one way to reduce the high rate of software project failure (Fitzgerald, Russo & Stolterman, 2002).

Siau and Tan (2005) define a formalized information systems development methodology[1] as: “A systematic approach to conducting at least one complete phase of information systems development, consisting of a recommended collection of phases, techniques, procedures, tools and documentation aids” (p. 863). Methodologies lie at the core of the Information Systems discipline and recommend various, often vendor-specific procedures, techniques, tools, and documentation aids relevant to different parts of the development process (Nandhakumar & Avison, 1999). Constituting an important element of the information systems infrastructure (Sauer & Lau, 1997), methodologies are frequently promoted as vehicles for improving management and control of the development process (Russo & Stolterman, 2000).

Methodologies include both formalized in-house methodologies and formalized commercial methodologies. A formalized in-house methodology is proprietary to an

[1] For simplicity, a formalized information systems development methodology is referred to as a methodology.

organization (Fitzgerald, 1998a). A formalized commercial methodology, on the other hand, is used across organizations (Fitzgerald, 1998a).

Although research indicates that methodology use generally leads to fewer software defects and reductions in development time (Harter, Krishnan & Slaughter, 2000; Herbsleb, Carleton, Rozum, Siegel & Zubrow, 1994), there have been consistent findings indicating *low* usage of methodologies (Glass, 1996; Fitzgerald, 1996; Fitzgerald, 1997; Fitzgerald, 1998a; Hardy, Thompson & Edwards, 1995; Necco, Gordon & Tsai, 1987; Roberts, Gibson, Fields & Rainer, 1998; Russo, Wynekoop & Walz, 1995). Motivated by the low usage of methodologies, previous research has investigated developer intention to use a methodology through the lens of established adoption theory (Hardgrave, Davis & Riemenschneider, 2003; Riemenschneider, Hardgrave & Davis, 2002). However, because methodology use is often mandatory (Hardgrave et al., 2003), developer attitude toward using a methodology, rather than developer intention to use a methodology, may represent a more relevant dependent variable.

Mandatory usage contexts are problematic for the Theory of Reasoned Action (TRA; Ajzen & Fishbein, 1980), the Technology Acceptance Model (TAM; Davis, 1989) and the Diffusion of Innovations Theory (DIT; Rogers, 1983, 1995) because they all assume that users of information systems have a choice about the extent to which they use a technology (Rawstorne, Jayasuriya & Caputi 1998). A primary dependent variable in these adoption models—intention to use—has little meaning when one is forced to use a system (Brown, Massey, Montoya-Weiss & Burkman, 2002; Hartwick & Barki, 1994; Ward, Brown & Massey, 2005). As a result, *attitude* becomes the dependent variable of interest when use is highly mandated (Brown et al., 2002; Ward et al., 2005).

Investigating developer attitude toward methodology use constitutes an important research avenue as organizations attempting to implement a methodology often encounter substantial resistance from developers (Kozar, 1989; Raghavan & Chand, 1989). Negative developer attitude may impede organizational attempts to implement a methodology, thereby possibly hindering the potential of the methodology to mitigate software failure. Consistent with the recommendations of Brown et al. (2002) and motivated by the high levels of developer resistance to using methodologies (Kozar, 1989), this research seeks to explain developer *attitude* toward using a formalized commercial methodology.

Although prior research has investigated developer intention to use a formalized *in-house* methodology (Hardgrave et al., 2003; Riemenschneider et al., 2002), research has not investigated developer intention to use or developer attitude toward using a formalized *commercial* methodology. The significance of some independent variables may be different for a formalized *commercial* methodology than for a formalized *in-house* methodology. For example, because a formalized *commercial* methodology is used across organizations, it may provide greater career opportunities than a formalized *in-house* methodology. Accordingly, this research seeks to explain developer attitude toward using a formalized *commercial* methodology rather than developer attitude toward using a formalized *in-house* methodology.

An important variable for explaining attitude toward a new innovation is perceived usefulness. Perceived usefulness, a central construct in the TAM (Davis, 1989), is defined as the degree to which using a particular system would enhance a user's job performance (Davis, 1989). Prior research investigating developer intention to use a methodology has adopted this definition and defines perceived usefulness as the degree to which using a methodology increases a developer's job performance (Hardgrave et al., 2003; Riemenschneider et al., 2002). If, however, a benefit of using a methodology is different than increased job performance, then limiting the definition of perceived usefulness to beliefs surrounding improved job performance may provide an incomplete representation of what makes a methodology useful to developers. Research indicates that a methodology may be used to achieve rational objectives such as increasing system quality (i.e., job performance) or to achieve political objectives particular to one person or group, such as justifying decisions made during systems development (Robey & Markus, 1984). Additional research corroborates this perspective and indicates that methodologies play political roles that relate to using a methodology to realize goals particular to one person or group (Fitzgerald 1998b; Fitzgerald et al., 2002; Kautz, Hansen & Jacobsen, 2004; Nandhakumar & Avison, 1999, Robey & Markus, 1984; Sauer & Lau, 1997; Wastell, 1996; 1999).

The political roles played by methodologies constitute important factors in how and why developers use methodologies (Fitzgerald, 1998b; Fitzgerald et al., 2002). In order to determine what makes a methodology useful to developers, perceived usefulness should include benefits of methodology use related toward achieving political objectives in addition to beliefs surrounding increased job performance. However, despite their importance, the political roles of methodologies have not been well recognized in the Information Systems Development literature (Fitzgerald et al., 2002). Prior research has not operationalized aspects of perceived usefulness related to the political roles played by methodologies nor included them in a nomological network.

This research seeks to fill these gaps in the literature by expanding the perceived usefulness construct to capture benefits of methodology use related to the perception of a methodology as a vehicle for realizing political objectives. Furthermore, this research operationalizes the political roles played by methodologies and includes them in a theoretical model. A broadened perceived usefulness construct, in turn, may explain more variance in developer attitude toward using a methodology than the perceived usefulness construct from the TAM (Davis, 1989). Accordingly, this research will also test whether an expanded perceived usefulness construct will explain more variance in developer attitude toward using a formalized commercial methodology than the perceived usefulness construct from the TAM (Davis, 1989).

In addition to broadening perceived usefulness, this research aims to break-down the perceived usefulness construct into its proposed referent dimensions. Decomposing perceived usefulness should provide a greater understanding of what makes a methodology useful to developers and should reveal the relative importance of each proposed dimension of perceived usefulness. Gaining broader insights into the political roles played by methodologies is particularly important because they do not form a suitable basis on which to build committed use of methodologies (Fitzgerald et al., 2002). Dimensions of perceived usefulness related to political roles played by methodologies may lead developers to use a methodology inappropriately and subsequently fail to realize the full benefits of methodology use (Fitzgerald et al., 2002; Wastell, 1996).

Research Model and Independent Variables

Figure 1.1 presents the research model that will be tested in this study. The following sections elaborate on each independent variable in the research model.

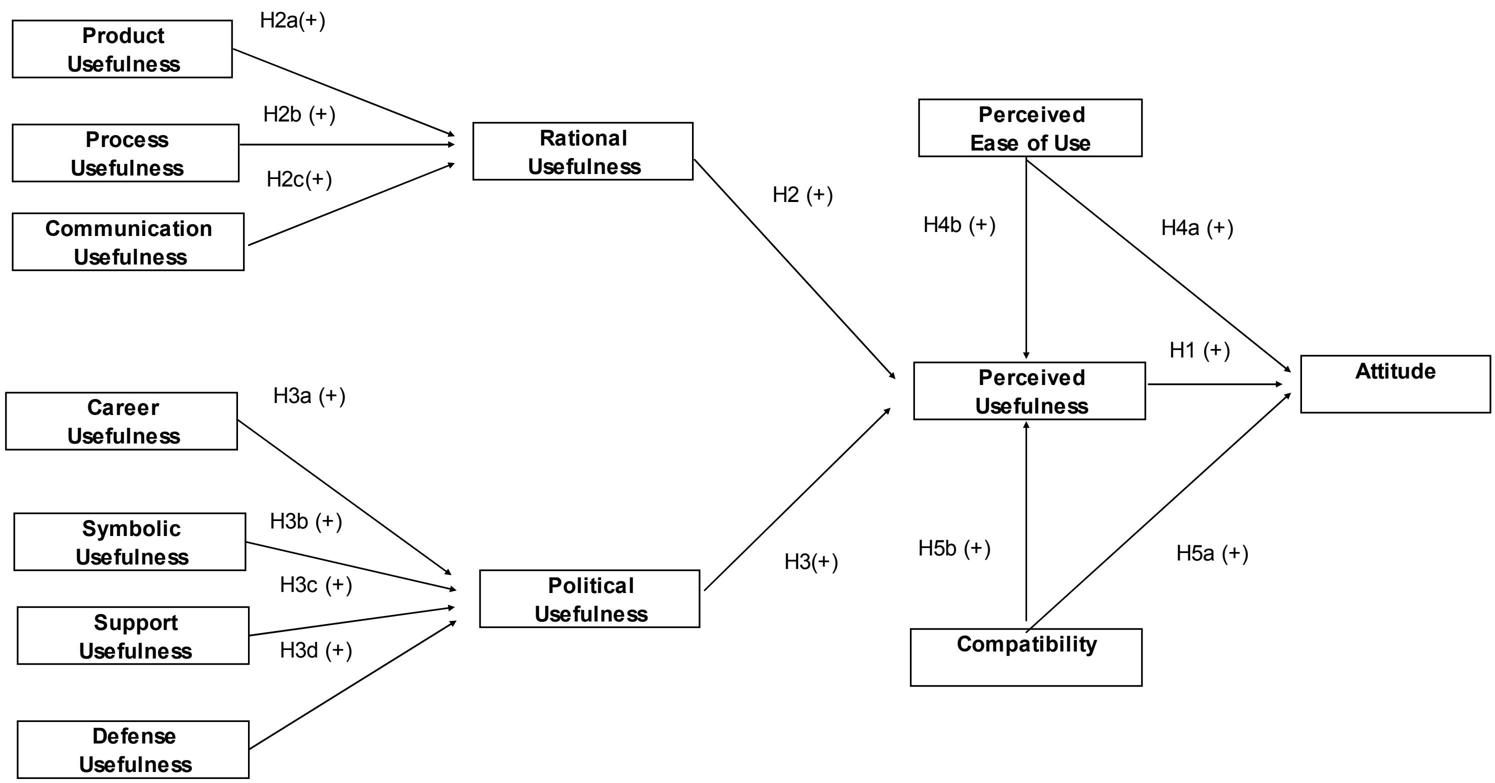

Figure 1.1 Research Model

Perceived usefulness

Perceived usefulness is defined as the degree to which using a particular system would enhance a user's job performance (Davis, 1989). Perceived usefulness is an important determinant of attitude toward using an innovation in both mandatory (Brown et al., 2002) and voluntary settings (Taylor & Todd, 1995) as well as for explaining developer intention to use a methodology (Hardgrave et al., 2003; Riemenschneider et al., 2002).

Rational usefulness [2]

Robey and Markus (1984) propose that the systems development process can be viewed as a rational process. The systems development process can be considered rational to the degree that it helps achieve two primary rational goals: first, to produce systems that are accepted and used appropriately, and second, to produce systems that enhance task performance and organizational effectiveness (Robey & Markus, 1984). Robey and Markus (1984) characterize a methodology as a rational process because it has been prescribed to meet these rational goals.

The perception of a methodology as a rational process suggests that it may be instrumental in achieving rational goals. For example, a methodology can be used to achieve higher system quality (Huisman & Iivari, 2006; Johnson, Hardgrave & Doke, 1999; Khalifa & Verner, 2000), a rational goal (Franz & Robey, 1984). Rational usefulness is defined as the degree to which using a methodology results in valuable outcomes agreed upon by stakeholders on the systems development project; specific valuable outcomes related to rational usefulness include increased task performance, organizational effectiveness and systems acceptance (Robey & Markus, 1984).

[2] Rational and political usefulness and their proposed referent dimensions are all perceived. For simplicity, these constructs are referred to as simply "rational" and "political" usefulness rather than perceived "rational" and "political" usefulness.

Dimensions of rational usefulness

In order to understand what makes a methodology useful to developers, it is important to decompose rational usefulness into its proposed referent dimensions. A review of the literature suggests that rational usefulness has three dimensions: product usefulness, process usefulness, and communication usefulness.

Product usefulness

Product usefulness is defined as the degree to which a methodology increases the quality of the developed system (Johnson et al., 1999; Huisman & Iivari, 2006; Khalifa & Verner, 2000). Although Khalifa and Verner (2000) find product quality, conceptually similar to product usefulness, is not a significant explanatory variable for methodology usage by individual developers, Johnson et al. (1999) find product usefulness is a dimension of perceived usefulness for Object-Oriented Systems Development (OOSD). Empirical research supports the findings of Johnson et al. (1999) and indicates methodology use is related to increases in software quality (Harter et al., 2000; Herbsleb et al., 1994).

Process usefulness

Another belief developers associate with a methodology is the degree to which it improves the productivity of the development process (Huisman & Iivari, 2006; Johnson et al., 1999; Khalifa & Verner, 2000). Process usefulness captures this idea and refers to the degree to which a methodology improves the productivity of the systems development process (Johnson et al., 1999; Huisman & Iivari, 2006; Khalifa & Verner, 2000). Prior research finds developers believe methodologies increase project control (Fitzgerald, 1998a). Furthermore, methodologies facilitate project management, help control the development process (Avison & Fitzgerald, 1995; Fitzgerald et al., 2002; Russo & Stolterman, 2000) and provide a useful framework for organizing systems development activities (Fitzgerald et al., 2002). Empirical research also finds a link between methodology use and increases in productivity (Harter et al., 2000; Herbsleb et al., 1994).

Communication usefulness

Communication usefulness is defined as the degree to which using a methodology improves communication with users and other team members (Johnson et al., 1999). Prior research finds that communication usefulness represents a dimension of perceived usefulness for a methodology (Johnson et al., 1999) and that communication with users represents a significant factor affecting perceptions of process quality for prototyping (Khalifa & Verner, 2000). Case study research corroborates these results and finds developers believe using a methodology can increase communication and coordination among various stakeholders on a software development project (Fitzgerald, 1998b; Fitzgerald et al., 2002).

Political usefulness

A methodology can be viewed as a political process because it can be used to achieve objectives particular to one group or individual to the relative disadvantage of other stakeholders (Robey & Markus, 1984). Furthermore, methodologies play several political roles, such as defending against unreasonable user demands (Fitzgerald 1998b; Fitzgerald et al., 2002; Kautz et al., 2004; Nandhakumar & Avison, 1999, Robey & Markus, 1984; Sauer & Lau, 1997; Wastell, 1996; 1999). The perception of a methodology as a political process coupled with research indicating that a methodology can play several political roles suggests it may be instrumental for achieving political objectives. Political usefulness is defined as the degree to which using a methodology results in valuable outcomes particular to one group or individual and may result in the relative disadvantage of other stakeholders on the systems development project.

Dimensions of political usefulness

Decomposing the political usefulness construct should help understand what makes a methodology useful to developers from a political perspective. A review of the research (Fitzgerald, 1998b; Fitzgerald et al., 2002; Johnson et al., 1999; Kautz et al., 2004; Nandhakumar & Avison, 1999; Sauer & Lau, 1997; Wastell, 1996; 1999) suggests political usefulness has four dimensions: career usefulness, symbolic usefulness, support usefulness, and defense usefulness.

Career usefulness

Career usefulness is defined as the extent to which using a methodology improves marketability, increases job flexibility, raises chances for promotion, and offers opportunity for more meaningful work (Johnson et al., 1999; Thompson, Higgins & Howell, 1991).

Johnson et al. (1999) find career usefulness represents a dimension of perceived usefulness for a methodology. Moreover, research on the motivation of Information Technology (IT) professionals suggests skill acquisition represents a key driver of motivation, satisfaction and turnover (Ferratt & Short, 1998; Igbaria, Parasuraman & Badawy, 1994).

Symbolic usefulness

Symbolic usefulness is defined as the defined as the degree to which using a methodology shows others that professional development processes are being used. Prior research finds rituals in systems development—using the systems development life cycle, for example—help to maintain the appearance of rationality (Robey & Markus, 1984). Case study research suggests methodologies help to "professionalize" development practices, thereby providing confidence that development decisions have been made on a systematic basis (Fitzgerald, 1998b; Fitzgerald et al., 2002). Additional findings from case study research further indicate that methodologies serve as a necessary "fiction" to present an image of control or provide a symbolic status (Kautz et al., 2004; Nandhakumar & Avison, 1999).

Support usefulness

Support usefulness is defined as the degree to which using a methodology reduces developer anxiety and provides feelings of security. Systems development is a stressful process, imposing pressures on developers, thereby raising their anxiety (Wastell & Newman, 1993). Developers may find a methodology useful for containing anxiety (Wastell, 1996; 1999; Nandhakumar & Avison, 1999).

Defense usefulness

Defense usefulness is defined as the degree to which using a methodology provides protection in case decisions made during systems development turn out to be wrong and protects against unreasonable user demands. Methodologies represent authority in that they not only outline how to conduct systems development, but also dictate how systems development should be conducted (Wastell, 1999). Thus, developers can draw upon the methodology to authorize their actions and attempt to absolve themselves of personal responsibility for their decisions (Wastell, 1999). Additionally, methodologies are used to insulate developers from unreasonable user deadlines and to provide an audit trail of development decisions (Fitzgerald, 1998b; Fitzgerald et al., 2002). By documenting all the steps in the development process and the rationale behind development decisions, some protection is provided in case development decisions turned out to be incorrect (Fitzgerald, 1998b; Fitzgerald et al., 2002).

Perceived ease of use

Perceived ease of use refers to the degree to which a developer perceives a methodology as difficult to use (Hardgrave et al., 2003). Prior research indicates perceived ease of use positively influences attitude toward using an innovation (Taylor & Todd, 1995) and perceived usefulness (Hardgrave et al., 2003).

Compatibility

Compatibility is defined as the degree to which a methodology is consistent with existing software development practices (Hardgrave et al., 2003). Prior research suggests that compatibility has a significant positive effect on developer intention to use a methodology (Riemenschneider et al., 2002; Hardgrave et al., 2003) and on attitude toward using an innovation (Taylor & Todd, 1995). Moreover, compatibility has a significant positive influence on perceived usefulness (Chau & Hu, 2001; Hardgrave et al., 2003; Oh, Anh, & Kim, 2003; Templeton & Byrd, 2003).

Expected Contributions

Overall, the anticipated results of this research should make several important contributions. By broadening the perceived usefulness construct, this research should provide a deeper understanding of what makes a methodology useful to developers. The expected results should show that methodologies are useful not only for achieving objectives related to the rational perspective, such as increasing system quality, they are also useful for achieving objectives related to the political perspective, such as enhancing career opportunities. This research should encourage researchers to explore the perceived usefulness construct in more depth for other technologies.

This research also intends to highlight the importance of the political roles played by methodologies and should encourage researchers to further explore their significance. The dimensions of perceived usefulness related to the political roles played by methodologies are not widely acknowledged in Information Systems Development literature (Fitzgerald et al., 2002) and have never been operationalized or included in a theoretical model. By operationalizing these dimensions, this research should lay the groundwork for further empirical research.

This research also aims to break-down perceived usefulness into its proposed referent dimensions in order to better understand what makes a methodology useful to developers. By decomposing the perceived usefulness construct, this study should show the relative importance of each dimension of perceived usefulness. Understanding the relative significance of the political dimensions of usefulness is important because facets of perceived usefulness related to the political roles played by methodologies may lead developers to use a methodology inappropriately (Fitzgerald et al., 2002). For example, research indicates that when developers use a methodology to simply demonstrate that systematic development processes are being used, they may use it in a superficial manner (Fitzgerald, 1998b; Kautz et al., 2004; Nandhakumar & Avison, 1999). Thus, developers may retrofit documentation to make it appear that the methodology has been used (Fitzgerald et al., 2002).

Furthermore, research indicates that when developers use a methodology to alleviate the stress and anxiety resulting from systems development (Wastell, 1996; 1999), they can be prone to goal displacement (Fitzgerald et al., 2002; Wastell, 1996; 1999). Goal displacement refers to a phenomenon wherein the developer becomes so absorbed with following the methodology that he or she loses sight of the real goal of systems development, i.e., developing the system (Fitzgerald et al., 2002). Thus, by blindly following the methodology, developers may perform unnecessary tasks.

Decomposing perceived usefulness into its proposed referent dimensions also has practical implications. By understanding the factors that make a methodology useful to developers, organizations can tailor training programs to foster positive attitudes toward methodology use.

Although prior research finds career consequences is not a significant explanatory variable for explaining developer intention to use a formalized *in-house* methodology (Riemenschneider et al., 2002), the predicted results of this study may show that career usefulness is an important aspect of what makes a formalized *commercial* methodology useful to developers. Because a formalized *commercial* methodology is used across organizations, whereas a formalized *in-house* methodology is used by a single organization, career usefulness may be a significant explanatory variable for explaining developer attitude toward using a formalized *commercial* methodology. The anticipated importance of career usefulness should emphasize the significance of skill acquisition for developers and underscore the point that researchers should not treat methodologies as homogeneous phenomena (Huisman & Iivari, 2006; Khalifa & Verner, 2000). Thus, the relative importance of each independent variable may vary depending on the methodology investigated. These results should encourage researchers to compare and contrast adoption models for different methodologies.

Finally, given that methodology use may be highly mandatory, this study investigates a more relevant dependent variable than intention to use—developer attitude toward using a formalized commercial methodology. Identifying the salient antecedents of developer attitude toward using a methodology represents an important research avenue, due to the high resistance to methodologies from developers (Kozar, 1989).

Overview of the Dissertation

The remainder of this document is structured as follows: Chapter Two contains a review of the literature and hypotheses development; Chapter Three includes the research design and methodology as well as items for operationalizing the constructs discussed in Chapter Two; Chapter Four contains a detailed analysis of the data collected; Chapter Five includes a discussion of the research results and their implications; and, Chapter Six provides a conclusion, including contributions, limitations of the research, as well as recommendations for future research.

Chapter Two

Literature Review and Hypotheses Development

Chapter Two is organized as follows: the first section provides an overview of methodologies; the second section discusses methodology usage, and methodology adoption by developers; and the final section elaborates on the dependent variable and independent variables in the research model.

Overview of Methodologies

Previous research has estimated the number of methodologies to range from hundreds (Avison & Fitzgerald, 1995) to over one thousand (Jayaratna, 1994). Numerous definitions for software development methodologies exist, resulting in confusion over the exact definition of a methodology (Fitzgerald et al., 2002; Huisman & Iivari, 2006; Siau & Tan, 2005).

Given the confusion surrounding the definition of a methodology, the question becomes: what is a formalized systems development methodology? Siau and Tan (2005) synthesize the definitions of a methodology from various sources (Avison & Fitzgerald, 1995, 2003; Lyytinen, 1987) and provide a reasonable definition. They define a methodology as: "A systematic approach to conducting at least one complete phase of information systems development, consisting of a recommended collection of phases, techniques, procedures, tools and documentation aides" (p. 863).

Additionally, Siau & Tan (2005) created a figure, Figure 2.1, based on the study by Iivari, Hirschheim, and Klein (2000), that is helpful for clarifying terminology surrounding methodologies.

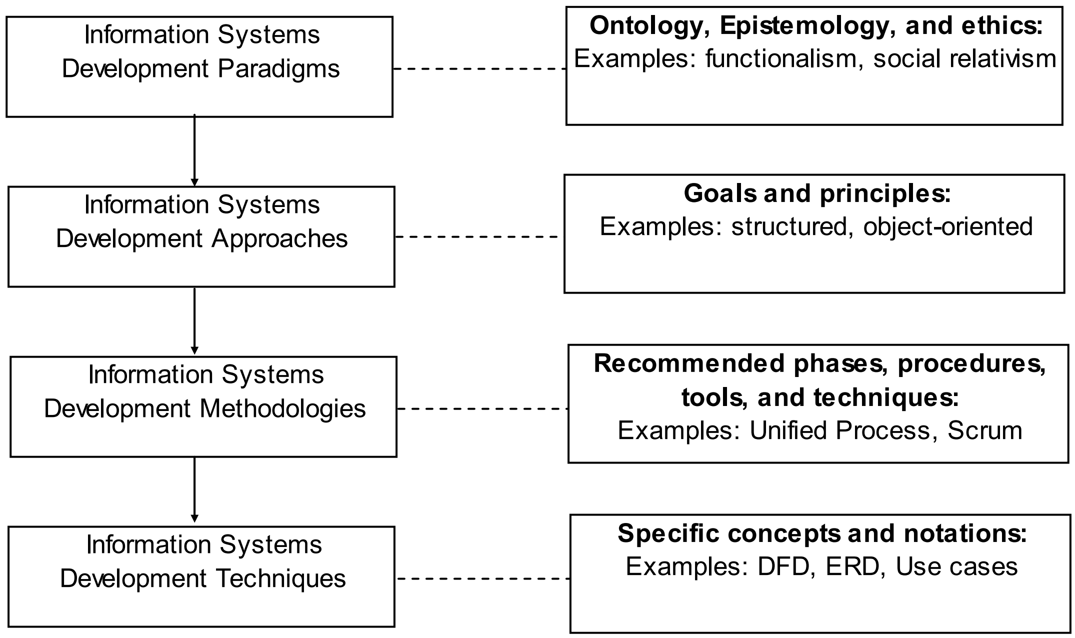

Figure 2.1 Hierarchy of ISD terms (Siau & Tan, 2005)

Information Systems Development (ISD) paradigms refer to a set of core beliefs held by the creator of specific ISD approaches or methodologies (Siau & Tan, 2005). Paradigms reflect beliefs revolving around the nature of reality (ontology), beliefs about how knowledge is acquired (epistemology) and the values that guide ISD (ethics) (Siau & Tan, 2005). The four main paradigms for ISD are functionalism, social relativism, neohumanism and radical structuralism (Hirschheim, Klein, Lyytinen, 1995). While the functionalist paradigm maintains that systems should be built to provide timely information to decision makers, the social relativist paradigm contends that systems should be created to facilitate sense making and mutual understanding (Hirschheim et al., 1995). Additionally, the radical structuralist paradigm asserts that systems should be developed to improve the conditions of the working class, whereas the neohumanist paradigm states that systems should be constructed to overcome injustice, social domination and other obstacles (Hirschheim et al., 1995).

ISD approaches represent collections of ISD methodologies sharing similar fundamental concepts, goals, guiding principles and beliefs, and principles for information systems development (Siau & Tan, 2005). Examples of approaches include the structured approach or the object-oriented approach.

Methodologies are defined as collections of procedures, including recommendations for tools and techniques, which steer the work of various stakeholders involved in conducting at least one complete phase of information systems development (Siau & Tan, 2005). Examples of methodologies include Extreme Programming (XP) and the Rational Unified Process (RUP).

Techniques are elementary operations, contained within a methodology, for conducting a specific activity in a particular phase of development (Siau & Tan, 2005). Examples of techniques include Entity-Relationship diagrams and Data-Flow diagrams.

Methodology Usage in Industry

While research findings differ as to the exact percentage of organizations and projects using methodologies, there have been consistent findings indicating low usage of formalized methodologies at both the organizational and project level. Hardy, Thompson and Edwards (1995) surveyed 100 companies in the United Kingdom. Results suggest that only 18% of organizations use a systems development methodology. Similarly, Chatzoglou and Macaulay (1996) surveyed 72 projects within the United Kingdom.

Findings reveal that less than half--47%--of organizations use a methodology. Russo, Hightower and Pearson (1996) surveyed 92 organizations as well as developers on individual projects. Results indicate that 20% of organizations claim to never use a methodology, 46% of organizations conduct systems development without a methodology at least occasionally, and 31% of systems development on individual projects is performed without any methodology.

In addition to low methodology usage at the project and organizational level, research also reports low methodology usage at the individual developer level. Fitzgerald (1998a) surveyed 162 developers and arguably conducted the most comprehensive study on the degree of methodology use by individual developers. Fitzgerald (1998a) finds, as shown in Table 2.1, that 60% of developers do not use a formalized systems development methodology.

Table 2.1 Methodology Usage (Fitzgerald, 1998a)
Copyright © 1998 with permission from Elsevier.

Methodology usage	Percentage
Percentage of organizations not using any Formalized Systems Development Methodology (FSDM)	60%
Percentage of organizations using FSDM	40%
Commercial FSDM	14%
Internal (based on commercial FSDM)	12%
Internal (not based on commercial FSDM)	14%

Fitzgerald (1998a) also finds that methodology usage is significantly higher in certain industries, in organizations consisting of more than 1,000 employees and in larger Information Systems departments with more than 20 personnel. Additionally, higher methodology usage is associated with high levels of in-house development and low levels of customization and outsourced development (Fitzgerald, 1998a). Finally, methodologies are more likely to be used on projects with more than five developers and when the project duration is greater than nine months (Fitzgerald, 1998a).

Methodology Adoption

Prior research has explored developer intention to use software development innovations such as programming languages, development techniques, and Computer Aided Systems Engineering (CASE) tools, (Agarwal & Prasad, 2000; Chau, 1996; Hardgrave, Wilson & Eastman, 1999; Iivari, 1996; Orlikowski, 1993; Palvia & Nosek, 1990; Sultan & Chan, 2000; Vessey, Jarvenpaa, & Tractinsky, 1992) and has investigated developer intention to use a methodology through the lens of adoption theory (Hardgrave et al., 2003; Hardgrave & Johnson, 2003; Huisman & Iivari, 2002; Khalifa & Verner, 2000; Riemenschneider et al., 2002).

The studies by Hardgrave and Johnson (2003), Riemenschneider et al. (2002), and Hardgrave et al. (2003) arguably represent the most theoretically grounded empirical studies on developer adoption of methodologies. Hardgrave and Johnson (2003) surveyed 150 developers and investigated the adoption of Object-Oriented Systems Development (OOSD) using the Theory of Planned Behavior (TPB), and goal-setting theory. Their theoretical model, shown in Figure 2.2, explains 63% of the variance of intention to use OOSD and indicates that perceived organizational usefulness is a direct determinant of intention to use. Subjective norms and perceived internal behavior control exhibit indirect effects on intention to use OOSD through perceived organizational usefulness.

In the context of methodologies, subjective norm refers to the degree to which people think that others who are important to them think they should use a methodology (Ajzen, 1988). Perceived organizational usefulness refers to a developer's perception of how using a methodology would be useful in performing systems development for the benefit of the organization (Hardgrave & Johnson, 2003). Perceived internal behavioral control is conceptually similar to perceived ease of use (Mathieson, 1991). In the context of methodologies, perceived internal behavioral control refers to the degree to which a developer perceives a methodology as difficult to use (Hardgrave & Johnson, 2003).

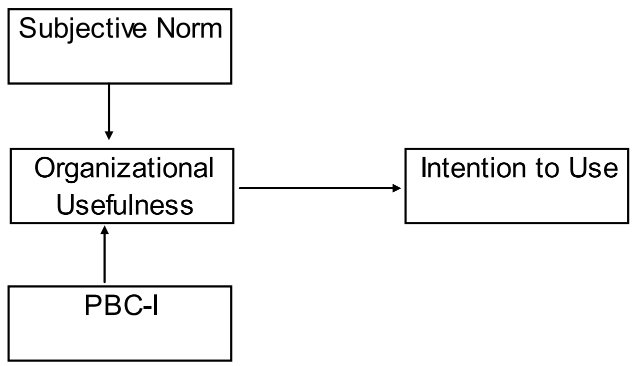

Figure 2.2 Information Systems Development Methodology Acceptance Model (Hardgrave and Johnson, 2003)

Riemenschneider et al. (2002) surveyed 128 developers using constructs from the Technology Acceptance Model (TAM), TAM2, Perceived Characteristics of Innovating (PCI), the Theory of Planned Behavior (TPB), and the Model of Personal Computer Utilization (MPCU) to explain individual developer adoption of methodologies. They find the following constructs to be significant across all five adoption models: perceived usefulness, subjective norm, voluntariness, and compatibility. Perceived usefulness refers to the degree to which a developer believes that using a methodology will improve his or her job performance (Hardgrave et al., 2003). Perceived social pressure, conceptually similar to subjective norm, refers to the degree to which a developer experiences inter-personal influence to use a methodology from important others in his or her environment

(Hardgrave et al., 2003). Perceived complexity refers to the extent to which a developer regards a methodology as difficult to use or learn (Hardgrave et al., 2003). Perceived compatibility refers to the extent to which a developer regards the practice of using a methodology as consistent with his or her existing development practices (Hardgrave et al., 2003). Perceived organizational mandate is conceptually similar to voluntariness and refers to the degree to which a developer believes that using a methodology has been dictated by official policy in effect within his or her organization (Hardgrave et al., 2003). Figure 2.3 shows the resulting model from Hardgrave et al. (2003).

Hardgrave et al. (2003) extended the work of Riemenschneider et al. (2002) and tested these constructs in an integrated path model. The findings of Hardgrave et al. (2003) indicate that 64% of the variance in intention to use a methodology can be explained by organizational mandate, perceived usefulness, social pressure, and compatibility.

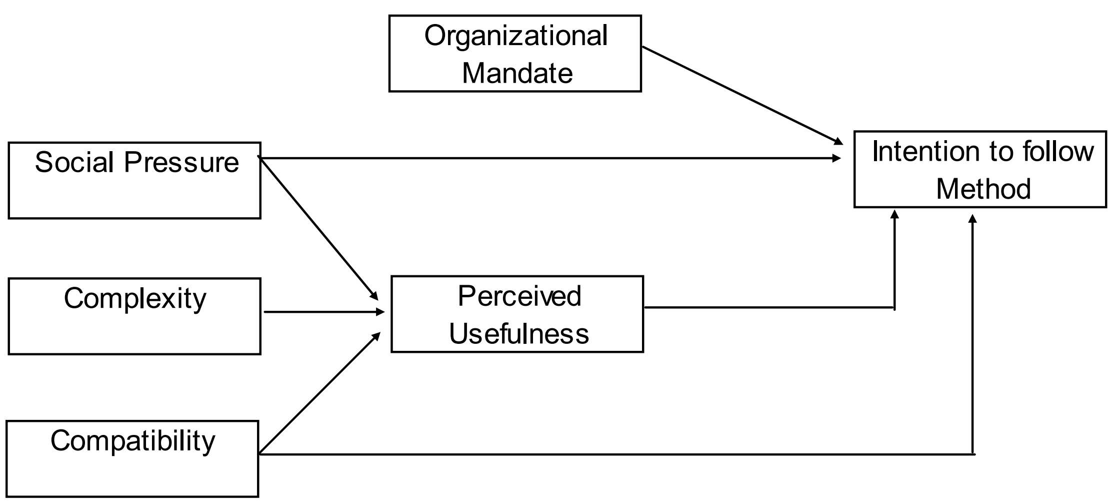

Figure 2.3 Methodology adoption model (Hardgrave et al., 2003).
Copyright © 2003 by M.E. Sharpe, Inc. Used by permission.

Developer Attitude toward Methodologies

The use of an information system in an organizational environment is becoming increasingly mandatory (Ram & Jung, 1991). Brown et al. (2002) offer two criteria for determining whether technology usage is highly mandatory in an organizational environment: the extent to which a technology is necessary for one's job and the extent to which a technology is tightly integrated with the tasks of other job performers.

Methodology use meets these two criteria because methodologies are becoming a necessity for software development and because methodology use is tightly integrated with the tasks of other developers on a given project team. Due to the need to retain Capability Model Maturity Integration (CMMi) certification (Duggan, 2004; Riemenschneider et al., 2002) and satisfy client mandate (Fitzgerald, 1996, 1998a), methodologies are necessary for software development. Methodologies have high levels of user inter-dependencies (Fichman, 1992) because their use is tightly integrated with the tasks of other developers on the project. Thus, it would be unreasonable for one developer on the project team to use a different methodology than the one chosen by the development team or the project manager. Furthermore, methodology use tends to be more mandated by upper-level management than tool use (Hardgrave et al., 2003).

Although Hardgrave et al. (2003) find organizational mandate is insufficient for methodology adoption, Hardgrave et al. (2003) may have investigated mandate at the wrong level of granularity. If the project manager decides to use a certain methodology, developers have little choice to either not use the methodology or use a different methodology. Thus, the appropriate level of mandate may be at the project level rather than at the organizational level.

Additionally, the organizational mandate in the Hardgrave et al. (2003) study is relatively weak. Hardgrave et al. (2003) state that the methodology was mandated via a memorandum from the Chief Information Officer and that there were no rewards or punishments for use or non-use. Further evidence of the weak organizational mandate is indicated by inspecting path coefficients of the constructs; voluntariness has the lowest path coefficient in the Hardgrave et al. (2003) study.

Mandatory usage contexts are difficult for the Theory of Reasoned Action (TRA; Azjen & Fishbein, 1980) and TAM (Davis, 1989) and Diffusion of Innovations Theory (DIT; Rogers, 1983; 1995) because they assume users of information systems have volitional control over their choice to use a technology. The main dependent variable in these adoption models, intention to use, has little meaning in a mandatory use environment (Brown et al., 2002). Rather, research suggests that in mandatory usage situations, attitude becomes the dependent variable of interest (Brown et al., 2002). Consistent with these arguments, this research investigates developer *attitude* toward using, rather than developer *intention to use,* a formalized commercial methodology (Brown et al., 2002).

In the context of methodologies, attitude refers to the degree to which a developer has favorable or unfavorable evaluations of using a methodology (Ajzen, 1988, 1991). An analysis of prior research on methodology use reveals most developers have a negative attitude toward using a methodology. For example, developers view the impact of a formalized methodology as a negligible factor in the success of a development project (Fitzgerald, 1998a). Additionally, Raghavan and Chand (1989) suggest methodologies are incompatible with the *ad hoc* systems development practices preferred by developers.

While it may be feasible to force developers to use a methodology, attitude can have an effect on the implementation process (Melone, 1990). Negative developer attitude may obstruct organizational attempts to implement a methodology, thereby possibly hindering the capability of the methodology to realize objectives such as higher software quality. For example, prior research finds employees who do not wholeheartedly accept an innovation can delay or obstruct its implementation (Markus, 1983; Leonard-Barton, 1988; Zuboff, 1988). Furthermore, while employees may use the technology in a mandatory environment, their job satisfaction, feelings toward their supervisors and loyalty to the organization may be negatively impacted (Zuboff, 1988), thereby leading to sabotage and unfaithful appropriation (Markus, 1983; Leonard-Barton, 1988; Zuboff, 1988). The high organizational resistance often put forward by developers against methodologies constitutes further testimony to the problems associated with negative attitude in the domain of methodology use (Kozar, 1989).

Explaining Developer Attitude toward Using a Methodology

This study uses the Decomposed Theory of Planned Behavior (DTPB; Taylor & Todd, 1995) as a theoretical framework for explaining developer attitude toward using a methodology. The DTPB includes three antecedents to attitude: relative advantage, compatibility, and complexity (Taylor & Todd, 1995). Relative advantage corresponds to perceived usefulness in the TAM (Davis, 1989). Compatibility refers to the degree to which an innovation is consistent with existing values, past experiences and needs of potential adopters (Rogers, 1983; 1995). Complexity corresponds to perceived ease of use in the TAM and is defined as the degree to which an innovation is difficult to understand and use (Davis, 1989). For the sake of consistency with prior literature, relative advantage is referred to as perceived usefulness and complexity is referred to as perceived ease of use (Taylor & Todd, 1995).

The DTPB has been used to explain the adoption of various information technologies, including internet banking (Shih, 2004), computer resource center use (Taylor & Todd, 1995), user acceptance of Wireless Application Protocol (WAP) services (Hung & Chang, 2005), adoption of mobile commerce (Pedersen, 2005), and adoption of Microsoft Access (Thompson, Compeau & Higgins, 2006). Additionally, the DTPB explains more variance in *attitude* than the TRA (Shih, 2004), the Theory of Planned Behavior (TPB; Shih, 2004; Taylor & Todd, 1995), and the TAM (Taylor & Todd, 1995).

The DTPB also includes the compatibility construct, whereas the TAM, the TRA, and the TPB do not. Due to the magnitude of change required by a methodology, compatibility is a highly salient independent variable for explaining developer intention to use a methodology (Hardgrave et al., 2003; Riemenschneider et al., 2002). Thus, using only the TAM, composed of perceived usefulness and perceived ease of use, and therefore excluding compatibility, may provide an incomplete picture of the factors explaining developer attitude toward using a formalized commercial methodology. The constructs presented in the research model shown in Figure 2.4 are further explored in the remainder of this section.

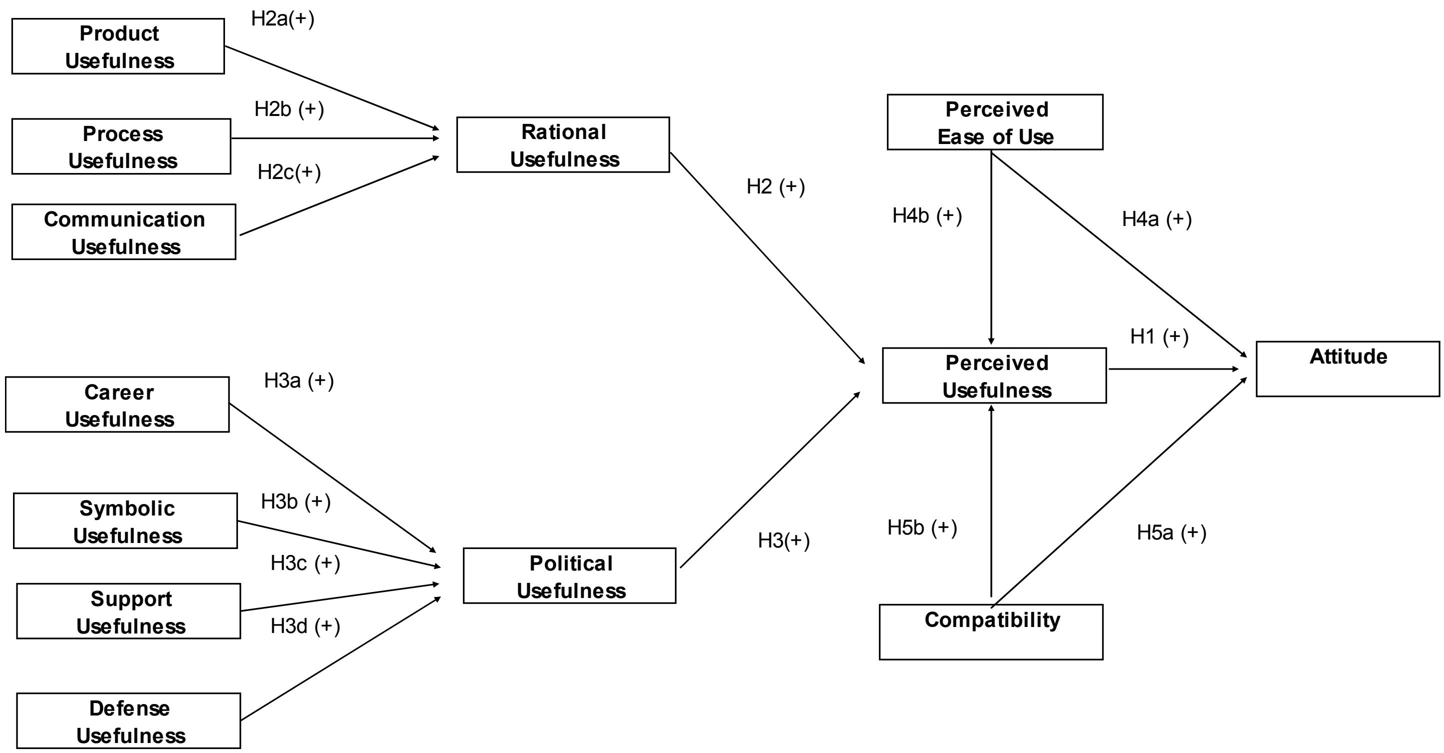

Figure 2.4 Research Model

Perceived usefulness

Perceived usefulness is based on the idea of extrinsic motivation (Brown et al., 2002; Venkatesh, Morris, Davis & Davis, 2003), which refers to the perception that users will perform an activity because it is instrumental in achieving valued outcomes that are distinct from the activity itself, such as improved job performance, pay, or promotions (Davis, Bagozzi & Warshaw, 1992). Perceived usefulness refers to the degree to which a person believes using a particular system will enhance his or her job performance (Davis, 1989; Davis et al., 1992). In the context of methodologies, perceived usefulness refers to the degree to which using a methodology will improve a developer's job performance (Hardgrave et al., 2003).

Research finds perceived usefulness to be an important variable for explaining an individual's attitude toward using an innovation in a voluntary setting (Taylor & Todd, 1995), in a mandatory setting (Brown et al., 2002) and for explaining developer intention to use a methodology (Hardgrave et al., 2003; Riemenschneider et al., 2002). Consistent with these findings, perceived usefulness should have a positive influence on developer attitude toward using a formalized commercial methodology.

> *H1: Higher levels of perceived usefulness of a formalized commercial methodology will lead to higher levels of positive developer attitude toward using a formalized commercial methodology.*

A close examination of the definition of perceived usefulness and its associated items reveals that it refers to increasing productivity and quality of work. For example, items used to operationalize perceived usefulness in the TAM (Davis, 1989) such as "Using the system in my job would enable me to accomplish tasks more quickly" and "Using the system would improve my job performance" relate to using a system to improve productivity. Limiting perceived usefulness to beliefs about productivity and quality of work may not capture all benefits of methodology use. As discussed further in the following sections, research indicates that a methodology may be viewed as a rational process, used to achieve identifiable and agreed-upon objectives, such as increasing system quality, or as a political process used to achieve objectives specific to one person or group, such as justifying decisions made during systems development (Robey & Markus, 1984). This perspective suggests that methodologies may have both rational and political usefulness.

Rational usefulness

Robey and Markus (1984) propose that the systems development process can be viewed from a rational perspective. They identify two requirements for a process to be considered rational: first, the process should have a set of identifiable and agreed-upon goals, and second, the process should be prescribed to achieve those goals (Robey & Markus, 1984). The systems development process can be considered rational to the extent that it helps achieve two goals: first, to produce systems that are accepted and used appropriately, and second, to produce systems that enhance task performance and organizational effectiveness (Robey & Markus, 1984). Robey and Markus (1984) characterize a methodology as a rational process because it has been prescribed to meet these two goals.

As shown in Table 2.2, additional case study research corroborates this perspective and indicates that methodologies play several rational roles related to improving the productivity of the development process and raising system quality (Fitzgerald, 1998b; Fitzgerald et al., 2002).

Table 2.2 Rational roles of methodologies

Rational Role	Description	Source
Reduction of complexity of ISD	Methodologies break down the development process into manageable parts, thereby reducing the complexity of development.	Fitzgerald, 1998b; Fitzgerald et al., 2002
Facilitation of project management and control	The phased approach recommended by methodologies improves project management and control.	Fitzgerald, 1998b; Fitzgerald et al., 2002
Division of labor	Methodologies help create an appropriate division of labor, thereby allowing management to develop pay rates by skill-set rather than pay one flat pay rate for all employees.	Fitzgerald, 1998b; Fitzgerald et al., 2002
Systematization of development knowledge	Methodologies accumulate best development practices, thereby reducing the dependency of the team on one person.	Fitzgerald, 1998b; Fitzgerald et al., 2002
Standardization of development process	Methodologies standardize the development process, thereby improving communication and coordination among team members.	Fitzgerald, 1998b; Fitzgerald et al., 2002

Methodologies decompose the development process into manageable parts, thereby helping to reduce the complexity of information systems development. The phased approach recommended by many methodologies aids in project management and control. Methodologies also provide a purposeful framework for organizing development activities and facilitate an appropriate division of labor. By breaking-down the development process into discrete tasks requiring different skill sets, methodologies help managers enable skill specializations. The skill specializations afforded by methodologies allow organizations to use differential pay rates based on certain skills. Methodologies promote learning by recommending the review of lessons learned during software project post-mortems. Methodologies collect best practices and serve as knowledge repositories, thereby reducing the dependency of the team on a single team member. Finally, methodologies standardize the development process, thereby facilitating the interchangeability of developers and helping new developers quickly learn the development practices of the project team. By standardizing the development process, methodologies also promote coordination and communication among team members.

The perception of a methodology as a rational process coupled with research indicating a methodology can play several rational roles suggests that a methodology may be instrumental in achieving rational objectives. For example, a methodology can be used to achieve higher system quality (Huisman & Iivari, 2006; Johnson et al., 1999; Khalifa & Verner, 2000), a rational goal (Franz & Robey, 1984). Accordingly, rational usefulness is defined as the degree to which using a methodology results in valuable outcomes that are agreed upon by stakeholders on the systems development project; specific valuable outcomes related to rational usefulness include increased task performance, organizational effectiveness and systems acceptance (Robey & Markus, 1984). This definition mirrors the definition of the perceived usefulness construct from the TAM (Davis, 1989) because it reflects beliefs associated with increasing job performance (i.e., improving productivity and quality). Based on research suggesting that a methodology may be instrumental in achieving rational objectives (Fitzgerald, 1998b; Fitzgerald et al., 2002; Robey & Markus, 1984), rational usefulness should be a significant dimension of perceived usefulness.

H2: Rational usefulness will be a significant component of perceived usefulness.

Dimensions of rational usefulness

In order to understand what makes a methodology useful to developers, it is important to decompose rational usefulness into its proposed referent dimensions. Johnson et al. (1999) find that product usefulness, process usefulness, and communication usefulness represent factors of perceived usefulness for OOSD. Similarly, Khalifa and Verner (2000) maintain that two benefits developers associate with a methodology are the extent to which it raises the productivity of the development process and the degree to which it increases the quality of the developed system. These findings suggest rational usefulness is composed of three dimensions: product usefulness, process usefulness, and communication usefulness (Fitzgerald 1998b; Fitzgerald et al., 2002; Johnson et al., 1999; Khalifa & Verner, 2000).

Product usefulness, process usefulness, and communication usefulness are first-order reflective constructs. Rational usefulness is hypothesized to be a second-order molar construct[3]. A molar construct is a combination of beliefs into one single construct in which the first-order constructs are reflective and higher-order, or molar constructs, are formative (Bagozzi, 1985; Chin & Gopal, 1995; Chin, 2000). The beliefs forming a molar construct are independent and form the higher-order construct (Bagozzi, 1985; Chin & Gopal, 1995; Chin, 2000). With a molar construct, an increase in the value of a lower-order construct translates into a larger score for the higher-order construct, regardless of the value for the other lower-order constructs (Bagozzi, 1985; Chin & Gopal, 1995). The final score for the higher-order construct is the sum of the weighted scores on all the lower-order constructs. In the molar approach, the lower-order constructs that form the

[3] The term "molar construct" is synonymous with a formative or emergent construct (Chin & Gopal, 1995)

higher-order molar construct are assumed to be uncorrelated with each other and may not represent the same underlying dimension (Chin, 1998a).

On the other hand, with the molecular[4] approach, each lower-order construct reflects an existing overall belief (Bagozzi, 1985; Chin & Gopal, 1995). The lower-order constructs serve as a representation of the higher-order construct and hence should exhibit high correlations (Chin & Gopal, 1995). Furthermore, in the molecular approach, the lower-order constructs are viewed as being caused by some underlying factor (Bagozzi, 1985; Chin & Gopal, 1995); an increase in the value of the higher-order construct translates into an increase in the value of the lower-order constructs (Bagozzi, 1985; Chin & Gopal, 1995).

Bollen (1989) recommends a “mental experiment” for choosing between formative or reflective indicators, which can also be applied when choosing between the molar or molecular orientation. The researcher envisions a change in the latent construct (higher-order construct) and then decides whether it is reasonable to expect a subsequent change in the observed variables (lower-order constructs). If the observed variables (lower-order constructs) change as a result of a change in the latent construct (higher-order construct), a reflective or molecular model is appropriate. On the contrary, if a change in one variable (lower-order construct) yields a change in the latent variable (higher-order construct), a formative, or molar model is appropriate.

An application of this mental experiment to the dimensions of perceived usefulness—rational and political usefulness—suggests that perceived usefulness is a third-order molar, rather than molecular, construct. For example, an increase in rational usefulness will not necessarily result in an increase in political usefulness. On the

[4] The term “molecular construct” is synonymous with a reflective construct (Chin & Gopal, 1995)

contrary, an increase in either rational or political usefulness will result in a change in perceived usefulness. Following this rationale, perceived usefulness is hypothesized to be a third-order molar construct composed of rational and political usefulness.

Applying this mental experiment to the dimensions of rational usefulness, a change in rational usefulness will not necessarily result in a change in all dimensions of rational usefulness (Bollen, 1989). On the other hand, a change in one of the dimensions of rational usefulness will result in a change in rational usefulness (Bollen, 1989). These arguments suggest that rational usefulness is a second-order molar construct, rather than a second-order molecular construct, composed of product usefulness, process usefulness, and communication usefulness (Bollen, 1989). Each proposed dimension of rational usefulness is a reflective first-order construct measured by multiple items. Table 2.3 summarizes the dimensions of rational usefulness.

The use of a molar construct provides several important benefits. First, molar constructs, because they are multi-dimensional, can reduce potential aggregation biases, thereby increasing prediction and explanation of the dependent variable (Bagozzi, 1988; Chin & Gopal, 1995). Second, the use of molar constructs helps researchers link the lower-order constructs and understand how beliefs are formed (Bagozzi, 1988; Chin & Gopal, 1995). Third, the molar model provides a lower level of detail—i.e., the dimensions of perceived usefulness—that can make the research model more managerially relevant (Chin & Gopal, 1995).

Table 2.3 Rational Usefulness and its Dimensions

Dimension	Definition	Source
Rational Usefulness	The degree to which using a methodology results in valuable outcomes that are agreed upon by stakeholders on the systems development project; specific valuable outcomes related to rational usefulness include increases in task performance, organizational effectiveness and systems acceptance.	Robey & Markus, 1984
Product Usefulness	The degree to which using a methodology improves the quality of the developed system.	Huisman & Iivari, 2006; Johnson et al., 1999; Khalifa & Verner, 2000
Process Usefulness	The degree to which using a methodology improves the productivity of the systems development process.	Huisman & Iivari, 2006; Johnson et al., 1999; Khalifa & Verner, 2000
Communication Usefulness	The degree to which using a methodology improves communication with other team members and users.	Johnson et al., 1999

Product usefulness

Prior research maintains that one belief developers associate with a methodology is the extent to which it improves the quality of the developed system (Huisman & Iivari, 2006; Johnson et al., 1999; Khalifa & Verner, 2000). Product usefulness encapsulates this idea and is defined as the degree to which using a methodology increases the quality of the developed system. Product usefulness is conceptualized as a dimension of rational usefulness because improving system quality would be considered a rational objective (Franz & Robey, 1984).

Prior research finds that product usefulness is a dimension of perceived usefulness for a methodology. Johnson et al. (1999) conducted an exploratory factor analysis based on a survey of 150 systems developers to identify specific salient beliefs regarding

OOSD underlying the attitude, subjective norm, and perceived behavioral control constructs of the TPB. Johnson et al. (1999) find that product usefulness represents a belief that may be a factor of perceived usefulness for OOSD. Johnson et al. (1999), however, did not empirically test the relationship between product usefulness and attitude, developer intentions, or usage of OOSD.

Using the Triandis model of human behavior as a theoretical foundation, Khalifa and Verner (2000) surveyed 82 experienced software developers in order to explain developer usage of the waterfall and prototyping approaches. Khalifa and Verner (2000) propose that beliefs associated with product quality, conceptually similar to product usefulness, are associated with usage of the waterfall and prototyping approaches. Khalifa and Verner (2000) measure product quality using the following formative indicators: the extent to which the methodology improves software quality and the extent to which the methodology improves software maintainability. Results from Khalifa and Verner (2000), however, suggest that product quality does not have a significant influence on waterfall or prototyping usage. Table 2.4 shows the t-statistics, standard error and coefficients for the effect of product quality on waterfall and prototyping usage from Khalifa and Verner (2000).

Table 2.4 Perceptions of Product Quality for Waterfall and Prototyping (Khalifa & Verner, 2000)

Approach	t-statistic	Standard Error	Coefficient
Waterfall	.05	.11	.05
Prototyping	.85	.12	.10

Although Khalifa and Verner (2000) find that product quality is not a significant explanatory variable for methodology usage by individual developers, Johnson et al. (1999) find that product usefulness is a dimension of perceived usefulness for OOSD. Furthermore, empirical research indicates that methodology use is positively related to increases in product quality (Harter et al., 2000; Herbsleb et al, 1994). These findings suggest that product usefulness should be a significant dimension of rational usefulness.

H2a: Product usefulness will be a significant component of rational usefulness.

Process usefulness

Another belief developers associate with a methodology is the extent to which it improves the productivity of the development process (Huisman & Iivari, 2006; Johnson et al., 1999; Khalifa & Verner, 2000). Methodologies play several rational roles related to increasing the productivity of the development process, including the facilitation of project management and control, the reduction of complexity, the facilitation of an appropriate division of labor and the systematization of development knowledge. Process usefulness captures beliefs related to these roles and is defined as the degree to which using a methodology improves the productivity of the systems development process. Process usefulness is conceptualized as a dimension of rational usefulness because increasing the productivity of the development process can help to reduce system delivery time (Harter et al., 2000); timely delivery of the system, in turn, is considered a rational objective (Franz & Robey, 1984).

Johnson et al. (1999) find that process usefulness represents an important belief about the perceived usefulness of OOSD. Johnson et al. (1999) find beliefs associated with process usefulness include the extent to which the methodology (OOSD) contributes to the development of more understandable analysis and design models, an easier and more natural modeling process, more effective analysis and design model reuse, a more flexible and adaptable development environment and an easier transition from Object-Oriented Analysis (OOA) to Object-Oriented Programming (OOP). Johnson et al. (1999), however, did not empirically test the relationship between process usefulness and attitude, developer intentions, or usage of OOSD.

Khalifa and Verner (2000) define process quality, conceptually similar to process usefulness, as the overall effect of the methodology on the productivity of the development process. Khalifa and Verner (2000) measure process quality using the following formative indicators: the extent to which the methodology improves communication with users, increases project control, aides in the early detection of problems and reduces development cost. Results from Khalifa and Verner (2000) suggest that process quality is a significant explanatory variable ($p<.01$) for waterfall and prototyping usage. Table 2.5 shows the t-statistics, standard error and coefficients for the effect of product quality on waterfall and prototyping usage.

Table 2.5 Perceptions of Process Quality for Waterfall and Prototyping (Khalifa & Verner, 2000)

Approach	t-statistic	Standard Error	Coefficient
Waterfall	4.04	.11	.46
Prototyping	1.74	.10	.24

Additional research finds that developers believe methodologies increase project control (Fitzgerald, 1998a). Methodologies also facilitate project management, help control the development process (Avison & Fitzgerald, 1995; Fitzgerald et al., 2002;

Russo & Stolterman, 2000), and provide an effective structure for organizing systems development activities (Fitzgerald et al., 2002). Moreover, empirical research finds a positive correlation between methodology use and faster development time (Harter et al., 2000; Herbsleb et al., 1994). Consistent with these findings, process usefulness should be a significant dimension of rational usefulness.

H2b: Process usefulness will be a significant component of rational usefulness.

Communication usefulness

Communication usefulness is defined as the degree to which using a methodology improves communication with users and other team members (Johnson et al., 1999). Communication usefulness is conceptualized as a dimension of rational usefulness because improving communication has the potential to result in more accurate requirements (Bostrom, 1989), a rational objective.

Although Khalifa and Verner (2000) included communication with users as a dimension of process usefulness, communication usefulness is conceptualized as a separate dimension of perceived usefulness, rather than a component of process usefulness for two reasons. First, Johnson et al. (1999) find via an exploratory factor analysis that communication usefulness is its own factor, apart from process usefulness. Second, Khalifa and Verner (2000) limit communication usefulness to communication with users and do not include communication with other team members. The communication usefulness dimension identified by Johnson et al. (1999), however, accounts for improved communication with other team members in addition to improved communication with users.

Johnson et al. (1999) find that communication usefulness is a factor of perceived usefulness for OOSD. Additionally, Khalifa and Verner (2000) find communication with users represents a significant factor affecting perceptions of the productivity of the development process. Finally, case study research finds developers believe following a standardized process can increase communication and coordination among various stakeholders on a software development project (Fitzgerald, 1998b; Fitzgerald et al., 2002). Overall, these findings suggest that communication usefulness should be a significant dimension of rational usefulness.

H2c: Communication usefulness will be a significant component of rational usefulness.

Political usefulness

Robey and Markus (1984) argue that the systems development process can also be viewed as a political process. A process must meet two requirements to be considered political—motive and opportunity (Robey & Markus, 1984). Motive refers to the existence of two or more individuals or groups having different objectives (Robey & Markus, 1984). Opportunity refers to a situation in which some individuals or groups may achieve their own objectives to the absolute or relative disadvantage of others (Robey & Markus, 1984). The systems development process satisfies these two requirements because it assembles numerous stakeholders with various objectives, thereby creating opportunities for some stakeholders to advance personal agendas to the disadvantage of other stakeholders (Robey & Markus, 1984).

A methodology can be viewed as a political process because it can be used to achieve objectives particular to one group or individual to the relative disadvantage of other stakeholders (Robey & Markus, 1984). As shown in Table 2.6, additional research corroborates this perspective and suggests that methodologies play several roles related to the perception of a methodology as a political process.

Table 2.6 Political roles of methodologies

Political Role	Description	Source
Professionalize ISD work	Methodology is used to show upper-level management and the client that systematic development practices are being used.	Fitzgerald, 1998b; Fitzgerald et al., 2002; Kautz et al., 2004; Nandhakumar & Avison, 1999; Sauer & Lau, 1997
Comfort/confidence factor	Methodology is used to justify and support expensive investment decisions in development. By showing management that a process exists, the methodology could be used to generate more confidence that the expenditure on the development project was justified.	Fitzgerald, 1998b; Fitzgerald et al., 2002; Sauer & Lau, 1997
Audit trail	Methodology insulates developers from unreasonable user demands and shows the rationale for all design decisions taken at each stage of development in case decisions turn out to be wrong.	Fitzgerald, 1998b; Fitzgerald et al., 2002; Kautz et al., 2004
Legitimacy factor	Methodology is used to win contracts or achieve CMMi certification. Methodology is used as a marketing tool to gain additional business.	Fitzgerald, 1998b; Fitzgerald et al., 2002
Reduce anxiety	The methodology is used to reduce developer anxiety.	Kautz et al., 2004; Wastell, 1996; 1999

Methodologies are used as a legitimacy factor in that they help software development organizations meet a certification set by an external standards-setting organization (e.g., CMMi) or to satisfy a client requirement (Duggan, 2004; Fitzgerald, 1998b; Fitzgerald, et al., 2002). Similarly, methodologies serve as marketing ploys to impress clients and help software vendors and consulting firms compete for software development contracts (Fitzgerald, 1998b; Fitzgerald, et al., 2002). Methodologies can also act as a confidence factor for upper-level management as well by demonstrating that the development process is conducted in a systematic fashion (Fitzgerald, 1998b; Fitzgerald et al., 2002; Nandhakumar & Avison, 1999; Sauer & Lau, 1997). Management spends large amounts of money on new systems development efforts; these systems development efforts need as much justification as possible. To build the business case for a systems development effort, methodologies are often pitched as ways to increase management confidence that the systems development effort will succeed.

When developers are confronted with the challenge of building a large and complex information system, using a methodology can provide them with the confidence that they are capable of developing the information system (Kautz et al., 2004). Used in this manner, a methodology can help reduce developer stress and anxiety resulting from systems development (Wastell, 1996).

Methodologies also provide an audit trail of decisions made during the development process (Fitzgerald, 1998b; Fitzgerald et al., 2002; Kautz et al., 2004). By providing the rationale for decisions made during systems development, methodologies can provide cover in case design decisions turn out to be wrong (Fitzgerald, 1998b; Fitzgerald et al., 2002; Wastell, 1999). Using a methodology also defends against complaints that a valid process was not followed (Fitzgerald, 1998b, Fitzgerald et al., 2002) and helps ameliorate a situation where users who “shouted the loudest” normally decided development priorities (Fitzgerald et al., 2002; Kautz et al., 2004; Sauer & Lau, 1997).

The perception of a methodology as a political process coupled with research indicating that a methodology can play several political roles suggests that it may be instrumental in achieving valuable outcomes related to the political perspective. For example, a methodology may be useful for achieving an objective particular to one individual or group, such as justifying decisions made during systems development, rather than for achieving a rational objective, such as improving the productivity of the development process. Political usefulness is defined as the degree to which using a methodology results in valuable outcomes that are particular to one group or individual and may result in the relative disadvantage of other stakeholders on the systems development project. Based on research suggesting that a methodology may be instrumental in achieving valuable outcomes related to the political perspective (Fitzgerald 1998b; Fitzgerald et al., 2002; Kautz et al., 2004; Nandhakumar & Avison, 1999, Robey & Markus, 1984; Sauer & Lau, 1997; Wastell, 1996; 1999), political usefulness should be a significant dimension of perceived usefulness.

H3: Political usefulness will be a significant component of perceived usefulness.

Dimensions of political usefulness

Decomposing political usefulness into its proposed referent dimensions should afford a deeper understanding of what makes a methodology useful to developers. A review of the research (Fitzgerald, 1998b; Fitzgerald et al., 2002; Johnson et al., 1999; Kautz et al., 2004; Nandhakumar & Avison, 1999; Sauer & Lau, 1997; Wastell, 1996; 1999) suggests that political usefulness has four dimensions: career usefulness, symbolic usefulness, support usefulness, and defense usefulness.

Similar to rational usefulness and its proposed referent dimensions, political usefulness is conceptualized as a second-order molar construct composed of four first-order reflective constructs—career usefulness, symbolic usefulness, support usefulness and defense usefulness. Applying the mental experiment recommended by Bollen (1989) to political usefulness and its proposed dimensions, a change in political usefulness will not necessarily result in a change in all dimensions of political usefulness (Bollen, 1989). On the contrary, a change in one dimension of political usefulness will result in a change in political usefulness (Bollen, 1989). These arguments suggest that political usefulness is a second-order molar, rather than molecular, construct (Bagozzi, 1985; Bollen, 1989; Chin & Gopal, 1995) formed by career usefulness, symbolic usefulness, support usefulness, and defense usefulness. Each proposed dimension of political usefulness is a first-order reflective construct composed of multiple items. Table 2.7 summarizes the dimensions of political usefulness.

Table 2.7 Political Usefulness and its Dimensions

Dimension	Definition	Source
Political Usefulness	The degree to which using a methodology results in valuable outcomes that are particular to one group or individual and may result in the relative disadvantage of other stakeholders on the systems development project.	Robey & Markus, 1984
Career Usefulness	The degree to which using a methodology improves marketability, increases job flexibility, raises chances for promotion, and offers opportunity for more meaningful work.	Hardgrave et al., 2003; Johnson et al., 1999; Thompson et al., 1991
Symbolic Usefulness	The degree to which using a methodology shows others that professional development processes are being used.	Fitzgerald, 1998b; Fitzgerald et al., 2002; Kautz et al., 2004; Nandhakumar & Avison, 1999; Robey & Markus, 1984
Support Usefulness	The degree to which using a methodology reduces developer anxiety.	Fitzgerald, 1998b; Fitzgerald et al., 2002; Kautz et al., 2004; Wastell ,1996; Wastell, 1999
Defense Usefulness	The degree to which using a methodology provides protection in case decisions made during systems development turn out to be wrong and protects against unreasonable user demands.	Fitzgerald, 1998b; Fitzgerald et al., 2002; Robey & Markus, 1984

Career usefulness

Career usefulness is defined as the extent to which using a methodology improves marketability, increases job flexibility, raises chances for promotion, and offers opportunity for more meaningful work (Johnson et al., 1999; Thompson, et al., 1991). Career usefulness is conceptualized as a dimension of political usefulness because the motive behind career usefulness relates to using a methodology for career advancement rather than to achieve rational objectives (Robey & Markus, 1984). Using a methodology for career benefits may also result in a relative disadvantage to other stakeholders. For example, the quest for career advancement could motivate developers to use the Rational Unified Process (RUP), rather than an in-house methodology. Thus, developers may recommend using the RUP because learning the RUP is valuable for career advancement, even though it may not result in the most efficient development process (Robey & Markus, 1984).

Prior research on technology adoption has identified a conceptually similar construct to career usefulness: career consequences (Thompson et al., 1991). In the context of methodologies, career consequences is defined as the degree to which the methodology increases the flexibility to change jobs and the opportunity for more meaningful work (Hardgrave et al., 2003; Thompson et al., 1991). Although prior research has conceptualized career consequences as a distinct construct, recent research has included career usefulness as a component of perceived usefulness (Johnson et al., 1999; Venkatesh et al, 2003). Consistent with Johnson et al. (1999) and Venkatesh et al. (2003), career usefulness is conceptualized as a dimension of political usefulness.

Research provides conflicting evidence regarding the effect of career usefulness on developer attitude toward a methodology. Riemenschneider et al. (2002) find career consequences, conceptually similar to career usefulness, does not significantly influence developer intention to use a formalized in-house methodology. On the other hand, Johnson et al. (1999) find career usefulness is a factor of perceived usefulness for a formalized commercial methodology (OOSD).

A closer inspection of the Riemenschneider et al. (2002) study reveals that the methodology investigated was a formalized *in-house* methodology, rather than a formalized *commercial* methodology. Skills acquired from using a formalized in-house methodology may not be easily transferable in the marketplace, thereby diminishing their marketability. On the contrary, formalized commercial methodologies may provide more marketable skills than formalized in-house methodologies. Developers who learn to use a formalized in-house methodology may be less able to transfer those skills to another organization than developers who learn to use a formalized commercial methodology because the formalized commercial methodology is used across many organizations. Additionally, skill acquisition represents a key driver of motivation, satisfaction, and turnover for Information Technology professionals (Ferratt & Short, 1988; Igbaria, Parasuraman & Badawy, 1994). Taken as a whole, these findings suggest that career usefulness should be a significant dimension of political usefulness.

H3a: Career usefulness will be a significant component of political usefulness.

Symbolic usefulness

Robey and Markus (1984) propose that practices recommended by methodologies, such as requirements approval by users, symbolize rationality and signify that the actions taken during systems development are systematic. Case study research by Fitzgerald (1998b) corroborates this perspective and finds that methodologies play several political roles such as professionalizing ISD work, as well as serving as comfort/confidence and legitimacy factors.

Fitzgerald (1997; 1998b) interviewed eight developers and eight project managers in eight different organizations (see Table 2.8 for background on the organizations). Results suggest methodologies are used to assure upper-level management and or clients that professional development processes are being used, thereby providing confidence that development decisions have been made on a systematic basis. As one interviewee stated, methodologies provide “a statement to the effect that we’ve got standards to our work, just like other departments.” (Fitzgerald, 1998b; Fitzgerald et al., 2002).

Nandhakumar & Avison (1999) conducted a case study on methodology use in a large multinational manufacturing company. Findings indicate that methodologies serve as a necessary “fiction” to present an image of control. The methodology is used primarily to give the impression that systematic standards and guidelines are being used. Similarly, Kautz et al. (2004) interviewed four developers, two software architects, and four project managers. Results suggest a methodology is used to show upper-level management that a systematic methodology is being used.

Overall, these findings suggest methodologies are useful from a symbolic standpoint in that they are used as a mark of professionalism and to show others that systematic development processes are being used. Symbolic usefulness captures these beliefs and is defined as the degree to which using a methodology shows others that professional development processes are being used. Thus, the methodology functions as a vehicle for maintaining the appearance of rationality in systems development, regardless of whether or not it actually achieves rational objectives (Robey & Markus, 1984). Robey and Markus (1984) classify these actions as political. Using a methodology for its symbolic benefits may result in a relative disadvantage to other stakeholders. For example, when used to simply show others that systematic development processes are being used, the methodology tends to be used in a superficial manner (Fitzgerald et al., 2002), thereby possibly hindering the full potential of the methodology to achieve rational objectives. Findings regarding how methodologies are used to portray control and professionalism (Fitzgerald, 1998b; Fitzgerald et al., 2002; Kautz et al., 2004; Nandhakumar & Avison, 1999) suggest that symbolic usefulness should be a significant dimension of political usefulness.

H3b: Symbolic usefulness will be a significant component of political usefulness.

Table 2.8 Organizational Background Information for study by Fitzgerald (1997)
Copyright © 1997 by Blackwell. Used by permission.

Organization	Business Description	Number of Employees	Number in Information Systems Department	Number of developers on a typical project	Number of months for a typical project	Methodology recommended in company
Allied Irish Banks (AIB)	Financial Services	16,000	600	10-12	20	Information Engineering
Department of Education	Government Department	950	15	4	6	SSADM
Core Computing Ltd.	Software House	52	43	5	3-6	Internal based on Oracle*Case & SSADM
Naval Service: Department of Defense	Government Department	1200	10	4-6	6-15	Internal based on Oracle*Case & SSADM
Dairygold Co-operative Society Ltd.	Food Co-operative	2500	20	1-2	1-3	Internal not based on commercial System Development Methodology (SDM)
Bord Gais Eireann (BGE) (National Gas Co.)	Energy Service Provider	800	30	5	30	Not using a formalized SDM
Trustee Savings Bank (TSB)	Financial Services	1200	40	2	1.5	Not using a formalized SDM
Pfizer Pharmaceuticals	Bulk Manufacturing	250	6	2	3	Not using a formalized SDM

Support usefulness

Systems development is a stressful process imposing considerable pressure on developers, thereby raising their anxiety (Wastell & Newman, 1993). The stressfulness of systems development originates from its high complexity (Wastell, 1999). Methodology use is one way in which developers can deal with the complexity, stress, and anxiety of software development because it transfers some of the complexity of software development from the developer to the methodology (Lee & Truex, 2000). Case study research reaffirms these arguments. Wastell (1996) conducted a case study on an organization moving to the Structured Systems Analysis and Design Methodology (SSADM). Results suggest that methodologies operate as more than rational tools to improve productivity, they also provide psychological support (Wastell, 1999). Used in this fashion, the methodology serves as a security blanket for developers (Wastell, 1999). Additional case study research has corroborated the findings of Wastell (1996; 1999) and finds methodologies provide psychological support, especially to inexperienced developers (Kautz et al., 2004).

These findings suggest methodologies may be useful because they provide psychological support and contain developer anxiety. Support usefulness summarizes these ideas and is defined as the degree to which using a methodology alleviates developer anxiety. Support usefulness is conceptualized as a dimension of political usefulness because the motive behind support usefulness relates to using a methodology to reduce anxiety rather than to achieve rational objectives. Using a methodology to alleviate anxiety may also result in a relative disadvantage to other stakeholders. For example, using a methodology to ease anxiety may result in goal displacement--a phenomenon wherein the developer becomes so absorbed with following the methodology that he or she loses sight of the real goal of systems development, i.e., developing the system (Fitzgerald et al., 2002; Wastell, 1996). Goal displacement may result in developers performing unnecessary steps during the development process (Fitzgerald et al., 2002), thereby potentially reducing productivity. Findings regarding how developers use methodologies to relieve anxiety (Kautz et al., 20004; Wastell, 1996, 1999) suggest that support usefulness should be a significant dimension of political usefulness.

H3c: Support usefulness will be a significant component of political usefulness.

An important covariate for support usefulness is systems development self-efficacy. Self-efficacy refers to a person's belief in his or her ability to accomplish a specific task (Bandura, 1977; 1986; 1997; Compeau & Higgins, 1995a). Self-efficacy is not concerned with the skills one has, but judgments of whatever one can do with the skills one possesses (Bandura, 1986). Self efficacy has been applied in a variety of settings including career choice and development (Betz & Hackett, 1981; Jones, 1986), academic achievement (Pajares 2002), sales performance (Barling & Beattie, 1983) and physical exercise (Marcus, Selby, Niaura, & Rossi, 1992). Additionally, the Information Systems literature has explored the impact of self-efficacy on information technology usage (Agarwal, Sambamurthy & Stair, 2000; Compeau & Higgins, 1995a; Compeau & Higgins, 1995b; Thompson et al., 2006; Venkatesh & Davis, 1996;). Compeau and Higgins (1995a) developed the concept of computer self-efficacy and define the construct as a judgment of one's ability to use an information technology.

Prior research suggests that development process knowledge—understanding how to appropriately apply methodologies and technologies to develop an information system—constitutes an important component of developer knowledge (Backlund, Hallenborg & Hallgrimsson, 2003; Vitalari, 1985). Extrapolating Bandura's definition of self-efficacy (1977; 1982; 1986; 1997) to the domain of systems development, systems development self-efficacy can be defined as a developer's judgment regarding his or her ability to apply the appropriate techniques, documentation aids, procedures, and tools to successfully build an information system.

Most studies show that experienced developers are less likely than inexperienced developers to use methodologies (Fitzgerald, 1997; Huisman & Iivari, 2002; Kautz et al., 2004; Kozar, 1989; Lee & Kim, 1992; Nandhakumar & Avison, 1999). Experienced developers are likely to rely more on the skills they have obtained through experience and find methodologies inhibiting (Fitzgerald, 1997). Experienced developers describe methodologies as plans or guides to action, rather than as deterministic rule-sets to be followed rigorously (Beynon-Davies & Williams, 2003; Fitzgerald, 1997; Kautz et al., 2004; Nandhakumar & Avison, 1999). Furthermore, experienced developers rely upon their development process knowledge and use methodologies in a toolbox fashion rather than following methodologies in a step-by-step fashion (Kautz et al., 2004).

On the other hand, inexperienced developers tend to follow methodologies in a step-by-step manner (Fitzgerald, 1997). Inexperienced developers find methodologies to be useful guides and express a need for explicated methodologies to help them learn the company's development practices (Kautz et al., 2004).

While research finds that development experience is negatively correlated with methodology use (Fitzgerald, 1997; Huisman & Iivari, 2002; Kautz et al., 2004; Kozar, 1989; Lee & Kim, 1992), other research suggests that systems development self-efficacy judgments, rather than experience, may change the pattern of methodology use (Wastell, 1999). Wastell (1999) conceptualizes methodologies as transitional objects. The transitional object is an important component in the development of self reliance. The teddy bear is a classic example; it serves to separate the child from its mother by acting as a supportive figure (Wastell, 1999). By using the teddy bear, the child gains self confidence and ultimately becomes more independent (Wastell, 1999).

Similarly, developers with low self-efficacy may use methodologies as transitional objects (Wastell, 1999). As developers gain more experience developing systems, they acquire greater systems development self-efficacy (Agarwal et al., 2000; Marakas, Yi & Johnson, 1998) and reduce their anxiety (Compeau & Higgins, 1995b). As a result, developers begin to use methodologies in a pragmatic, rather than in a cookbook fashion (Fitzgerald, 1997). Inexperienced developers use methodologies to alleviate their anxiety during systems development (Wastell, 1996) and to ease their lack of self-efficacy (Fitzgerald, 1997).

These findings suggest that developers with low systems development self-efficacy may be more prone to use a methodology for its support usefulness (Wastell, 1999). Developers with low systems development self efficacy may tend to use methodologies more for psychological support and to alleviate anxiety than developers with high systems development self-efficacy. Consequently, systems development self-efficacy was included as a covariate to account for possible differences in perceptions of support usefulness resulting from systems development self-efficacy judgments.

Defense usefulness

Prior research indicates that information technologies are used defensively to cover one's position. For example, Markus (1994) investigated email use within organizations and finds employees use email defensively to cover themselves. By using email rather than face-to-face communication, employees are able to construct an audit trail of all communication that they can later use to hold other employees accountable or to cover their own position.

Similar to email, the systems development process creates the opportunity for developers to justify decisions made during systems development and escape responsibility for poor performance (Robey & Markus, 1984). Methodologies represent authority in that they not only outline how to conduct systems development, but also that systems development should be done a certain way (Wastell, 1999). Thus, developers can draw upon the methodology to authorize their actions and attempt to absolve themselves of personal responsibility for their decisions (Wastell, 1999).

Additional case study research on methodology use confirms the arguments of Wastell (1999) and finds developers use a methodology to justify decisions made during systems development. Fitzgerald (1997; 1998b) interviewed eight developers and eight project managers in 8 different organizations on how they use methodologies; Table 2.8 provides a background on the organizations. Fitzgerald (1998b) finds that methodologies allow developers to create an audit trail of the development process. By documenting all the steps in the development process and the rationale behind development decisions, developers can use methodologies to provide some protection in case decisions made during systems development turn out to be wrong (Fitzgerald 1998b; Fitzgerald et al., 2002). Furthermore, using a methodology also defends against complaints from upper-level management that a systematic process was not followed (Fitzgerald, 1998b; Fitzgerald et al., 2002). Finally, methodologies can insulate developers from unreasonable user deadlines and help resolve a situation where users dictated development priorities (Fitzgerald et al., 2002; Kautz et al., 2004).

Overall, these findings suggest that methodologies can be useful to developers from a defensive standpoint. Defense usefulness captures this idea and is defined as the degree to which using a methodology provides protection in case decisions made during systems development turn out to be wrong and protects against unreasonable user demands. Defense usefulness is conceptualized as a dimension of political usefulness because the motive behind defense usefulness relates to using a methodology to provide protection from criticism and unreasonable user demands rather than using a methodology to achieve rational goals. Using a methodology for its defensive benefits may also result in a relative disadvantage to other stakeholders. For example, developers may use a methodology to try to escape personal responsibility for their actions (Robey & Markus, 1984; Wastell, 1999) or to block requirements changes (Wastell, 1996). Findings on how developers use methodologies for their defensive benefits (Fitzgerald, 1998b; Fitzgerald et al., 2002; Kautz et al., 2004; Wastell, 1999) suggest that defense usefulness should be a significant dimension of political usefulness.

H3d: Defense usefulness will be a significant component of political usefulness.

Perceived ease of use

Perceived ease of use refers to the degree to which an innovation is easy to understand and use and is comparable to the complexity construct in DIT (Davis, 1989). In the context of methodologies, perceived ease of use refers to the degree to which developers perceive a methodology as difficult to use (Hardgrave et al., 2003). Although prior research investigating developer intention to use a methodology finds that complexity (perceived ease of use) is not a significant predictor of developer intention to use a methodology (Riemenschneider et al., 2002), the DTPB indicates that perceived ease of use is positively associated with attitude (Taylor & Todd, 1995).

Consistent with research on tool adoption (Mathieson, 1991; Taylor & Todd, 1995; Venkatesh & Davis, 2000), prior research on developer intention to use a methodology finds a significant negative (positive) effect for complexity (perceived ease of use) on perceived usefulness (Hardgrave et al., 2003). Assuming that perceived ease of use positively affects attitude (Taylor & Todd, 1995) and that complexity (perceived ease of use) negatively (positively) affects perceived usefulness (Hardgrave et al., 2003), perceived ease of use should have a positive influence on both attitude toward using a formalized commercial methodology and perceived usefulness.

H4a: Higher levels of perceived ease of use of a formalized commercial methodology will lead to higher levels of positive developer attitude toward using a formalized commercial methodology.

H4b: Higher levels of perceived ease of use of a formalized commercial methodology will lead to higher levels of perceived usefulness of a formalized commercial methodology.

Compatibility

Compatibility refers to the degree to which an innovation is consistent with existing values, past experiences, and needs of potential adopters (Rogers, 1983, 1995). In the context of methodologies, compatibility is defined as the degree to which a methodology is consistent with existing software development practices (Hardgrave et al., 2003).

Prior research has generated conflicting results regarding the importance of compatibility. While the DTPB posits that compatibility positively affects attitude, research on tool adoption finds that the direct effects of facilitating conditions—compatibility is related to facilitating conditions—on intention to use become non-significant when performance expectancy (perceived usefulness) and effort expectancy (perceived ease of use) are included in the theoretical model (Venkatesh et al, 2003).

Although similar to the adoption of software development tools and techniques, the adoption of methodologies constitutes a more radical change (Orlikowski, 1993; Roberts et al., 1998). Thus, compatibility is more important in methodology adoption than tool adoption due to the greater magnitude of change required by a methodology (Hardgrave et al., 2003; Riemenschneider et al., 2002). Consistent with these arguments, Riemenschneider et al. (2002) and Hardgrave et al. (2003) find that compatibility has a significant positive influence on developer intention to use a methodology. Prior research also finds that compatibility has a significant positive influence on perceived usefulness (Chau & Hu, 2001; Hardgrave et al., 2003; Oh et al., 2003; Templeton & Byrd, 2003). These findings suggest compatibility should have a positive influence on both developer attitude toward using a formalized commercial methodology and perceived usefulness.

H5a: Higher levels of compatibility with existing software development processes will lead to higher levels of positive developer attitude toward using a formalized commercial methodology.

H5b: Higher levels of compatibility with existing software development processes will lead to higher levels of perceived usefulness of a formalized commercial methodology.

Chapter Three

Methodology

This study seeks to explain developer attitude toward using a formalized commercial methodology. Given the importance of perceived usefulness in explaining intention to use methodologies (Hardgrave et al., 2003), this study decomposed the perceived usefulness construct into its referent dimensions in order to understand the construct on a deeper level.

The first section of this chapter presents the research hypotheses, followed by the research design. The second section discusses instrument development as well as the research instrument, and the third and final section addresses data analysis, the sample and threats to internal validity.

Research Model and Hypotheses

The following is a list of detailed hypotheses that have been created after a thorough review of the literature (see Chapter Two). These hypotheses will be tested in this research to empirically validate the proposed research model presented in Figure 3.1.

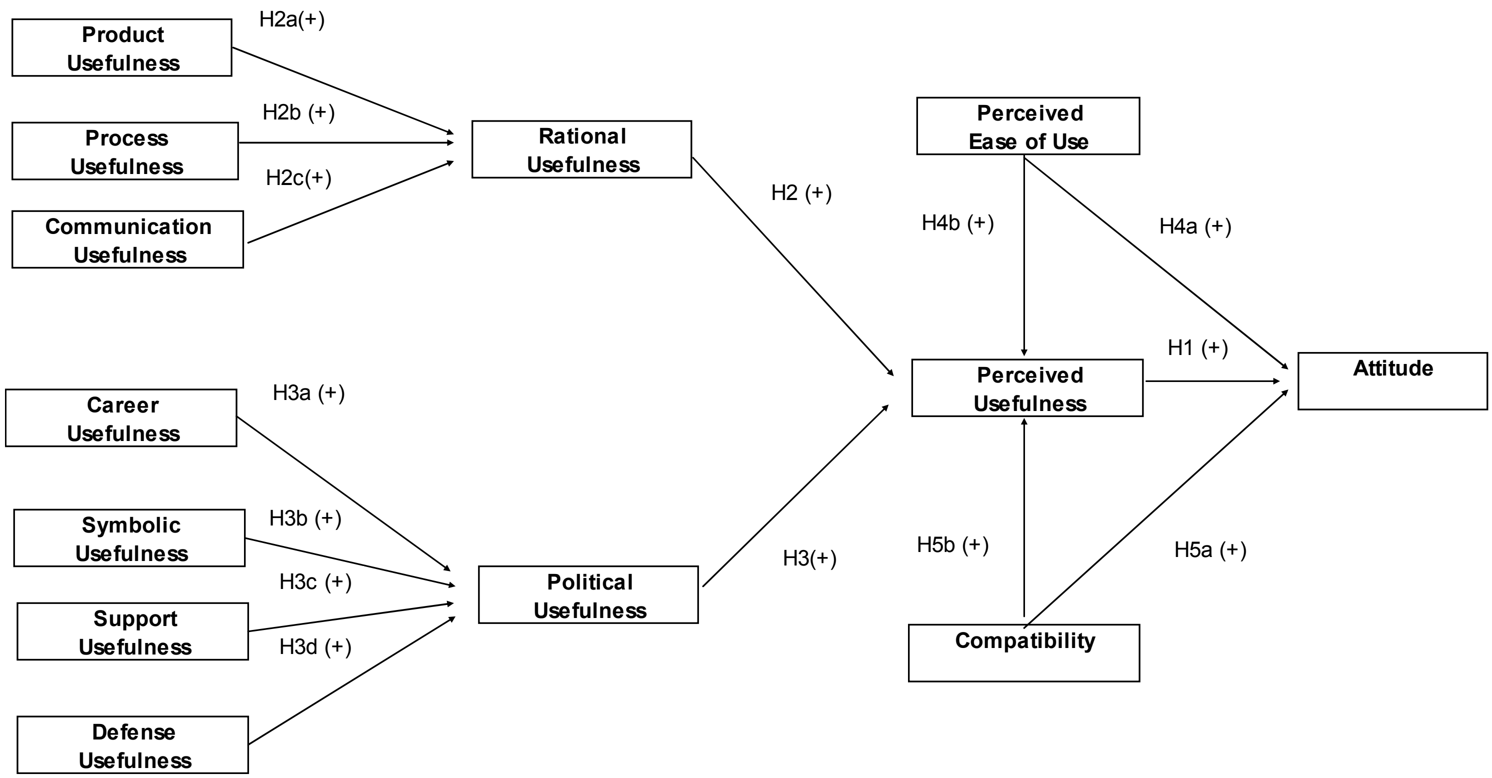

Figure 3.1 Research Model

Hypothesis 1

- H1: Higher levels of perceived usefulness of a formalized commercial methodology will lead to higher levels of positive developer attitude toward using a formalized commercial methodology.

Hypothesis 2

- H2: Rational usefulness will be a significant component of perceived usefulness.
- H2a: Product usefulness will be a significant component of rational usefulness.
- H2b: Process usefulness will be a significant component of rational usefulness.
- H2c: Communication usefulness will be a significant component of rational usefulness.

Hypothesis 3

- H3: Political usefulness will be a significant component of perceived usefulness.
- H3a: Career usefulness will be a significant component of political usefulness.
- H3b: Symbolic usefulness will be a significant component of political usefulness.
- H3c: Support usefulness will be a significant component of political usefulness.
- H3d: Defense usefulness will be a significant component of political usefulness.

Hypothesis 4

- H4a: Higher levels of perceived ease of use of a formalized commercial methodology will lead to higher levels of positive developer attitude toward using a formalized commercial methodology.
- H4b: Higher levels of perceived ease of use of a formalized commercial methodology will lead to higher levels of perceived usefulness of a formalized commercial methodology.

Hypothesis 5

- H5a: Higher levels of compatibility with existing software development processes will lead to higher levels of positive developer attitude toward using a formalized commercial methodology.
- H5b: Higher levels of compatibility with existing software development processes will lead to higher levels of perceived usefulness of a formalized commercial methodology.

Research Design

This study uses a survey research methodology for several reasons. First, prior research using the Decomposed Theory of Planned Behavior (DTPB) has used surveys to gather data (Hung & Chang, 2005; Pederson, 2005; Shih, 2004; Taylor & Todd, 1995; Thompson et al., 2006). Second, because surveys are often used to investigate subject opinions (Whitley, 2002), they represent a viable approach for investigating developer attitude toward methodologies. Third, prior research on the political uses of methodologies has used a case study research strategy (Fitzgerald, 1998b; Fitzgerald et al., 2002; Wastell, 1996; 1999). Although case studies allow the researcher to study participants in a natural setting (Whitley, 2002) and allow the researcher to study phenomena in great depth (Eisenhardt, 1989), findings from case study research suffer from low external validity and may be subject to researcher bias (Whitley, 2002). Surveys represent a viable research strategy when the researcher wants to investigate subject opinions (Whitley, 2002) and when the researcher wants to determine whether a relationship holds across a number of cases, not just one case (Whitley, 2002). Thus, by using a survey, this study will investigate the perceptions of a wider sample of software developers regarding methodologies.

Instrument Development and Validation

Straub (1989) and Boudreau, Gefen and Straub (2001) emphasize the importance of instrument validation. Accordingly, this research follows the recommendations of Straub (1989), Churchill (1979) and Netemeyer, Bearden and Sharma (2003) for construct definition and item development. Table 3.1 details the instrument development and validation process.

Phase I: Construct Definition

Netemeyer et al. (2003) recommend that researchers provide clear definitions for each new construct and advocate that construct definition and conceptualization be grounded in a thorough literature review. Following these recommendations, a thorough literature review (see Chapter Two) was conducted of the methodology adoption and methodology usage literature to uncover possible dimensions of perceived usefulness for a methodology. Prior empirical research on methodology adoption (Johnson et al., 1999; Khalifa & Verner, 2000) as well as case study research on methodology usage (Fitzgerald, 1998b; Fitzgerald et al., 2002) provided definitions for product, process, communication, and career usefulness. Case study research on methodology usage (Fitzgerald, 1998b; Fitzgerald et al., 2002; Johnson et al., 1999; Kautz et al., 2004; Nandhakumar & Avison, 1999; Sauer & Lau, 1997; Wastell, 1996; 1999) was reviewed to provide construct definitions for symbolic, support, and defense usefulness. The construct definition for systems development self-efficacy was adapted from Bandura's (1977; 1982; 1986; 1997) definition of self-efficacy. A complete list of the dimensions of perceived usefulness and associated definitions are provided in Tables 2.3 and 2.7 in Chapter Two.

Netemeyer et al. (2003) also recommend the use of experts and individuals from relevant populations to help narrow construct definitions. Following these recommendations, five informal interviews with developers were conducted to evaluate the face validity of the dimensions of perceived usefulness. Results from these discussions indicate that the construct definitions are understandable and relevant.

Table 3.1 Instrument Development and Validation Process

Phase	Name	Description of Phase	Validation Tests Performed
I	Construct Definition	Defined each construct based upon literature review.	• Qualitative validation of the constructs
II	Development of initial item pool	Developed initial item pool.	• Qualitative validation of the items
III	Card Sorting	Nine developers and six graduate Information Systems students performed a card sort on the initial item pool to provide insight into convergent and discriminate validity. Items were also checked for content validity and item wording.	• Item hit ratio • Item-level analysis • Card placement percentage
IV	Pretest	Items were pre-tested by an additional five developers.	• Qualitative validation of the items
V	Pilot Study	Pilot study was conducted on 100 developers.	• Cronbach's alpha • Confirmatory factor analysis • Item-total correlations

Phase II: Initial Item Development

Netemeyer et al. (2003) recommend using prior literature to generate the initial item pool. Following these recommendations, a thorough literature review was conducted (see Chapter Two). Items were drawn from existing scales as much as possible (Straub, 1989). Where items already existed for some scales, additional items were added to increase the potential of the new scales.

Netemeyer et al. (2003) and Churchill (1979) recommend that researchers include items with slightly different meaning. Netemeyer et al. (2003) further maintain that it is better to be over-inclusive of the construct's domain rather than under-inclusive when generating an item pool. Accordingly, Netemeyer et al. (2003) recommend a large pool of initial items. DeVellis (1991) recommends that for narrowly defined constructs, a pool that is twice the size of the final scale will suffice. Thus, the initial pool consisted of at **least eight items** for each dimension of perceived usefulness.

When writing items, Netemeyer et al. (2003) advocate using language common to the target population. Netemeyer et al. (2003) also suggest keeping items as short as possible. Following these recommendations, care was taken to ensure that the items used language common to software developers. Items were also kept as short as possible. Five Information Systems academics checked the initial item pool for content validity and wording. Results indicated that the items in the initial item pool were understandable and reflect the constructs in the proposed research model.

Phase III: Card Sorting

The goal of the card sorting phase was to provide insight into the validity of the scales for each dimension of perceived usefulness and to identify ambiguous or poorly worded items. To achieve these goals, an additional nine developers and six graduate Information Systems students sorted the various items into categories. Each item was printed on one 3x5 index card (Moore & Benbasat, 1991). The cards were then shuffled into random order for the presentation to the judges[5] (Moore & Benbasat, 1991). Each judge sorted the cards into categories and labeled the categories of items independently (Moore & Benbasat, 1991). In all four rounds of card sorting, a different set of judges was used (Moore & Benbasat, 1991). The judges were all developers in order to mirror the target audience of the final survey (Moore & Benbasat, 1991).

Card Sorting Procedures

Prior to sorting the cards, judges were read a standard set of instructions. Judges were allowed to ask as many questions as necessary to ensure they understood the procedure (Moore & Benbasat, 1991). Following the instructions, a trial sort was

[5] To remain consistent with the terminology used in Moore and Benbasat (1991), participants in the card sorting exercise are referred to as “judges”.

conducted with the judges on nine sample items unrelated to the constructs of the study. For the trial sort, nine statements were written about the various aspects of an automobile (Moore & Benbasat, 1991). Any misunderstandings resulting from the instructions were clarified after the trial sort.

During the sorting procedure, judges could place a given item in more than one category (Wing & Nelson, 1972) and there was no upper or lower limit for the number of categories that could be created by each judge (Wallace, Keil, & Rai, 2004). Prior research has shown that allowing judges to sort a given item into only one category or allowing judges to sort items into more than one category yields similar solutions (Wing & Nelson, 1972). After sorting the cards into piles, judges were asked to provide their own labels and definitions for each category (Moore & Benbasat, 1991). Judges were given blank cards for which to write their category labels.

To ensure that the categories developed by the judges matched the dimensions of perceived usefulness, judges were then asked to map their own categories into the seven dimensions of perceived usefulness. Judges were told that they could map their categories into one or more of the dimensions of perceived usefulness. Furthermore, participants were told that they did not have to map their category into a dimension of perceived usefulness if they thought that it did not fit with any of the dimensions of perceived usefulness. If the definitions provided by the judges matched the intent of the target dimension of perceived usefulness, then confidence in the construct validity of that dimension was higher (Moore & Benbasat, 1991).

Three measurements were calculated to evaluate the results of the card sorting experiments. First, an item placement ratio was calculated in each of the three rounds of card sorting (Moore & Benbasat, 1991). The item placement ratio indicated the overall frequency with which all judges placed items within the target dimension of perceived usefulness (Moore & Benbasat, 1991). The higher the percentage of items placed in the target dimension of perceived usefulness, the higher the degree of agreement across the judges (Moore & Benbasat, 1991). Although there are no established guidelines for "high" levels of item placement, scales based on categories having a high degree of correct item placement should have a reasonable degree of construct validity (Moore & Benbasat, 1991).

Second, an item level analysis was conducted to evaluate the card sorting results from the final round of card sorting. The item level analysis shows how frequently items intended to measure each dimension were grouped together (Wallace et al., 2004). If an item had greater than 2/3 placement with other items, it was retained, otherwise it was either revised or removed (Wallace et al., 2004). This analysis helped verify that the items were being grouped in a manner consistent with the proposed dimensions of perceived usefulness, thereby indicating a high level of convergent and discriminant validity.

Third, a card placement percentage was also calculated for the card sorting results from the final round of card sorting. The card placement percentage was calculated using a spreadsheet from www.boxesandarrows (Lamantia, 2006). The card placement percentage worksheet helps to quickly determine the percentile distribution of placements for any card [6] in relation to another card (Lamantia, 2006). A higher card placement percentage implies that more participants placed a certain item in the target dimension of

[6] The term "card" is synonymous with the term "item".

perceived usefulness (Lamantia, 2006). Items not placed in 66% of the target construct were either modified or removed (Lamantia, 2006).

Results of First Round of Card Sorting

The first round of card sorting was conducted on two developers. The first developer created eight categories and the second developer created nine categories. The item placement ratio and the categories provided by the judges are shown in Tables 1 and 2 in Appendix A.

As shown by the categories provided by the judges in Appendix A, the judges were sorting the cards into "personal" and "team" categories. To mitigate this problem, items in product, process, communication, symbolic and defense usefulness were reworded to a generic level. (i.e., references to "my", "I", and "me" were removed). It was hoped that rephrasing the items to a generic level would also diminish social desirability bias. Items for support usefulness and career usefulness were kept at the "me/my" level for the second round of card sorting as they were considered to be individual level constructs.

Results of the Second Round of Card Sorting

The second round of card sorting was conducted on four developers. The judges created an average of nine categories, ranging from five to fourteen categories. The item placement ratio is shown in Appendix B. Items placed in the target construct by 75% of the judges (three out of four) were carried over to the final round of card sorting. Items not placed in the target construct by 75% of the judges were revised or removed from the item pool.

Several changes were made to the items as a result of the second round of card sorting. First, comments gathered from judges indicated that the large volume of items made it difficult to sort the cards by the underlying dimension. Consequently, it appeared that the judges, due to the large number of items, were sorting the cards based on key words.

Second, as shown in Table 1 in Appendix B, several items for the process usefulness loaded on numerous constructs, indicating a lack of discriminant validity (Moore & Benbasat, 1991). For example, as shown in Table 1 in Appendix B, process usefulness items were placed in product usefulness 29 times. These items were removed from the item pool. Furthermore, items in process usefulness relating to user requirements, systems maintenance, documentation of the system, cost of systems development or reducing the dependency of the team on one person were removed because they did not meet the 75% threshold for placement in the target construct.

Third, career usefulness and support usefulness were not showing discriminant validity, as illustrated in Table 1 in Appendix B. For example, as shown in Table 1 in Appendix B, career usefulness items were placed in support usefulness 10 times; support usefulness items were placed in career usefulness 16 times. One possible reason for this problem is that the items in both of these constructs were phrased at the individual developer level, causing judges to group items in these two categories together. Thus, the decision was made to reword items for support usefulness to a generic level. For example, “Using this methodology reduces my anxiety” was reworded to “Using this methodology reduces developer anxiety”. It was hoped that rephrasing the items for support usefulness to a generic level would also alleviate social desirability bias, as it seems reasonable to assume that many developers may be hesitant to admit that they are nervous during systems development. Items for career usefulness were retained at the individual level to remain consistent with prior research and because it seemed reasonable to assume social desirability bias would not be a concern for this construct.

The items for communication usefulness were initially divided into items assessing frequency of communication and items assessing quality of communication. For example, the item "Using this methodology improves communication with users" would have been composed of two items: "Using this methodology improves the frequency of communication with users" and "Using this methodology improves the quality of communication with users". After reviewing the categories provided by the judges (Table 2 in Appendix B), it appeared that some judges were segregating the communication usefulness construct into quality and frequency of communication; this distinction resulted in concerns over convergent validity. Thus, the decision was made to remove the frequency-quality distinction and rephrase the items for communication usefulness to a generic level. Measuring communication usefulness at a generic level, without the frequency-quality distinction, is consistent with prior research (Johnson et al., 1999; Khalifa & Verner, 2000).

As illustrated in Table 1 in Appendix B, the item placement ratios for symbolic, support and defense usefulness were low. Several new items were added for the third round of card sorting to increase the potential of these scales.

Results of Third Round of Card Sorting

The third round of card sorting was conducted on three developers and six graduate Information Systems students at a large southeastern university. The judges created an average of 8.5 categories, ranging from 6 to 11 categories. The item placement ratio and categories defined by the judges are shown in Appendix C. The item placement ratios for product, process, career, and support usefulness all increased from round two to round three, indicating stronger convergent and discriminant validity for these constructs (Moore & Benbasat, 1991). Although, the item placement ratios for communication, symbolic and defense usefulness declined slightly from round two to round three, the item placement ratios for these three constructs are still 65% or higher, indicating that judges placed the items for these two constructs into the target construct the majority of the time.

Items loading more strongly on constructs other than the target construct were removed from the item pool (Moore & Benbasat, 1991). Two additional rules were used to determine whether an item should be removed or revised. First, items with less than 66% placement with other items were revised or removed (Wallace et al., 2004). Second, items not placed in 66% of the categories the correct time were either modified or removed (Lamantia, 2006). Items failing to meet both of the two rules were removed from the item pool. If an item passed one rule, but not the other, the item was revised and given to three developers for evaluation. If a majority of the developers agreed with the revision, the item was revised, otherwise the item was removed.

An analysis of the categories provided by the judges for defense usefulness (Tables 2 and 3 in Appendix C) revealed that some judges were making a distinction between defending against unreasonable demands in general and user demands. As a result, two additional items were added to defense usefulness. These two items offered more specific versions of the term “user demands”.

Phase IV: Pre-Testing

Items should be judged by multiple judges from the population of interest. Netemeyer et al. (2003) recommend the use of five or more judges. Following these recommendations, an additional five developers, two doctoral students and two Information Systems professors pre-tested the survey. Participants in the pre-test were asked to evaluate the survey along the following lines: clarity and quality of instructions, clarity and quality of the items, time to complete the survey and general flow of the survey. Follow-up one-on-one meetings were conducted to clarify comments on item wording and survey instructions.

Results of the pre-test indicated the survey took approximately 35 minutes to complete. Additional changes were made to address concerns over item wording and to improve the quality of the instructions for the survey.

Phase V: Pilot Study

Straub (1989) recommends calculating reliability statistics and performing a confirmatory factor analysis on data from the pilot study. Floyd and Widaman (1995) recommend a sample size of 5 to 10 observations per parameter estimated. Applying this rule to the proposed research model, a sample size of at least 55, based on 11 parameters (seven dimensions of perceived usefulness, perceived ease of use, compatibility, attitude and self-efficacy) was needed. Accordingly, a pilot study was conducted on 98 Information Technology professionals. Measures of internal consistency were calculated, a confirmatory factor analysis was conducted, and item-total correlations were examined (Straub, 1989). Due to subject availability concerns, the pilot study was open to all Information Technology professionals who use any methodology (e.g., an in-house methodology, an Agile methodology, or the RUP). The means and standard deviations for each reflective scale were also inspected.

The pilot study was conducted from April 9th, 2007 until May 11th, 2007. The survey received 139 responses; 40 responses were removed because they were incomplete and one response was removed because it was a duplicate. Accordingly, the confirmatory factor analysis and reliability analysis was conducted on the 98 complete and non-duplicate responses. The following paragraphs detail the changes made to the instrument as a result of the pilot study. Appendix D contains the items used in the pilot study.

Two items were removed from product usefulness-- PROD4, PROD7-- due to low item loadings. After removing these two items, there were five items for product usefulness in the final version of the instrument. The remaining five items for product usefulness had high item loadings (>.707), loaded more strongly on product usefulness than any other construct and exhibited high reliability (>.7).

Three items were removed from process usefulness-- PROC1, PROC 3, PROC7-- due to low item loadings. After removing these items, process usefulness had five items in the final version of the instrument. The remaining five items for this construct had high item loadings (>.707), loaded more strongly on process usefulness than any other construct and exhibited high reliability (>.7).

All four items for communication usefulness exhibited high item loadings (>.707) and high reliability (>.7).

Two items from symbolic usefulness—SYMB6 and SYMB7--were dropped because they loaded higher on attitude than on symbolic usefulness. Additionally, many of the symbolic usefulness items had an unexpected negative correlation with attitude. The items exhibiting a negative correlation with attitude related to portraying a façade of professionalism, which suggested a negative connotation. For example, the item "Using this methodology gives the appearance to others that industry standards are being used." implies that developers may be faking use of the methodology to make it look like they are using the methodology. The meaning of symbolic usefulness does not relate to faking use of the methodology, but rather relates to showing to others that systematic development practices are being used. Thus, six items-- SYMB1, SYMB2, SYMB3, SYMB4, and SYMB8--were dropped because they were negatively correlated with attitude. Only one item in symbolic usefulness—SYMB5--exhibited a high item loading, loaded on symbolic usefulness more strongly than any other construct and had a positive correlation with attitude. As a result, four new items-- SYMB10, SYMB11, SYMB12 and SYMB13--were added to the final version of the instrument. These new items did not include the words "impression", or the clause "gives an appearance" that may have conveyed a negative connotation.

One item was removed from support usefulness—ANX7--due to low loading and because it loaded higher on attitude than support usefulness. Two items—ANX2 and ANX6--were dropped due to low item loadings and low reliability; one item—ANX4--was dropped due to low reliability. The remaining five items for this construct had high item loadings (>.707), loaded more strongly on support usefulness than any other construct and exhibited high reliability (>.7).

Several items were removed from career usefulness. One item—CAREER3--was dropped because it loaded more highly on attitude than on career usefulness and another item—CAREER8--was dropped because it loaded more strongly on process usefulness than career usefulness. Two items—CAREER1 and CAREER4--were dropped due to low item loadings. The remaining five items for this construct had high item loadings (>.707), loaded more strongly on career usefulness than any other construct and exhibited high reliability (>.7).

Several items were removed from defense usefulness. Four items--DEF1, DEF11, DEF2, DEF7--were dropped due to low item loadings. One item—DEF3--was dropped because it loaded more highly on symbolic usefulness than defense usefulness, while another item—DEF4--was dropped because it loaded more strongly on communication usefulness than defense usefulness. Furthermore, another item—DEF8--was dropped because it loaded more strongly on symbolic usefulness than defense usefulness. Although item loadings for the following items--DEF5, DEF10 and DEF 12-- were low, they were kept to ensure adequate content validity. DEF6, 9, 13 had high item loadings (> .707), loaded more strongly on defense usefulness than any other construct and exhibited strong reliability (>.7).

Several attitude items were removed because they contained individual references (e.g., contained the words “I”, “my”, or “me”) when many of the items operationalizing the dimensions of perceived usefulness are at the generic level. Thus, to remain consistent with the generic wording of the items for the dimensions of perceived usefulness, only attitude items at the generic level were kept for the final instrument. The items kept for the attitude construct--ATT1, ATT5, ATT7, ATT8, ATT10-- exhibited high loadings (>.707), loaded more strongly on attitude than any other construct and exhibited high reliability (>.7).

All items for the self-efficacy construct were kept. All items except SE12, SE14 had high item loadings (>.707) and loaded more strongly on self-efficacy than any other construct. Although the item loadings for SE12 and SE14 were below the .707 threshold, they were kept for content validity reasons.

One item was dropped from compatibility--COMPAT6—due to a low item loading. COMPAT1, COMPAT2, COMPAT3, COMPAT4 and COMPAT5 had high item loadings (>.707), loaded more strongly on compatibility than any other construct and exhibited high reliability (>.7).

One item was dropped from perceived ease of use--PEOU2--because it loaded more highly on attitude and because it exhibited low reliability. Although PEOU 3, PEOU5 and PEOU7 exhibited low item loadings, they were kept to be consistent with prior research. Even though PEOU8 loaded more strongly on attitude than perceived ease of use, it was kept to remain consistent with prior research. PEOU1, PEOU4 and PEOU6 had high item loadings (>.707), loaded more strongly on perceived ease of use than any other construct and exhibited high reliability (> .7).

The average for Voluntariness was 3.82—in between slightly disagree and undecided. This implies participants, on average, perceived a methodology as more mandatory than voluntary.

Research Instrument

The survey instrument contains 3 sections: 1) items measuring the dimensions of perceived usefulness, perceived ease of use, compatibility, and developer attitude toward using a systems development methodology, 2) items measuring systems development self-efficacy, and 3) demographic and additional questions. Items in each section were randomized to reduce order effects. The items included in the final instrument are contained in Appendix E.

Section 1: Questions measuring dimensions of perceived usefulness, perceived ease of use, compatibility, and developer attitude toward using a systems development methodology.

All items, except self-efficacy, were measured on a 1 to 7 Likert scale where 1 represents strongly disagree and 7 represents strongly agree. Although some researchers question the statistical validity of analyzing Likert scale data with parametric statistical techniques (Townsend & Ashby, 1984) , prior research has used 7-point Likert scales to measure compatibility and perceived ease of use (Hardgrave et al, 2003; Riemenschneider et al., 2002). These two constructs are included in the research model shown in Figure 3.1. Thus, to remain consistent with prior research and to report findings in an accepted manner, this study used a 7-point Likert scale to measure the constructs in the proposed research model.

In addition to using a 7-point Likert scale to remain consistent with prior research, this study used a 7-point Likert scale for an additional four reasons. First, prior research has established that Likert scales are more reliable than the Semantic Differntial scale, the Guttman scale and the Thurstone scale (Tittle & Hill, 1967).

Second, Likert-scale data is close enough to interval data (Borgatta & Bohrnstedt, 1980) so that analyzing Likert scale data for many statistical tests does not severely affect Type I or Type II errors (Jaccard & Choi, 1996). Third, the categories developed by Stevens (1951) may be inappropriate (Velleman & Wilkinson, 1993); thus, the nominal, ordinal, interval and interval classifications may not constitute the best typology for data.

As a result, the distinction between nominal, ordinal, interval and ratio may not be clear cut. Fourth, Partial Least Squares (PLS) regression can be used with ordinal or even dichotomous variables (Chin, 1996) and does not require parametric assumptions for the bootstrapping method, which is used to determine the significance of the path coefficients. (Gefen et al., 2000). Although the constructs in the proposed research model were measured using a 7 point Likert scale, systems development self-efficacy was measured on a 1 to 10 anchored scale to remain consistent with prior research (Compeau & Higgins, 1995a).

Product usefulness

Product usefulness is defined as the degree to which using a methodology increases the quality of the developed system. All items are measured on a 1 to 7 Likert scale where 1 represents strongly disagree and 7 represents strongly agree. Table 3.2 presents the items for Product Usefulness.

Table 3.2 Product Usefulness Items

Item	Source
Using the Rational Unified Process[7] helps to develop more efficient systems.	Huisman & Iivari, 2006
Using the Rational Unified Process helps to develop better systems.	Huisman & Iivari, 2006; Johnson et al., 1999; Khalifa & Verner, 2000
Using the Rational Unified Process helps to develop more reliable systems.	Huisman & Iivari, 2006
Using the Rational Unified Process helps to develop higher quality systems.	Huisman & Iivari, 2006; Johnson et al., 1999; Khalifa & Verner, 2000
Using the Rational Unified Process helps to develop more maintainable systems.	Huisman & Iivari, 2006; Johnson et al., 1999; Khalifa & Verner, 2000

[7] All questions in the final instrument were oriented toward the Rational Unified Process. As discussed later in this chapter, the Rational Unified Process was chosen as a representative formalized commercial methodology.

Process usefulness

Process usefulness is defined as the degree to which using a methodology improves the productivity of the systems development process. All items are measured on a 1 to 7 Likert scale where 1 represents strongly disagree and 7 represents strongly agree. Table 3.3 presents the items for process usefulness.

Table 3.3 Process Usefulness Items

Item	Source
Using the Rational Unified Process helps to keep the development project on schedule.	Huisman & Iivari, 2006; Johnson et al., 1999; Khalifa & Verner, 2000
Using the Rational Unified Process improves schedule performance.	Huisman & Iivari, 2006; Johnson et al., 1999; Khalifa & Verner, 2000
Using the Rational Unified Process helps to meet scheduled delivery dates.	Huisman & Iivari, 2006; Johnson et al., 1999; Khalifa & Verner, 2000
Using the Rational Unified Process results in a more efficient development process.	Huisman & Iivari, 2006; Johnson et al., 1999; Khalifa & Verner, 2000
Using the Rational Unified Process increases the productivity of the development process.	Huisman & Iivari, 2006; Johnson et al., 1999; Khalifa & Verner, 2000

Communication usefulness

Communication usefulness is defined as the degree to which using a methodology improves communication with users and other team members. All items are measured on a 1 to 7 Likert scale where 1 represents strongly disagree and 7 represents strongly agree. Table 3.4 presents the items for communication usefulness.

Table 3.4 Communication Usefulness Items

Item	Source
Using the Rational Unified Process improves communication with the user's manager(s).	Johnson et al., 1999
Using the Rational Unified Process improves communication with managers.	Johnson et al., 1999
Using the Rational Unified Process improves communication with users.	Khalifa & Verner, 2000
Using the Rational Unified Process improves communication with other team members.	Johnson et al., 1999

Career usefulness

Career usefulness represents the degree to which a methodology improves marketability, increases job flexibility, raises chances for promotion, and offers opportunity for more meaningful work. All items are measured on a 1 to 7 Likert scale where 1 represents strongly disagree and 7 represents strongly agree. Table 3.5 presents the items for the career usefulness.

Table 3.5 Career Usefulness Items

Item	Source
Using the Rational Unified Process increases my opportunity to gain job security.	Hardgrave et al., 2003
Using the Rational Unified Process increases my chances for pay increases.	Hardgrave et al., 2003
Using the Rational Unified Process increases my chances for promotion.	Hardgrave et al., 2003
Using the Rational Unified Process improves my marketability.	Johnson et al., 1999
Using the Rational Unified Process increases my flexibility of changing jobs.	Hardgrave et al., 2003

Symbolic usefulness

Symbolic usefulness is defined as the degree to which using a methodology shows others that professional development processes are being used. Symbolic usefulness has not been included in prior theoretical models; the items used to operationalize symbolic usefulness are new. All items are measured on a 1 to 7 Likert scale where 1 represents strongly disagree and 7 represents strongly agree. Table 3.6 presents the items for the Symbolic Usefulness.

Table 3.6 Symbolic Usefulness Items

Item	Source
Using the Rational Unified Process indicates to others that sound development practices are being used.	Fitzgerald, 1998b; Fitzgerald et al., 2002; Kautz et al., 2004; Nandhakumar & Avison, 1999; Robey & Markus, 1984
Using the Rational Unified Process shows others that professional development practices are being used.	Fitzgerald, 1998b; Fitzgerald et al., 2002; Kautz et al., 2004; Nandhakumar & Avison, 1999; Robey & Markus, 1984
Using the Rational Unified Process shows others that industry standard systems development practices are being used.	Fitzgerald, 1998b; Fitzgerald et al., 2002; Kautz et al., 2004; Nandhakumar & Avison, 1999; Robey & Markus, 1984
Using the Rational Unified Process indicates to others that systematic systems development practices are being used.	Fitzgerald, 1998b; Fitzgerald et al., 2002; Kautz et al., 2004; Nandhakumar & Avison, 1999; Robey & Markus, 1984
Using the Rational Unified Process signifies to others that accepted systems development practices are being used.	Fitzgerald, 1998b; Fitzgerald et al., 2002; Kautz et al., 2004; Nandhakumar & Avison, 1999; Robey & Markus, 1984

Support usefulness

Support usefulness is defined as the degree to which using a methodology alleviates developer anxiety. Support usefulness has not been included in prior theoretical models; the items used to operationalize support usefulness are new. All items are measured on a 1 to 7 Likert scale where 1 represents strongly disagree and 7 represents strongly agree. Table 3.7 presents the items for support usefulness.

Table 3.7 Support Usefulness Items

Item	Source
Using the Rational Unified Process reduces developer stress during systems development.	Fitzgerald, 1998b; Fitzgerald et al., 2002; Kautz et al., 2004; Wastell ,1996; Wastell, 1999
Using the Rational Unified Process helps developers deal with the pressures of systems development.	Fitzgerald, 1998b; Fitzgerald et al., 2002; Kautz et al., 2004; Wastell ,1996; Wastell, 1999
Using the Rational Unified Process relieves developer apprehension during systems development.	Fitzgerald, 1998b; Fitzgerald et al., 2002; Kautz et al., 2004; Wastell ,1996; Wastell, 1999
Using the Rational Unified Process helps developers cope with the stress of systems development.	Fitzgerald, 1998b; Fitzgerald et al., 2002; Kautz et al., 2004; Wastell ,1996; Wastell, 1999
Using the Rational Unified Process reduces developer anxiety during systems development.	Fitzgerald, 1998b; Fitzgerald et al., 2002; Kautz et al., 2004; Wastell ,1996; Wastell, 1999

Defense usefulness

Defense usefulness is defined as the degree to which using a methodology provides protection in case decisions made during systems development turn out to be wrong and protects against unreasonable user demands. Defense usefulness has not been included in prior theoretical models; the items used to operationalize defense usefulness are new. All items are measured on a 1 to 7 Likert scale where 1 represents strongly disagree and 7 represents strongly agree. Table 3.8 presents the items for defense usefulness.

Table 3.8 Defense Usefulness Items

Item	Source
The Rational Unified Process can be used to justify decisions made during systems development.	Fitzgerald, 1998b; Fitzgerald et al., 2002; Robey & Markus, 1984
If a decision made during systems development turns out to be wrong, the Rational Unified Process can be used to show the rationale for the decision.	Fitzgerald, 1998b; Fitzgerald et al., 2002; Robey & Markus, 1984
Using the Rational Unified Process provides protection from unreasonable demands.	Fitzgerald, 1998b; Fitzgerald et al., 2002; Robey & Markus, 1984
The Rational Unified Process can be used to authorize decisions made during systems development.	Fitzgerald, 1998b; Fitzgerald et al., 2002; Robey & Markus, 1984
Using the Rational Unified Process defends against unreasonable demands.	Fitzgerald, 1998b; Fitzgerald et al., 2002; Robey & Markus, 1984
The Rational Unified Process can be used to authorize decisions made during systems development.	Fitzgerald, 1998b; Fitzgerald et al., 2002; Robey & Markus, 1984

Perceived ease of use

Perceived ease of use refers to the degree to which developers perceive a methodology as easy to use (Hardgrave et al., 2003). In order to remain consistent with prior research on methodology adoption, the items from Hardgrave et al. (2003) and Riemenschneider et al. (2002) are used in this study. All items are measured on a 1 to 7 Likert scale where 1 represents strongly disagree and 7 represents strongly agree. Table 3.9 summarizes the items for the perceived ease of use scale.

Table 3.9 Perceived Ease of Use Items

Item	Source
Learning the Rational Unified Process was easy for me.	Hardgrave et al., 2003; Riemenschneider et al., 2002
I think the Rational Unified Process is clear and understandable.	Hardgrave et al., 2003; Riemenschneider et al., 2002
It was easy for me to become skillful at using the Rational Unified Process.	Venkatesh et al., 2003
Using the Rational Unified Process does not require a lot of mental effort.	Hardgrave et al., 2003; Riemenschneider et al., 2002
I find the Rational Unified Process easy to use.	Hardgrave et al., 2003; Riemenschneider et al., 2002
Using the Rational Unified Process does not take too much time from my normal duties.	Hardgrave et al., 2003; Riemenschneider et al., 2002
The Rational Unified Process is not cumbersome to use.	Hardgrave et al., 2003; Riemenschneider et al., 2002

Compatibility

Compatibility is defined as the degree to which a methodology is consistent with existing software development practices (Hardgrave et al., 2003). In order to remain consistent with prior research on methodology adoption, the items from various sources are used in this study. All items are measured on a 1 to 7 Likert scale where 1 represents strongly disagree and 7 represents strongly agree. Table 3.10 summarizes the items for compatibility.

Table 3.10 Compatibility Items

Item	Source
Using the Rational Unified Process is compatible with all aspects of my work.	Hardgrave et al., 2003; Riemenschneider et al., 2002
The Rational Unified Process is compatible with the way I develop systems.	Hardgrave et al., 2003; Riemenschneider et al., 2002
Using the Rational Unified Process is completely compatible with my current situation.	Templeton & Byrd, 2003
Using the Rational Unified Process fits into my work style.	Moore & Benbasat, 1991
Using the Rational Unified Process fits well with the way I work.	Hardgrave et al., 2003; Riemenschneider et al., 2002

Developer attitude toward using a methodology

This section contains items that measure the dependent variable—developer attitude toward using a methodology. In the context of methodologies, attitude refers to the degree to which a developer has favorable or unfavorable evaluations of using a methodology (Ajzen, 1988; 1991). Items used to measure attitude were combined from various sources and were modified to assess developer attitude toward using formalized commercial methodologies. All items are measured on a 1 to 7 Likert scale where 1 represents strongly disagree and 7 represents strongly agree. Table 3.11 summarizes the items used to assess attitude.

Table 3.11 Attitude Items
***Denotes reverse-coded item**

Item	Source
Using the Rational Unified Process is a good idea.	Nah, Tan & Teh, 2004
*Using the Rational Unified Process is unpleasant.	Taylor & Todd, 1995
Using the Rational Unified Process is a wise idea.	Taylor & Todd, 1995
Using the Rational Unified Process is beneficial.	Oh, Ahn, & Kim, 2003
*Using the Rational Unified Process is a foolish idea.	Moon & Kim, 2001

Section 2: Questions measuring systems development self-efficacy

Systems development self-efficacy is defined as a developer's judgment regarding his or her ability to apply the appropriate techniques, documentation aids, procedures, and tools to successfully build an information system. As discussed in Chapter Two, self-efficacy represents an important covariate for support usefulness.

Self-efficacy is not a measurable construct at a general level because it is domain specific (Dishaw, Strong & Brandy, 2002). New measures of self-efficacy must be developed for each different application of the construct (Dishaw et al., 2002). Consistent with these recommendations, existing measures of self-efficacy were adapted to measure systems development self-efficacy. The measures developed by Compeau and Higgins (1995a) were used as a baseline for this study; the items used in this study were modified to measure systems development self-efficacy. Additional items were added to improve the quality of the scales.

The items from Compeau and Higgins (1995a) are used as a baseline in this study for three reasons. First, consistent with recommendations from Bandura (1986), Compeau and Higgins (1995a) assess both the strength and magnitude of self-efficacy. Second, the Compeau and Higgins (1995a) measure exhibited a high reliability—.95. Third the measure of computer self-efficacy developed by Compeau and Higgins (1995a) has been widely used in the Information Systems literature (Agarwal et al., 2000; Compeau & Higgins, 1995b; Dishaw et al., 2002; Taylor & Todd, 1995; Thompson et al., 2006; Venkatesh & Davis, 1996; Venkatesh et al., 2003). Table 3.12 presents the original items from Compeau and Higgins (1995a).

Table 3.12 Original Computer Self-Efficacy Items (Compeau & Higgins, 1995a)

I could complete the job using the software package….
…if there was no one around to tell me what to do as I go.
…if I had never used a package like it before.
…if I had only the software manuals for reference.
…if I had seen someone else using it before trying it myself.
…if I could call someone for help if I got stuck.
…if someone else had helped me get started.
…if I had a lot of time to complete the job for which the software was provided.
…if I had just the built-in help facility for assistance.
…if someone showed me how to do it first.
...if I had used similar packages before this one to do the same job.

When modifying the items from Compeau and Higgins (1995a), care was taken to ensure that the guidelines proposed by Marakas et al. (1998) were followed. Marakas et al. (1998) offer a framework comprised of five guidelines for the development and modification of instruments intended to measure computer self-efficacy. First, when developing self-efficacy items, researchers should focus on an individual's ability to perform a specific task rather than any benefits related to task performance. Items should not include any wording related to beliefs that performing a specific task will result in desirable outcomes. Second, items should gauge an individual's perception of his or her ability to perform a specific task, rather than ability in a general context. Third, items should be constructed so that they extract perceptions of ability for only one kind of task. For example, the item "I could complete the job using the software application" measures two perceptions of ability: the ability to use the software package and the ability to complete the job. Fourth, the level of analysis for an individual's perceived level of ability must agree with the level of analysis of the task. Fifth, items should be randomized to avoid order effects (Marakas et al., 1998).

The systems development self-efficacy scale measures both the strength and magnitude of systems development self-efficacy (Marakas et al., 1998). Following the approach suggested by Bandura (1986), the systems development self-efficacy scale requires subjects to answer "yes" or "no" to a series of statements. The sum of the positive responses is considered to represent the magnitude of the individual's self-efficacy. Each "yes" response collected during measurement is then rated by the subject on a scale ranging from 1 to 10, where 1="Not Confident" and 10 equals "Totally Confident". The sum of these confidence ratings is used to measure self-efficacy strength. The items for systems development self-efficacy are presented in Table 3.13 and an example item illustrating how an individual would answer a sample question is shown in Table 3.14.

Table 3.13 Self-Efficacy Items

Item	Source
I could develop an information system if there was no one around to tell me what to do as I go.	Compeau & Higgins, 1995a
I could develop an information system if the Rational Unified Process was not available to tell me what to do.	Compeau & Higgins, 1995a
I could develop an information system if I had only the Rational Unified Process for reference.	Compeau & Higgins, 1995a
I could develop an information system without following the Rational Unified Process.	New item
I could develop an information system if I had never developed one like it before.	Compeau & Higgins, 1995a
I could develop an information system if I could call someone for help if I got stuck.	Compeau & Higgins, 1995a
I could develop an information system if someone else helped me get started.	Compeau & Higgins, 1995a
I could develop an information system if I could have a lot of time to create it.	Compeau & Higgins, 1995a
I could develop an information system if someone showed me how to do it first.	Compeau & Higgins, 1995a
I could develop an information system if I had built a similar information system in the past.	Compeau & Higgins, 1995a
I could develop an information system if I had seen someone develop one before developing one myself.	Compeau & Higgins, 1995a

Table 3.14 Example Self-Efficacy Item (Compeau & Higgins, 1995a)

Item	Scale
I could develop a new information system if there was someone giving me step by step instructions.	Not Confident · Moderately Confident · Totally Confident 1 2 3 4 (5) 6 7 8 9 10 (Yes) No

Section 3: Demographic and additional questions

Several demographic variables were also measured. Venkatesh et al. (2003) find gender and age moderate the effect of performance expectancy, conceptually similar to perceived usefulness, on intention to use, such that the effect of performance expectancy on intention to use is stronger for men, especially younger men. Venkatesh et al. (2003) also find age, gender, and tool experience moderate the effect of effort expectancy (perceived ease of use) on intention to use, such that the effect is stronger for women, especially younger women with less tool experience. Accordingly, age, gender, and tool (methodology) experience were measured. The following additional demographic variables were collected: level of education, job title, years of work experience, years of systems development experience, and the country in which the participant works. Collecting these variables allows for further analysis if required once the research model has been tested.

Prior research also finds that methodology use is higher in certain industries, in organizations consisting of more than 1,000 employees, and in larger Information Systems departments with more than 20 personnel (Fitzgerald, 1998a). Additionally, methodology usage is positively associated with high levels of in-house development and low levels of customization of purchased application and outsourced development (Fitzgerald, 1998a). Finally, methodologies are more likely to be used on projects with more than five developers and when the project duration is greater than nine months (Fitzgerald, 1998a). Accordingly, this research collected the following additional variables: number of employees in the organization, number of employees in the Information Systems department, degree of in-house development, degree of outsourced development, degree of software development package customization, average number of developers on a given software development project within the organization, and average project duration. Collecting these variables allows further analysis if needed once the research model has been tested. Table 3.15 lists the demographic questions.

Table 3.15 Demographic variables

Variable	Question Type	Item	Selection Options	Source
Systems development experience	Open-ended	How much systems development experience do you have? (Select number of years and months)	Years: 1-40 Months: 1-12	New item
Systems development experience	1-7 Likert	I am an experienced systems developer.	Strongly Disagree to Strongly Agree	New item
Methodology experience	Open-ended	How much experience do you have using the Rational Unified Process? (Select number of years and months)	Years: 1-40 Months: 1-12	New item
Job title	Selection	What is your job title?	• Business Analyst • Chief Information Officer • Configuration Management • Database Administrator • Programmer • Programmer/Analyst • Program Manager • Project Leader • Quality Assurance • Software Architect • Software Developer • Software Engineer • Software Metrics Coordinator • Software Process Engineer • Software Tester • Systems Analyst • Systems Architect • Team Leader • Don't Know • Other	New item
Job title	Open-ended	How many years have you had this job title? (Select number of years and months)	Years: 1-40 Months: 1-12	New item
Work experience	Open-ended	How long have you been working in the software industry? (Select number of years and months)	Years: 1-40 Months: 1-12	New item
Work experience	Open-ended	How long have you worked for your current employer? (Select number of	Years: 1-40 Months: 1-12	New item

		years and months)		
Country	Open-ended	In which country do you work?	N/A	New item
Gender	Selection	What is your gender?	• Male • Female	New item
Age	Open-ended	What is your age? (enter a number of years)	N/A	New item
Education	Selection	Select the highest level of education completed.	• Grade school/Some High school • High-school diploma or GED • Some college/no degree • Undergraduate degree • Master's degree • Doctorate degree	New item
Industry	Selection	In what industry do you work?	• Consulting/software house • Govt/pub.sector/educ. • Constr./manuf./distrib • Wholesale/retail trade • Fin./insur./real estate • Services/communication • Other (please specify)	Fitzgerald (1998a)
Organization size	Selection	How many employees are in your organization?	• 1–10 • 10–100 • 100–1000 • 1000–5000 • >5000	Fitzgerald (1998a)
ISD size	Selection	How many employees are in your Information Systems department?	• 1–5 • 5–20 • 20–100 • >100	Fitzgerald (1998a)
In-house development	Open-ended	Describe the development environment in your organization (distribute to a total of 100%)	• Development in-house__% • Development outsourced__% • Use and integration of commercial packages__%	Fitzgerald (1998a)
Number of developers on project	Open-ended	How many developers are normally on a given software development project in your organization?	N/A	Fitzgerald (1998a)
Project duration	Open-ended	What is the average project duration of a software development project in your organization? (Enter number of months)	N/A	Fitzgerald (1998a)
Capability Maturity Model Integration Rating	Selection	What is your organization's current Capability Maturity Model Integration (CMMI) rating?	• 1 • 2 • 3 • 4 • 5 • Don't Know • Not yet assessed	New item
Methodology tailoring	Open-ended	On average, how much does your team adapt a systems development methodology for a given systems development project (Enter in a	N/A	New item

		percentage as a whole number, e.g., 25)?		
Voluntariness	1-7 Likert	My superiors expect me to use this methodology.	Strongly Disagree to Strongly Agree	Moore & Benbasat, 1991
Voluntariness	1-7 Likert	My use of this methodology is voluntary (as opposed to being required).	Strongly Disagree to Strongly Agree	Moore & Benbasat, 1991
Voluntariness	1-7 Likert	My boss does not require me to use this methodology.	Strongly Disagree to Strongly Agree	Moore & Benbasat, 1991
Voluntariness	1-7 Likert	Although it might be helpful, using this methodology is certainly not compulsory in my job.	Strongly Disagree to Strongly Agree	Moore & Benbasat, 1991
Voluntariness	1-7 Likert	The use of this methodology is required for my job.	Strongly Disagree to Strongly Agree	Moore & Benbasat, 1991

Voluntariness

Voluntariness refers to the degree to which an innovation is perceived as being voluntary or of free will (Moore & Benbasat, 1991). The items used to measure voluntariness were derived from Moore and Benbasat (1991). All items are measured on a 1 to 7 Likert scale where 1 represents strongly disagree and 7 represents strongly agree. Table 3.16 summarizes the items for the voluntariness construct.

Table 3.16 Voluntariness items
***Denotes reverse-coded Item**

Item	Source
*My superiors expect me to use the Rational Unified Process.	Moore & Benbasat, 1991
My boss does not require me to use the Rational Unified Process.	Moore & Benbasat, 1991
*The use of the Rational Unified Process is required for my job.	Moore & Benbasat, 1991
Although it might be helpful, using the Rational Unified Process is certainly not mandatory in my job.	Moore & Benbasat, 1991
My use of the Rational Unified Process is voluntary (as opposed to being required).	Moore & Benbasat, 1991

Data Analysis

Partial Least Squares (PLS) regression was used to test the hypotheses presented at the beginning of the chapter. PLS was chosen as the data analysis technique for several reasons. First, PLS deals with both reflective and formative indicators (Gefen et al., 2000). Second, PLS is better than covariance-based SEM (i.e., LISREL) for exploratory research (Gefen et al., 2000). PLS is used for theory-building and exploratory research, whereas covariance-based SEM is used for confirmatory studies (Gefen et al., 2000). While covariance-based SEM requires strong theory base and supports only confirmatory research, PLS does not require a sound theory base and supports both exploratory and confirmatory research (Gefen et al., 2000). This research uses new constructs derived from case study research on methodology use; these constructs could be construed as exploratory as they have never been operationalized or included in prior empirical research. Finally, PLS is appropriate when theory is untested in an application domain (Gopal, Bostrom & Chin, 1992) and when the research model is in the early stage of development (Bahli & Buyukkurt, 2005).

PLS was also chosen due to its capability to handle higher-order models; PLS supports modeling higher-order factors in which the higher-order factor is measured using the indicators used to measure each of the first-order factors (Bassellier & Benbasat, 2004; Lankton & McKnight, 2007). PLS is also more robust to deviations from a multivariate distribution (Gefen et al., 2000).

An additional analysis was conducted to test whether the research model presented in Figure 3.2 explains more variance in attitude than a competing model using perceived usefulness and its associated items from the TAM (Davis, 1989) as shown in Figure 3.2.

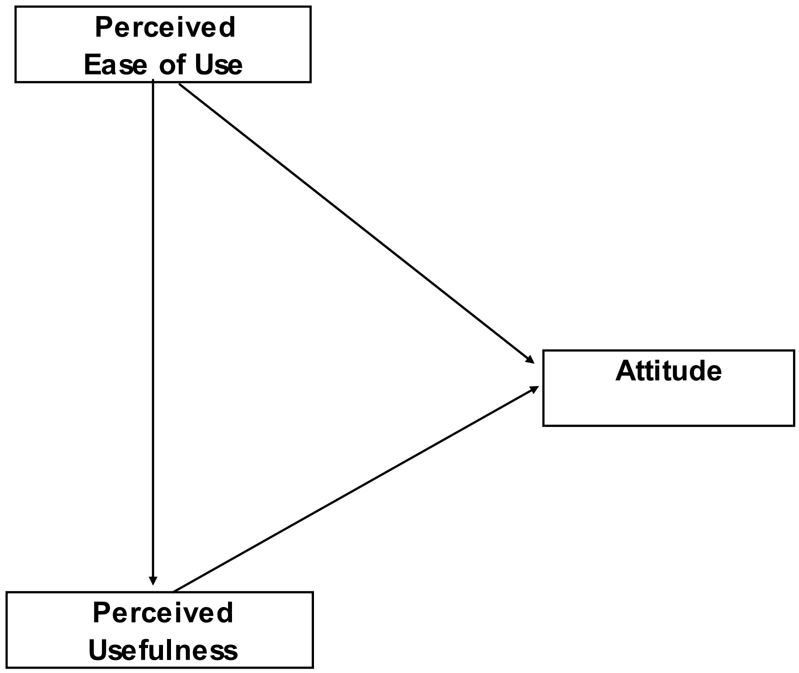

Figure 3.2 Competing Model with TAM Perceived Usefulness Construct

Sample

PLS requires a sample size of 10 times either the construct with the greatest number of formative indicators (Chin, 2000) or 10 times the largest number of structural paths directed at a particular construct in the structural model (Chin, 2000). Political usefulness has the largest number of structural paths going into it, with four. Applying the heuristic for sample size requirements to the research model, a sample size of at least 40 developers will be needed. However, Goodhue, Lewis and Thompson (2006) conducted an experiment to determine whether PLS exhibited adequate power at small sample sizes. Results indicate that the "10 times" rule of thumb results in low power. As a result, an *a-*

priori power analysis was conducted to estimate the appropriate sample size (Chin & Newsted, 1999). Based on an alpha of .05, a medium effect size of .15, and power of .8, approximately 120 subjects were needed (Cohen, 1988).

The target sample was developers who have used the RUP. Although it would be possible to collect data from developers who use other methodologies (e.g., Agile methodologies), prior research suggests that methodologies should not be treated as homogeneous phenomena (Huisman & Iivari, 2006). Therefore, a widely used methodology—the RUP—was chosen as a representative example of a formalized commercial systems development methodology (Ambler, 2005).

To ensure that only developers who have used the RUP participate in the survey, the email included instructions stating that prior experience with the RUP is necessary to take the survey. Furthermore, data collection efforts focused on finding potential subjects via personal contacts and via various RUP user groups (for a list of RUP user groups, see http://www.rational-ug.org/groups.php).

The survey was distributed to potential subjects via email; thus, a web-based survey rather than a paper-based postal survey was used to collect the data. Web-based surveys and experiments are becoming increasingly prevalent in information systems research. They are less costly, more convenient, and provide more control than postal surveys (Couper, 2000). Web-based surveys have been shown to be similar to postal surveys in terms of the quality of data gathered (Bachmann, Elfrink & Gary, 1996; Coderre, Mathieu, St-Laurent, 2004) and the number of items answered (Coderre et al., 2004). Furthermore, computer-administered questionnaires are useful for reducing social desirability bias (Whitley, 2002).

Bryant, Hunton and Stone (2004) discuss the validity of web-based experiments in behavioral accounting research. They identify internal validity threats such as increased dropout rates and multiple submissions from the same participant. Consistent with the recommendations of Bryant et al. (2004), controls were included to account for some of these potential threats to internal validity. To encourage participation in the study and reduce non-response bias, a letter describing the intent of the survey was included in the email to participants (Lazar & Preece, 1999). To further encourage participation, participants were encouraged to provide their email address which entered them into a prize drawing; email addresses, however, were not associated with participant responses. To address the threat of multiple submissions from the same participant, the Internet Protocol addresses and email addresses of participants were examined to ensure that there were no multiple responses. Furthermore, the data collection window was limited and the time of each participant's response session was examined to verify that the survey was completed in one sitting.

Internal Validity Threats

This study faces an internal validity threat due to social desirability bias. Social desirability bias refers to the tendency for participants to respond in a way that makes them look good to others (Whitley, 2002). Because the dimensions of political usefulness represent more covert ways of using a methodology, rather than those ways of using a methodology as advertised by methodologists (e.g., improve software quality), it is possible that subjects may believe that using a methodology for its symbolic, support, or defense usefulness is socially undesirable.

Social desirability bias can be ameliorated through anonymity and through the use of computer-administered questionnaires (Whitley, 2002). Consistent with these recommendations, a computer-administered questionnaire (web-based survey) was used and anonymity was guaranteed.

Chapter Four

Data Analysis

This chapter presents an analysis of the data obtained from the final data collection. The detailed discussion of the results and their implications is presented in Chapter 5. Chapter 6 summarizes the contributions and limitations of this study and provides avenues for future research.

This chapter is organized as follows. The first section provides a descriptive analysis of the data. The second section provides an overview of the partial least squares statistical technique, used to test the research model. The third section discusses the measurement model and provides measures of internal consistency as well as convergent and discriminant validity. The fourth section discusses the structural model and provides the results of the hypotheses testing.

Descriptive Analysis of the Data

The survey was conducted from June 4th, 2007 until July 13th, 2007. The invitation letter to the survey was distributed via email to a point of contact within each organization; each point of contact then subsequently emailed the survey to potential participants. Each participant's response session was inspected to verify that the survey was completed in one sitting.

A total of 165 responses were received. An analysis of the Internet Protocol and email addresses revealed no duplicate responses. Participants who did not answer at least 66% of the questions measuring the constructs in the research model were eliminated from the sample, resulting in the removal of 27 responses. Six outliers that were plus/minus 3 standard deviations from the mean were also removed (Cohen, 1969). The eight participants who completed at least 66% of the questions measuring the constructs in the research model, but did not complete the demographic questions or self-efficacy questions, were included in the analysis. The item mean was substituted for questions which were not answered by these eight participants (Cohen & Cohen, 1983).

As shown in Table 4.2, the job titles reported by the participants ran the gamut of various Information Technology functions. The variety of job titles listed is consistent with prior research on developer adoption of methodologies (Hardgrave & Johnson, 2003; Hardgrave et al., 2003; Huisman & Iivari, 2006; Johnson et al., 1999; Kozar, 1989; Riemenschneider et al., 2002; Sultan & Chan, 2000).

Independent samples t-tests were calculated to determine whether participants who indicated that their job title was not specifically developer-related (e.g., “Configuration Management” or the eight participants with missing job titles) exhibited significantly different perceptions of the constructs in the research model from participants who indicated that their job title was developer-related. Results from the t-tests indicate that participants who identified their job title as “Quality Assurance” and “Systems Planning Specialist” have significantly different perceptions of the dimensions of perceived usefulness from participants who reported a developer-related job title. As a result, 11 participants who identified their job title as “Quality Assurance” and one participant who identified his job code as “Systems Planning Specialist” were removed from the sample.

The results of the t-tests also showed marginally significant differences (p=.05) between participants who reported a developer-related job title and participants who reported their job title as "Manager" or "Software Process Engineer". Although there were marginally significant differences between these groups, participants who reported their job title as either "Manager" or "Software Process Engineer" were retained for several reasons. First, the difference between the groups is borderline significant at the .05 alpha level for the support usefulness construct, but not for the other constructs in the research model. Second, prior research has included managers and other job titles when conducting research on developer adoption of methodologies (Hardgrave & Johnson, 2003; Hardgrave et al., 2003; Huisman & Iivari, 2006; Johnson et al., 1999; Kozar, 1989; Riemenschneider et al., 2002; Sultan & Chan, 2000). Third, the inclusion of participants who identified their job code as "Manager" or "Software Process Engineer" does not change the significance of the t-statistics for each path coefficient, does not significantly change the strength of the path coefficients, and increases the power of the analysis and the generalizability of the results.

The final sample size is 120 subjects, which meets the sample size needed per the *a priori* power analysis conducted as discussed in Chapter 3. Table 4.1 below shows how the final sample size was calculated.

Table 4.1 Sample Size Calculation

Total # of Responses	165
Less Incomplete	27
Less Outliers	6
Less 'Quality Assurance' job title	11
Less 'Systems Planning Specialist' job title	1
Final Sample Size	120

Table 4.2 Participant Job Titles Included in Final Sample

Job Title	Frequency	Percentage	Cumulative Percentage
Analyst	22	18.3%	18.3%
Configuration Management	5	4.2%	22.5%
Program Manager	5	4.2%	26.7%
Programmer	2	1.7%	28.3%
Programmer/Analyst	1	0.8%	29.2%
Project Leader	10	8.3%	37.5%
Software Architect	15	12.5%	50.0%
Software Developer	9	7.5%	57.5%
Software Engineer	6	5.0%	62.5%
Software Metrics Coordinator	1	0.8%	63.3%
Software Process Engineer	8	6.7%	70.0%
Software Tester	1	0.8%	70.8%
Systems Architect	7	5.8%	76.7%
Team Leader	1	0.8%	77.5%
Other	19	15.8%	93.3%
Missing	8	6.7%	100.0%

Table 4.3 Job titles for participants who identified their job title as "Other"

Job Title	Frequency	Percentage	Cumulative Percentage
Business Architect	1	5.3%	5.3%
Configuration team manager/ Development project manager	1	5.3%	10.5%
Consultant	2	10.5%	21.1%
CTO	1	5.3%	26.3%
Director of Software Development	1	5.3%	31.6%
Engagement Manager/Process Engineer	1	5.3%	36.8%
Enterprise Architect	1	5.3%	42.1%
Process Engineer	1	5.3%	47.4%
Process Improvement Leader	1	5.3%	52.6%
Project Manager III	1	5.3%	57.9%
Quality Manager	1	5.3%	63.2%
Requirements Analyst	1	5.3%	68.4%
RUP Coach and Mentor	1	5.3%	73.7%
Software Integrator	1	5.3%	78.9%
Systems Engineering Consultant	1	5.3%	84.2%
Vice President of Operations	1	5.3%	89.5%
Vice President, Operations	1	5.3%	94.7%
Web applications manager	1	5.3%	100.0%

As shown in Table 4.4 below, the sample came from a variety of sources. Pulling the data from a variety of sources should increase the external validity of the results.

Table 4.4 Sample Sources

Source	Frequency	Percent	Cumulative Percent
Company A	3	2.5%	2.5%
Company B	10	8.3%	10.8%
Company C	33	27.5%	38.3%
Company D	25	20.8%	59.1%
Requirements Engineering/CMMi User Groups	13	10.8%	69.9%
Rational Unified Process User Groups	26	21.7%	91.6%
Agile Unified Process and Enterprise Unified Process User Groups	10	8.3%	100%

Table 4.5 shows the means and standard deviations for each construct in the research model.

Table 4.5 Construct Means and Standard Deviations

Construct	# Items	Mean	Standard Deviation
Attitude	4	5.53	0.94
Product Usefulness	5	5.33	1.00
Process Usefulness	5	5.09	0.99
Communication Usefulness	4	5.36	0.99
Career Usefulness	3	4.36	1.33
Symbolic Usefulness	5	5.70	0.82
Support Usefulness	5	4.69	1.14
Self Efficacy	8	7.36	2.31
Defense Usefulness	3	4.23	1.39
Perceived Ease of Use	6	4.90	1.09
Compatibility	3	5.48	1.03
Perceived Usefulness (TAM)	6	5.32	0.99
Voluntariness	5	3.87	1.50

Various demographic variables were also collected. The average participant had 13 years and three months of systems development experience, five years and two months of experience using the Rational Unified Process, and 14 years and two months of experience in the software industry. The average age of the participant was 40 years old; 72.5% of the participants were male, 16.7% of the participants were female, and 10.8% provide their gender. On average, participants adapt the Rational Unified Process 57.2% of the time. The typical software development project is two years and one month in duration. Participants indicate that on average, 61.22% of development is conducted in-house, 14.56% of development is outsourced and 24.21% of development is conducted using commercial software. As shown in Table 4.6, most participants have an undergraduate or graduate degree.

Table 4.6 Education Distribution

Education	Frequency	Percentage	Cumulative Percentage
Grade School/Some High School	2	1.8%	1.8%
High-school diploma or GED	3	2.5%	4.6%
Some college/no degree	5	4.6%	9.2%
Undergraduate degree	48	44%	53.2%
Master's degree	45	41.3%	94.5%
Doctorate degree	6	5.5%	100%

Most participants are members of large organizations and large Information Systems departments, as shown in Tables 4.7 and 4.8.

Table 4.7 Number of employees in organization

Category	Frequency	Percent	Cumulative Percent
1–10	2	1.8%	1.8%
10–100	11	9.8%	11.6%
100–1000	49	43.8%	55.4%
1000–5000	14	12.5%	67.9%
>5000	36	32.1%	100%

Table 4.8 Number of employees in Information Systems department

Category	Frequency	Percent	Cumulative Percent
1–5	8	7.1%	7.1%
5–20	16	14.3%	21.4%
20–100	20	17.9%	39.3%
>100	68	60.7%	100%

As shown in Table 4.9, most participants are either unsure of their organization's CMMi rating or their organization had not yet been assessed.

Table 4.9 Capability Maturity Model Integration (CMMi) rating

CMMi Rating	Frequency	Percent	Cumulative Percent
1	7	6.3	6.3
2	11	9.8	16.1
3	19	17	33
4	5	4.5	37.5
5	2	1.8	39.3
Don't know	27	24.1	63.4
Not yet assessed	41	36.6	100

Overview of Partial Least Squares (PLS) Regression

Structural equation modeling (SEM) constitutes the most appropriate statistical technique as the proposed research model hypothesizes relationships among multiple latent constructs measured with numerous manifest variables (Gefen et al., 2000). SEM techniques are second-generation statistical techniques that unite an econometric perspective focusing on prediction with a psychometric perspective focusing on measuring latent constructs with manifest variables (Gefen et al., 2000). Unlike ordinary least squares regression (OLS), SEM allows the researcher to simultaneously test the structural model—the proposed causation among a set of dependent and independent variables—and the measurement model—the loadings of manifest variables on their proposed latent constructs (Gefen et al., 2000).

The two most widely used SEM techniques in the Information Systems literature are covariance-based SEM and partial-least squares based (PLS) SEM (Gefen et al., 2000). For the reasons discussed in Chapter 3 of this study, PLS was deemed the more appropriate SEM technique for testing the proposed research model.

Analysis of the proposed research model consists of first estimating the measurement model and then subsequently testing the structural model (Gefen et al., 2000). Assessment of the measurement and structural model was conducted using Smart PLS version 2.0 beta (Ringle, Wend & Will, 2005). Smart PLS was chosen over PLS-Graph because it provides enhanced support for moderating variables (Temme, Kreis, Hildebrandt, 2006). Furthermore, Smart PLS provides similar results to PLS-Graph (Temme et al., 2006). The assessment of the measurement and structural models are discussed in the following sections.

Measurement Model

As discussed in Chapter 2, product usefulness, process usefulness, and communication usefulness are first-order reflective constructs. Rational usefulness is hypothesized to be a second-order molar construct formed by product usefulness, process usefulness and communication usefulness.

Similarly, career usefulness, symbolic usefulness, support usefulness and defense usefulness are first-order reflective constructs. Political usefulness is conceptualized as a second-order molar construct composed of career usefulness, symbolic usefulness, support usefulness and defense usefulness. Perceived usefulness is hypothesized to be a third-order molar construct composed of rational usefulness and political usefulness.

Construct validity

Construct validity pertains to the "validity of inferences about unobserved variables (constructs) on the basis of observed variables (manifest indicators)" (Pedhazur & Schmelkin, 1991, p.52). A construct exhibits high validity when all items load on one and only one construct. A Confirmatory Factor Analysis (CFA) using PLS was conducted to assess the convergent and discriminant validity of the constructs in the research model.

Convergent validity

Convergent validity is the degree to which the items of a particular scale measure the same construct (Whitley, 2002). Convergent validity should only be assessed for reflective first-order constructs, not for higher-order formative constructs (Bassellier & Benbasat, 2004; Staples & Seddon, 2004). Thus, convergent validity was assessed for the seven dimensions of perceived usefulness, attitude, compatibility, perceived ease of use and self-efficacy, but not for the second-order formative constructs of rational usefulness and political usefulness or the third-order formative construct of perceived usefulness.

Convergent validity is assessed in PLS by examining the composite reliability, the Average Variance Extracted (AVE), the item factor loadings (Fornell & Larcker, 1981; Lankton & McKnight, 2007) and the significance of the outer loadings (Gefen & Straub, 2005). Convergent validity is shown when the AVE for each construct is greater than .5, the composite reliability is greater than .7, the items load on their respective construct greater than or equal to .707 (Bassellier & Benbasat, 2004; Fornell & Larcker, 1981), and the t-statistic of the outer loading is greater than 1.96 (Gefen & Straub, 2005).

The internal consistency of the scales used to measure each construct is assessed via Cronbach's alpha and composite reliability (Fornell & Larcker, 1981). Measures of internal consistency are only calculated for the first-order reflective scales, not for the second-order formative constructs (Bassellier & Benbasat, 2004; Bollen, 1984; Staples & Seddon, 2004). Nunally (1978) suggests that values of .7 or higher are adequate for Cronbach's alpha. As shown in Table 4.10, Cronbach's alpha for each dimension of perceived usefulness, compatibility, perceived ease of use, self-efficacy and attitude are all above the .7 recommended cutoff.

Composite reliability is another measure of internal consistency. According to Chin (1998b):

> In comparison to Cronbach's alpha, this (the internal consistency) measure does not assume tau equivalency among the measures with its assumption that all indicators are equally weighted. Therefore, alpha tends to be a lower bound estimate of reliability, whereas the internal consistency measure (composite reliability) is a closer approximation under the assumption that the parameter estimates are accurate. (p.320)

Composite reliability of .7 or higher is considered acceptable (Agarwal & Karahanna, 2000; Staples & Seddon, 2004) As shown in Table 4.10, composite reliability for each dimension of perceived usefulness, compatibility, perceived ease of use, self-efficacy and attitude are all above the .7 recommended cutoff.

Table 4.10 Measures of Internal Consistency

Scale	No. of Items	Cronbach's Alpha	Composite Reliability
Product Usefulness	5	.89	.92
Process Usefulness	5	.89	.92
Communication Usefulness	4	.89	.92
Career Usefulness	3	.83	.90
Symbolic Usefulness	5	.89	.92
Support Usefulness	5	.90	.92
Defense Usefulness	3	.85	.91
Self Efficacy	8	.92	.92
Compatibility	3	.90	.94
Perceived Ease of Use	6	.87	.90
Attitude	4	.84	.89

To show convergent validity, all item loadings (outer loadings) should be greater than .707, indicating that more than half of the variance is captured by the constructs (Agarwal & Karahanna, 2000; Bassellier & Benbasat, 2004). The following items, with their associated loadings, were removed because their respective loadings were below .707 (Bassellier & Benbasat, 2004): ATT8 (.5346), CAREER7 (.6849), CAREER9 (.5717), COMPAT1 (.6363), COMPAT3 (.7005), DEF10 (.547), DEF12 (.4604), DEF5 (.6462), PEOU5 (.3458), SE11 (.0617), SE12 (.3445), and SE14 (-0.0661). Although the item loadings for SE1 and SE2 are below the recommended .707 cutoff, they were retained to remain consistent with prior literature (Compeau & Higgins, 1995a). All items have t-statistics greater than 1.96, further indicating strong convergent validity (Gefen & Straub, 2005). Table 4.11 shows the item loadings.

Table 4.11 Item Loadings

	SUP	ATT	CAREER	COMM	COMPAT	DEF	PEOU	PROC	PROD	SE	SYMB
ANX1	**0.79**	0.46	0.36	0.57	0.35	0.28	0.34	0.53	0.46	0.09	0.34
ANX3	**0.84**	0.49	0.27	0.58	0.48	0.31	0.33	0.58	0.61	0.18	0.41
ANX5	**0.83**	0.47	0.43	0.49	0.46	0.53	0.43	0.64	0.53	0.16	0.44
ANX8	**0.89**	0.41	0.33	0.56	0.40	0.44	0.30	0.55	0.44	0.22	0.38
ANX9	**0.86**	0.40	0.29	0.48	0.46	0.52	0.36	0.59	0.45	0.04	0.34
ATT1	0.48	**0.87**	0.21	0.56	0.61	0.17	0.52	0.71	0.67	0.10	0.46
ATT10	0.35	**0.74**	0.11	0.44	0.53	0.02	0.50	0.53	0.50	0.19	0.37
ATT5	0.46	**0.82**	0.23	0.50	0.62	0.15	0.61	0.63	0.57	0.18	0.55
ATT7	0.45	**0.86**	0.37	0.59	0.70	0.22	0.50	0.67	0.60	0.21	0.61
CAREER2	0.25	0.24	**0.87**	0.25	0.23	0.32	0.27	0.27	0.25	0.01	0.29
CAREER5	0.42	0.26	**0.89**	0.42	0.32	0.47	0.25	0.37	0.35	0.04	0.35
CAREER6	0.36	0.25	**0.83**	0.30	0.24	0.30	0.15	0.33	0.25	-0.04	0.29
COMM1	0.50	0.50	0.39	**0.84**	0.51	0.34	0.28	0.51	0.58	0.20	0.46
COMM2	0.48	0.46	0.33	**0.87**	0.47	0.34	0.32	0.57	0.61	0.24	0.53
COMM3	0.63	0.54	0.26	**0.86**	0.52	0.37	0.43	0.63	0.68	0.21	0.48
COMM4	0.55	0.66	0.33	**0.84**	0.57	0.34	0.41	0.66	0.61	0.21	0.53
COMPAT2	0.46	0.66	0.28	0.59	**0.90**	0.23	0.43	0.58	0.54	0.28	0.53
COMPAT4	0.40	0.62	0.25	0.46	**0.84**	0.28	0.48	0.55	0.39	0.02	0.37
COMPAT5	0.51	0.72	0.31	0.58	**0.94**	0.28	0.49	0.64	0.51	0.14	0.45
DEF13	0.51	0.22	0.46	0.37	0.31	**0.87**	0.17	0.43	0.28	0.01	0.28
DEF6	0.40	0.08	0.33	0.32	0.21	**0.90**	0.11	0.33	0.21	-0.13	0.14
DEF9	0.41	0.15	0.34	0.38	0.24	**0.88**	0.24	0.40	0.31	-0.12	0.21
PEOU1	0.27	0.30	0.16	0.27	0.24	0.17	**0.74**	0.19	0.23	0.06	0.25
PEOU3	0.27	0.50	0.20	0.32	0.28	0.12	**0.77**	0.35	0.40	0.09	0.41
PEOU4	0.25	0.34	0.23	0.26	0.29	0.15	**0.79**	0.20	0.23	0.06	0.36
PEOU6	0.29	0.49	0.33	0.40	0.40	0.19	**0.81**	0.38	0.37	0.10	0.42
PEOU7	0.41	0.61	0.10	0.35	0.54	0.17	**0.75**	0.50	0.42	0.10	0.37
PEOU8	0.39	0.59	0.21	0.33	0.53	0.14	**0.81**	0.47	0.42	0.00	0.37
PROC2	0.53	0.61	0.33	0.55	0.46	0.32	0.36	**0.81**	0.65	0.05	0.40
PROC4	0.60	0.59	0.32	0.53	0.53	0.42	0.43	**0.86**	0.56	0.09	0.51
PROC5	0.59	0.67	0.32	0.55	0.50	0.38	0.38	**0.88**	0.61	0.08	0.51
PROC6	0.58	0.64	0.35	0.71	0.51	0.39	0.34	**0.78**	0.71	0.19	0.50
PROC8	0.57	0.70	0.24	0.54	0.73	0.33	0.49	**0.82**	0.59	0.01	0.47
PROD1	0.47	0.52	0.32	0.59	0.34	0.26	0.27	0.63	**0.83**	0.14	0.43
PROD2	0.47	0.73	0.26	0.59	0.61	0.21	0.48	0.71	**0.85**	0.10	0.52
PROD3	0.47	0.50	0.27	0.61	0.37	0.22	0.35	0.49	**0.81**	0.27	0.46
PROD5	0.48	0.65	0.18	0.61	0.51	0.19	0.47	0.66	**0.88**	0.20	0.58
PROD6	0.60	0.58	0.38	0.68	0.45	0.39	0.38	0.69	**0.85**	0.17	0.47
SE1	-0.10	-0.14	-0.12	-0.19	-0.07	-0.23	-0.11	-0.27	-0.18	**0.43**	-0.18
SE2	0.12	0.10	-0.05	0.08	0.23	-0.06	0.01	0.06	0.09	**0.64**	0.08
SE3	0.00	0.04	-0.10	0.03	0.01	-0.19	-0.08	-0.02	0.03	**0.80**	0.13
SE4	0.15	0.08	-0.03	0.17	0.03	-0.16	-0.02	0.05	0.14	**0.82**	0.18
SE5	0.16	0.22	0.05	0.23	0.20	-0.02	0.14	0.11	0.20	**0.82**	0.31
SE6	0.09	0.06	-0.06	0.18	0.01	-0.12	0.04	-0.02	0.11	**0.82**	0.15
SE7	0.09	0.19	0.02	0.20	0.17	-0.06	0.08	0.06	0.14	**0.87**	0.20
SE8	0.07	0.12	-0.01	0.12	0.08	-0.14	0.04	-0.01	0.08	**0.76**	0.17
SYMB10	0.35	0.49	0.29	0.56	0.43	0.18	0.34	0.46	0.45	0.13	**0.72**
SYMB11	0.38	0.52	0.35	0.53	0.40	0.19	0.43	0.48	0.56	0.25	**0.88**
SYMB12	0.29	0.50	0.29	0.38	0.43	0.21	0.38	0.48	0.45	0.14	**0.82**
SYMB13	0.45	0.47	0.34	0.46	0.37	0.23	0.39	0.47	0.43	0.36	**0.86**
SYMB5	0.43	0.58	0.26	0.54	0.51	0.22	0.46	0.53	0.55	0.30	**0.90**

As further evidence of convergent validity, the AVE for each construct should be greater than .5. The AVE shows the percent of variance captured by a construct (Gefen & Straub, 2005). As shown in Table 4.12, all constructs have an AVE greater than .5, indicating strong convergent validity (Chin, 1998b; Fornell & Larker, 1981).

Table 4.12 AVE

Scale	AVE
Product Usefulness	.70
Process Usefulness	.69
Communication Usefulness	.75
Career Usefulness	.74
Symbolic Usefulness	.70
Support Usefulness	.71
Defense Usefulness	.77
Self Efficacy	.61
Compatibility	.80
Perceived Ease of Use	.59
Attitude	.69

Discriminant validity

Discriminant validity is the degree to which the items of a particular scale measure only the construct they should measure (Whitley, 2002). Discriminant validity was only assessed for the first-order reflective constructs, not for the second-order or third-order formative constructs (Bassellier & Benbasat, 2004; Staples & Seddon, 2004). Thus, discriminant validity was assessed for the seven dimensions of perceived usefulness, attitude, compatibility perceived ease of use and self-efficacy, but not for rational usefulness, political usefulness or perceived usefulness.

Discriminant validity is assessed in two ways. First, the items should load more strongly on their own constructs than on any other constructs in the model (Gefen &

Straub, 2005). As shown in Table 4.11, all items loaded more strongly on their own construct than any other construct.

The second rule for assessing discriminant validity is to determine whether the square root of the AVE of a given construct is larger than its correlation with any other construct (Gefen & Straub, 2005). For a construct to demonstrate discriminant validity, each square root of the AVE should be larger than its correlation with the other constructs (Gefen & Straub, 2005). As shown in Table 4.13, all constructs meet this requirement, thereby indicating discriminant validity.

Table 4.13 AVE and Latent Variable Correlations

	ATT	CAREER	COMM	COMPAT	DEF	PEOU	PROC	PROD	SE	SUP	SYMB
ATT	1.00										
CAREER	0.24	1.00									
COMM	0.63	0.33	1.00								
COMPAT	0.75	0.22	0.61	1.00							
DEF	0.20	0.40	0.42	0.26	1.00						
PEOU	0.64	0.25	0.40	0.52	0.25	1.00					
PROC	0.78	0.37	0.72	0.68	0.49	0.46	1.00				
PROD	0.72	0.32	0.77	0.57	0.37	0.49	0.75	1.00			
SE	0.21	0.01	0.25	0.16	-0.06	0.05	0.12	0.22	1.00		
SUP	0.55	0.41	0.69	0.55	0.58	0.41	0.73	0.61	0.20	1.00	
SYMB	0.67	0.38	0.61	0.51	0.27	0.49	0.61	0.64	0.28	0.51	1.00
AVE	0.69	0.74	0.75	0.80	0.77	0.59	0.69	0.70	0.61	0.71	0.70
Sq(AVE)	0.83	0.86	0.87	0.89	0.88	0.77	0.83	0.84	0.78	0.84	0.84
Pass/Fail	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass

Structural Model

Assessment of the structural model was conducted to test the hypothesized structure of perceived usefulness as well as the hypothesized relationships among the constructs in the research model. Contrary to covariance-based SEM, PLS does not offer goodness of fit statistics. Rather, with PLS, adequate model fit is established with significant path coefficients, acceptably high r-square values and internal consistency above .70 for each construct (Gefen et al., 2000). As shown in the measurement model section of this chapter, the dimensions of perceived usefulness, compatibility, perceived ease of use, attitude and self-efficacy all exhibit composite reliability and Cronbach's alpha above .7.

The t-statistics used to interpret the significance of the path coefficients in the structural model were generated via the bootstrap procedure in PLS (Chin, 1998b). The bootstrap is a nonparametric and distribution-free approach for estimating the significance of the path coefficients in the structural model (Chin, 1998b). The bootstrap was chosen over the jackknife because it is more efficient (Chin, 1998b). The bootstrap was performed with 500 re-samples (Rai, Patnayakuni & Seth, 2006; Yi & Davis, 2003). Statistical tests were assessed using one-tailed, rather than two-tailed, t-tests because the hypotheses are unidirectional.

The repeated indicators approach was used to estimate the second-order molar constructs—rational and political usefulness—and the third-order molar construct—perceived usefulness (Bassellier & Benbasat, 2004). With the repeated indicators approach, the higher-order constructs are directly measured by manifest indicators for the first-order constructs (Chin, 2000; Lohmoller, 1989, p.130-133; Wold, 1982). While this approach repeats the number of manifest indicators, the model can be estimated by the standard PLS algorithm (Chin, 2000; Lohmoller, 1989, p.130-133; Wold, 1982). The repeated indicators approach allows investigation of the relative path weights of the factors forming the higher-order constructs (Bassellier & Benbasat, 2004) and has been used in prior Information Systems research (Bassellier & Benbasat, 2004; Staples & Seddon, 2004).

Systems development self-efficacy

As discussed in Chapter 3, systems development self-efficacy may be a significant covariate for support usefulness. Developers with low systems development self-efficacy may be more prone to use a methodology for its support usefulness (Wastell, 1999) as they may be more likely to use a methodology to alleviate anxiety than developers with high systems development self efficacy. The possible moderating effects of systems development self-efficacy on support usefulness were tested using Smart PLS (Temme et al., 2006). Smart PLS follows the approach recommended by Chin, Marcolin & Newsted (1996) for implementing moderating effects (Temme et al., 2006). Following the recommendations of Chin et al. (1996), the indicators were mean centered prior to estimation of the t-statistics.

Results indicate that systems development self-efficacy is not a significant moderator of support usefulness. Table 4.14 shows the t-statistics and path coefficients for self-efficacy. When a moderating variable is not significant, it can be reasonably concluded that a main effect without an interaction or moderating effect exists (Whitley, 2002); thus, the hypotheses in the proposed research model were tested without systems development self-efficacy.

Table 4.14 Systems Development Self Efficacy

Construct	Path Coefficient	T-statistic
Support Usefulness	0.4449	14.2420
Systems Development Self Efficacy	0.1408	1.3914
Systems Development Self-Efficacy * Support Usefulness	-0.0019	0.5274

As shown in Table 4.15 and in Figure 4.1, all hypotheses are supported. As hypothesized in Chapter 3, perceived usefulness, perceived ease of use and compatibility have a significant positive effect on attitude. Inspection of the path coefficients reveals the relative importance of the independent variables for explaining attitude (Chin & Gopal, 1995). In terms of the relative importance of explaining attitude, perceived usefulness is most important, followed by compatibility and perceived ease of use. The proposed research model explains 72% percent—the r-square value[8]--of attitude.

Examination of the path coefficients also reveals the relative importance of the dimensions for explaining the structure of perceived usefulness (Chin & Gopal, 1995). As indicated by the stronger path coefficient, rational usefulness is a more important component of perceived usefulness than political usefulness. Further examination of the path coefficients reveals that product usefulness is the most important component of rational usefulness followed by process usefulness and communication usefulness. Support usefulness is the most important component of political usefulness, as indicated by the highest path coefficient, followed by symbolic usefulness, defense usefulness and career usefulness. Furthermore, as hypothesized, perceived usefulness, perceived ease of use and compatibility all have a significant positive effect on perceived usefulness. A detailed discussion of these results is presented in Chapter 5.

[8] PLS does not provide an adjusted r-square statistic

Table 4.15 Hypotheses Testing

Hypothesis	Path Coefficient	T-statistic	P-Value
H1: Higher levels of perceived usefulness of a formalized commercial methodology will lead to higher levels of positive developer attitude toward using a formalized commercial methodology.	0.3981	4.0045	p < .0005
H2: Rational usefulness will be a significant component of perceived usefulness.	0.63	24.75	p < .0005
H2a: Product usefulness will be a significant component of rational usefulness.	0.3984	23.6565	p < .0005
H2b: Process usefulness will be a significant component of rational usefulness.	0.3898	23.9429	p < .0005
H2c: Communication usefulness will be a significant component of rational usefulness.	0.3159	18.5185	p < .0005
H3: Political usefulness will be a significant component of perceived usefulness.	0.4277	18.0359	p < .0005
H3a: Career usefulness will be a significant component of political usefulness.	0.2062	7.2374	p < .0005
H3b: Symbolic usefulness will be a significant component of political usefulness.	0.4194	12.4566	p < .0005
H3c: Support usefulness will be a significant component of political usefulness.	0.4697	12.9078	p < .0005
H3d: Defense usefulness will be a significant component of political usefulness.	0.2128	7.1384	p < .0005
H4a: Higher levels of perceived ease of use of a formalized commercial methodology will lead to higher levels of positive developer attitude toward using a formalized commercial methodology.	0.2529	4.1779	p < .0005
H4b: Higher levels of perceived ease of use of a formalized commercial methodology will lead to higher levels of perceived usefulness of a formalized commercial methodology.	0.2734	3.1732	p < .001
H5a: Higher levels of compatibility with existing software development processes will lead to higher levels of positive developer attitude toward using a formalized commercial methodology.	0.3433	4.011	p < .0005
H5b: Higher levels of compatibility with existing software development processes will lead to higher levels of perceived usefulness of a formalized commercial methodology.	0.5443	7.0537	p < .0005

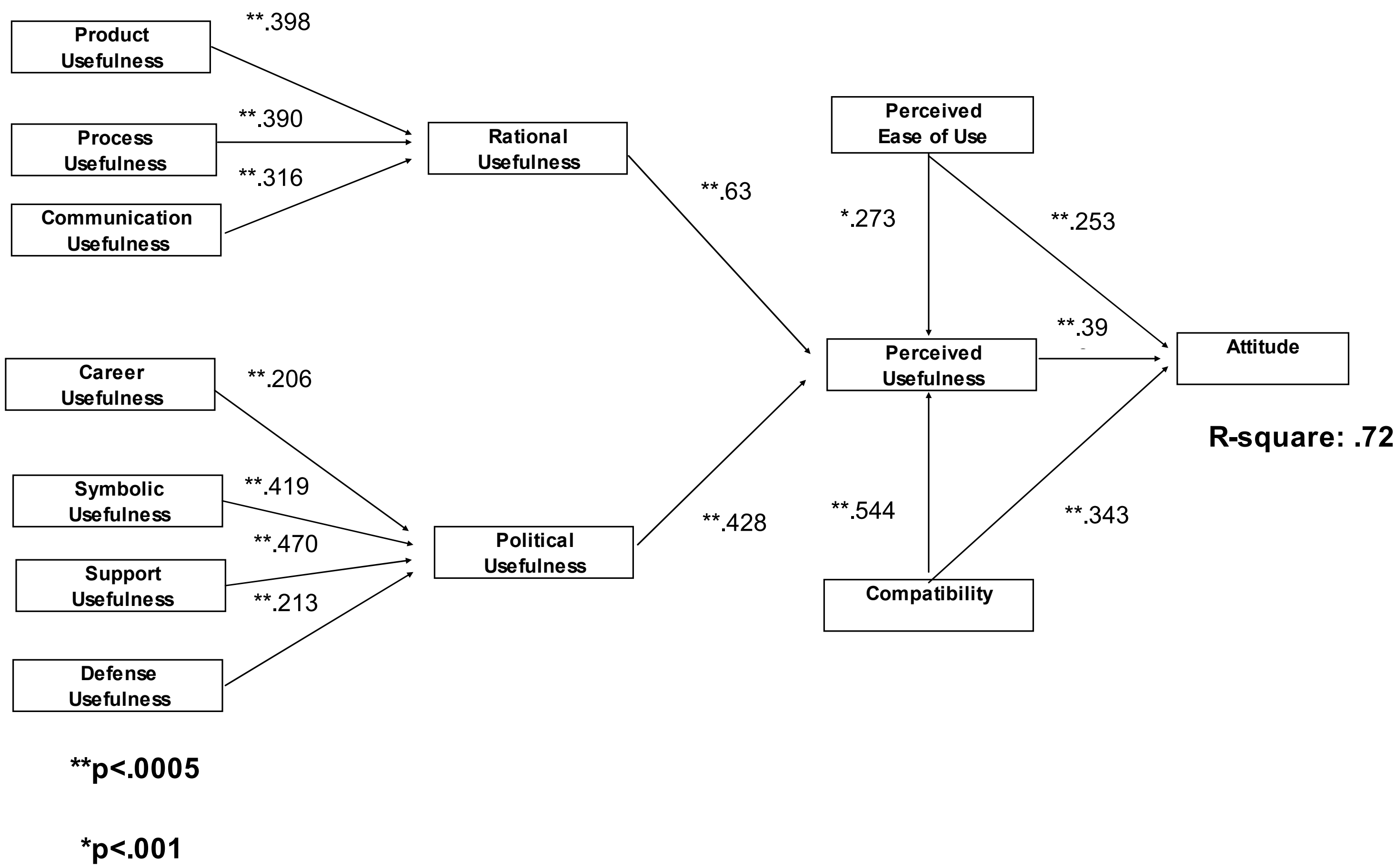

Figure 4.1 Research Model

Comparison to the Perceived Usefulness construct from the TAM

A comparison of the proposed research model to a competing research model composed of the perceived usefulness items from the Technology Acceptance Model (TAM; Davis, 1989) was also conducted. All scales show composite reliability and Cronbach's alpha above .7, indicating strong reliability. Table 4.16 shows Cronbach's alpha and the composite reliability measure for each construct in the competing TAM model.

Table 4.16 Measures of Internal Consistency

Scale	No. of Items	Cronbach's Alpha	Composite Reliability
Perceived Ease of Use	6	.87	.90
Attitude	4	.84	.89
Perceived Usefulness	6	.91	.93

As shown in Table 4.17, all items for each construct in the competing TAM model—perceived usefulness, perceived ease of use, and attitude--load above .707 (Agarwal & Karahanna, 2000; Benbasat & Bassellier, 2004) and have t-statistics for their outer loadings greater than 1.96, thereby indicating strong convergent validity (Gefen & Straub, 2005).

Table 4.17 Item Loadings

	ATT	PEOU	PU
ATT1	**0.87**	0.53	0.71
ATT10	**0.75**	0.50	0.57
ATT5	**0.82**	0.61	0.68
ATT7	**0.86**	0.51	0.72
PEOU1	0.30	**0.73**	0.16
PEOU3	0.50	**0.76**	0.30
PEOU4	0.34	**0.78**	0.25
PEOU6	0.49	**0.80**	0.39
PEOU7	0.61	**0.76**	0.53
PEOU8	0.59	**0.82**	0.51
PU1	0.72	0.49	**0.88**
PU2	0.61	0.39	**0.84**
PU3	0.63	0.30	**0.79**
PU4	0.76	0.43	**0.84**
PU5	0.68	0.46	**0.85**
PU6	0.65	0.42	**0.80**

As shown in Table 4.18, the AVE for perceived usefulness, perceived ease of use, and attitude all exceed .5, thereby providing further evidence of strong convergent validity (Chin, 1998b; Fornell & Larker, 1981).

Table 4.18 AVE

Scale	AVE
Perceived Usefulness	.69
Perceived Ease of Use	.60
Attitude	.69

As shown in Table 4.17, all items loaded more strongly on their own construct than any other construct, providing evidence of discriminant validity. Furthermore, as shown in Table 4.19, the latent variable correlations for each construct are less than the square root of the AVE for each construct, indicating strong discriminant validity.

Table 4.19 AVE and Latent Variable Correlations

	ATT	PEOU	PU
ATT	1.00		
PEOU	0.65	1.00	
PU	0.81	0.51	1.00
AVE	0.69	0.6	0.69
Sq(AVE)	0.83	0.77	0.83
Pass/Fail	Pass	Pass	Pass

As shown in Table 4.20 and Figure 4.2, perceived usefulness and perceived ease of use both have a significant positive effect on attitude. Furthermore, perceived ease of use positively affects perceived usefulness.

Table 4.20 Hypothesis Testing

Hypothesis	Path Coefficient	T-statistic	P-Value
Perceived usefulness will have a positive effect on attitude.	0.6531	13.7092	p < .0005
Perceived ease of use will have a positive effect on attitude.	0.32	5.8701	p < .0005
Perceived ease of use will have a positive effect on perceived usefulness.	.5158	8.6359	p < .0005

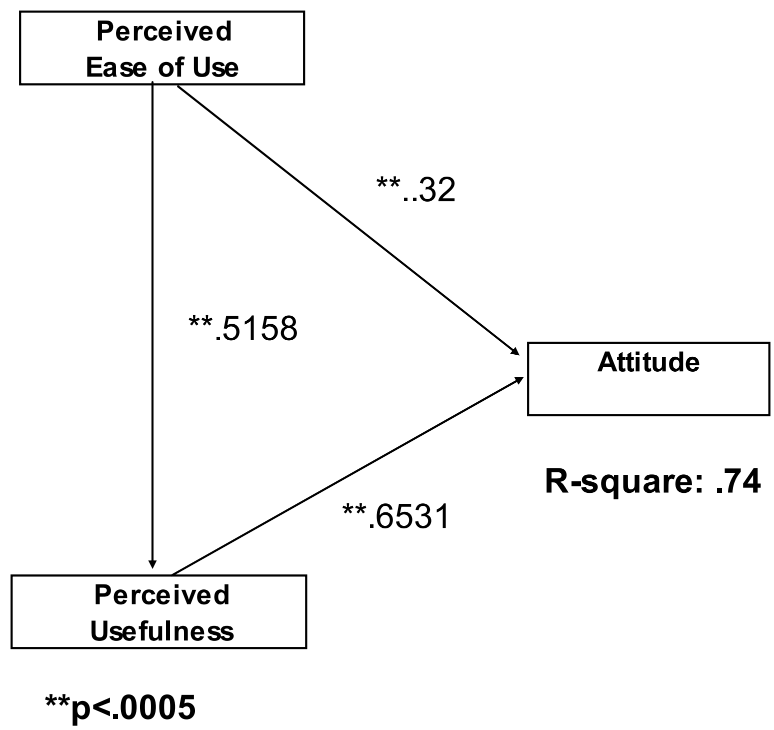

Figure 4.2 Competing Model with TAM Perceived Usefulness Construct

The competing TAM model explains 2% more variance in attitude—r-square of 74%--than the proposed research model. Furthermore, the path coefficient for the perceived usefulness construct from the competing TAM model is stronger--.6531--than the path coefficient for the perceived usefulness construct from the proposed research model--.398. The implications of these findings are discussed further in Chapter 5.

Chapter Five

Discussion

This chapter provides a discussion and interpretation of the results of this study. This study investigated the determinants of developer attitude toward using a formalized commercial methodology and explored the underlying factors of perceived usefulness for a methodology. This research fills a gap in the literature as prior research has not attempted to explain developer *attitude* toward using a formalized *commercial* methodology. Furthermore, prior research has not attempted to uncover and subsequently operationalize the underlying dimensions of perceived usefulness for a methodology.

This chapter is organized as follows: the first section discusses significant findings and the second section draws comparisons between the proposed research model and a competing research model composed of the perceived usefulness construct and its associated items from the Technology Acceptance Model (TAM; Davis, 1989). The final section explores the overall implications of this research. Contributions, limitations, and recommendations for future research are discussed in Chapter 6.

Significant Findings

Overall, the findings of this study support the hypotheses developed in Chapter 2. The findings are discussed separately in this section, before overall implications of the research are presented.

Explaining developer attitude

Methodology use is becoming increasingly mandatory due to Capability Maturity Model Integration (CMMi) requirements (Duggan, 2004; Riemenschneider et al., 2002) and client mandates (Fitzgerald, 1996, 1998a). Methodologies also have high levels of user inter-dependencies (Fichman, 1992), and tend to be more mandated by upper-level management than tools (Hardgrave et al., 2003). Combined, these factors suggest that methodology use may be highly mandatory. In highly mandatory use situations, the dependent variable of interest switches from *intention to use* to *attitude* toward use (Brown et al., 2002). Consistent with these recommendations, this study investigated developer *attitude* toward using a formalized commercial methodology, rather than developer intention to use a formalized commercial methodology.

Voluntariness has a mean of 3.87, in between “Slightly Disagree” and “Undecided”. Given that the mean for voluntariness is between “Slightly Disagree” and “Undecided”, it appears participants do not believe that methodology use is voluntary, thereby implying that participants believe methodology use is more mandatory than voluntary. Even though the mean for voluntariness was not “Strongly Disagree”, *attitude*, rather than *intention to use*, still constitutes the more relevant dependent variable on a conceptual level given the high user inter-dependencies for a methodology (Fichman,

1992) and the pressures toward increased formalism during systems development (Fitzgerald et al., 2002).

An examination of prior research suggests that developers exhibit a negative attitude toward using a methodology. For example, developers perceive the influence of a formalized methodology as a minor factor in the success of a development project (Fitzgerald, 1998a). Investigating developer attitude constitutes a worthwhile research endeavor because negative attitude can hinder the capability of the methodology to achieve rational goals. For instance, prior research finds employees who do not wholeheartedly accept an innovation can impede its implementation (Markus, 1983; Leonard-Barton, 1988; Zuboff, 1988), thereby leading to sabotage and unfaithful use (Markus, 1983; Leonard-Barton, 1988; Zuboff, 1988). Overall, the problems associated with negative attitude further justify its choice as the dependent variable of interest.

As shown in Figure 4.1 in Chapter 4, the proposed research model explains 72% (r-square) of the variance in attitude. Prior research using the Decomposed Theory of Planned Behavior (DTPB) in a technology, not methodology, adoption context explains 76% of attitude (Taylor & Todd, 1995). Additionally, prior research explains 63% (Hardgrave & Johnson, 2003) and 64% (Hardgrave et al., 2003) of developer intention to use a methodology. In sum, it appears that the amount of variance explained in this study is consistent with prior research.

Consistent with the theoretical predictions of the DTPB, results indicate that developer attitude toward using a formalized commercial methodology is explained by perceived usefulness, compatibility and perceived ease of use. The significance of these independent variables reaffirms the theoretical predications of the DTPB (Taylor & Todd, 1995).

As indicated by its high path coefficient, perceived usefulness is the most important independent variable for explaining attitude (beta=.398, p<.0005), followed by compatibility (beta=.343, p<.0005) and perceived ease of use (beta=.253, p<.0005). These findings suggest managers can foster positive developer attitudes toward using a formalized commercial methodology use by discussing how the methodology can be useful, how it is consistent with existing systems development practices, and its ease of use.

Perceived usefulness

Perceived usefulness was hypothesized to have a positive influence on developer attitude toward using a formalized commercial methodology. Perceived usefulness is an important variable for explaining an individual's attitude toward using an innovation in a voluntary setting (Taylor & Todd, 1995), in a mandatory setting (Brown et al., 2002) and for explaining developer intention to use a methodology (Hardgrave et al., 2003; Riemenschneider et al., 2002). As shown in Table 4.15 in Chapter 4, consistent with these findings, the significant path coefficient for perceived usefulness indicates that perceived usefulness positively influences developer attitude. These results imply that developers have more favorable attitudes toward methodologies they perceive as useful.

Examination of the relative strength of the path coefficients indicates that perceived usefulness is the most salient independent variable for explaining attitude. This finding is consistent with other studies investigating methodology adoption (Hardgrave et al., 2003; Riemenschneider et al., 2002) and studies investigating technology adoption

(Davis, 1989; Venkatesh et al., 2003). Consequently, when attempting to encourage positive attitudes toward using formalized commercial methodologies, training programs should focus on how a methodology can be useful.

Rational usefulness

In order to better understand what makes a methodology useful to developers, perceived usefulness was broken-down into rational usefulness and political usefulness. Rational usefulness was hypothesized to be a significant component of perceived usefulness. As shown in Table 4.15 in Chapter 4, the significant path coefficient for rational usefulness suggest that it is a salient factor in what makes a methodology useful to developers.

The significance of rational usefulness supports the arguments of Robey and Markus (1984) who contend that a methodology can be viewed as a rational process. More specifically, the significant path coefficient for rational usefulness suggests that a methodology can be useful for achieving objectives that are agreed upon by stakeholders on the systems development effort, such as improving software quality, increasing the productivity of the development process, and enhancing communication with users and among members of the development team. Furthermore, this finding reaffirms case study research suggesting that methodologies play several rational roles related to improving the productivity of the development process and raising system quality (Fitzgerald, 1998b, Fitzgerald et al., 2002).

Rational usefulness (beta=.63) also has a larger path coefficient than political usefulness (beta=.428). The stronger path coefficient for rational usefulness implies that methodologies may be more useful for achieving rational objectives, such as improving software quality, than for achieving political objectives. Overall, the significance of rational usefulness suggests that methodologies can be useful to developers because they improve software quality, increase the productivity of the development process and enhance communication with users and other members of the development team.

Dimensions of rational usefulness

Rational usefulness was further decomposed into three referent dimensions: product usefulness, process usefulness, and communication usefulness. Each of these three dimensions was hypothesized to be a significant component of rational usefulness. As shown in Table 4.15 in Chapter 4, the significant path coefficients for these three dimensions indicate that they are all salient components of rational usefulness. As indicated the relative strength of the path coefficients, results further suggest that product usefulness (beta=.398) is the most salient component of rational usefulness, followed by process usefulness (beta=.390) and communication usefulness (beta=.316). These results imply that developers believe a methodology is useful, from a rational standpoint, for increasing system quality, but is also useful for improving the productivity of the development process and enhancing communication with users and among members of the development team.

The significance of product usefulness is not surprising. Prior research finds a primary belief developers associate with a methodology is the degree to which it improves the quality of the developed system (Huisman & Iivari, 2006; Johnson et al., 1999). Consistent with these findings, results indicate that developers believe product usefulness is a significant factor in what makes a methodology useful. Thus, to foster positive developer attitude toward using a formalized commercial methodology, training programs should emphasize how using a methodology can improve software quality.

Although the results of this study suggest that product usefulness is a significant factor in what makes a methodology useful, Khalifa and Verner (2000) find that product quality, conceptually similar to product usefulness, does not have a significant effect on methodology use. One possible reason for this discrepancy is that product usefulness in this study was hypothesized to be a component of rational usefulness, but was not hypothesized to have a direct effect on attitude. A post-hoc analysis indicates that product usefulness has a significant positive influence directly on attitude (beta=.1847, t-statistic=2.044, $p < .025$), suggesting that another reason may exist for this inconsistency.

A second possible reason for this difference is that this study investigated developer attitude, whereas Khalifa and Verner (2000) investigated methodology usage. Perhaps the difference in dependent variables investigated accounts for the discrepancy in results.

A third possible reason for the difference in significance is that Khalifa and Verner (2000) did not use psychometrically validated scales of product quality, which could have resulted in measurement error (Riemenschneider et al., 2002). The scales used in this study were subjected to a rigorous development and evaluation procedure, which could have resulted in better measurement of product usefulness. Future research should further explore the significance of product usefulness in more detail.

Results also indicate that process usefulness is a salient component of rational usefulness, as indicated by its significant path coefficient. This result implies that developers believe a methodology can be useful for improving the productivity of the development process. The significance of process usefulness is also consistent with findings from prior empirical research (Huisman & Iivari, 2006; Johnson et al., 1999; Khalifa & Verner, 2000). Additionally, prior case study research finds that methodologies play several rational roles related to increasing the productivity of the development process, such as the facilitation of project management and control of the development process (Fitzgerald, 1998b; Fitzgerald et al., 2002). The significance of process usefulness is congruent with these findings and further substantiates their importance by using a survey, rather than case study. Thus, training programs may be able to create more favorable developer attitudes by explaining how a methodology can improve the productivity of the development process.

Similar to product and process usefulness, the significant path coefficient for communication usefulness indicates that it is also a salient factor in what makes a methodology useful to developers. The significance of communication usefulness implies that developers believe a methodology is useful for improving communication with users and among members of the development team. These results provide support for the exploratory factor analysis conducted by Johnson et al. (1999) which finds that communication usefulness is a significant factor of perceived usefulness for a methodology. Furthermore, the importance of communication usefulness corroborates case study research indicating that developers believe following a standardized process can increase communication and coordination among various stakeholders on a software development project (Fitzgerald, 1998b; Fitzgerald et al., 2002).

Political usefulness

Robey and Markus (1984) maintain that a methodology may be useful from a political standpoint. Accordingly, political usefulness was hypothesized to be a significant component of perceived usefulness for a methodology. As illustrated in Table 4.15 in Chapter 4, the significant path coefficient (beta=.428, $p<.0005$) for political usefulness suggests that political usefulness is a salient factor in what makes a methodology useful to developers.

The significance of political usefulness implies that methodologies may not only be useful from a rational perspective, but that they might also be useful from a political perspective. More specifically, this finding indicates that methodologies may be able to help one group or individual achieve its own objectives to the relative disadvantage of other groups or individuals. The significance of political usefulness empirically reaffirms the arguments of Robey and Markus (1984) as well as findings from case study research indicating that methodologies play several political roles (Fitzgerald, 1998b; Fitzgerald et al., 2002). Although political usefulness is a weaker factor in what makes a methodology useful, as indicated by its lower path coefficient, the significance of political usefulness is concerning as the political roles played by methodologies do not form a suitable base for sustained methodology use and may lead developers to use a methodology inappropriately (Fitzgerald et al., 2002).

Dimensions of political usefulness

Political usefulness was decomposed into four referent dimensions: career usefulness, symbolic usefulness, support usefulness and defense usefulness. Each of these four dimensions was hypothesized to be a significant component of rational usefulness.

Inspection of the path coefficients listed in Table 4.15 in Chapter 4 indicates all four dimensions are significant components of political usefulness. Results further suggest that support usefulness (beta=.470, p<.0005) is the most important component of political usefulness, followed by symbolic usefulness (beta=.419, p<.0005), defense usefulness (beta=.213, p<.0005) and career usefulness (beta=.206, p<.0005). These findings imply that developers believe a methodology is useful, from a political standpoint, for reducing developer anxiety, for showing others that professional development practices are being used, for defending against unreasonable user demands and for enhancing career opportunities.

The significant path coefficient from support usefulness to political usefulness implies that methodologies may be useful for reducing developer anxiety. Systems development can be a stressful process (Wastell & Newman, 1993); methodology use is one way in which developers can cope with the stress of systems development (Lee & Truex, 2000). On the surface, support usefulness may be seen as positive benefit of methodology use because using a methodology may lower developer stress. However, developers who use methodologies to reduce stress and anxiety may be prone to goal displacement (Fitzgerald 1998b; Fitzgerald et al., 2002). Goal displacement can result in developers becoming so obsessed with following the methodology that they may perform unnecessary steps during systems development, thereby slowing the productivity of the development process (Fitzgerald et al., 2002).

The significance of symbolic usefulness suggests that developers believe a methodology is useful for showing others that professional development practices are being used. Case study research finds that methodologies are useful for showing others—

users and upper-level management—that professional systems development processes are being used (Fitzgerald, 1998b; Fitzgerald et al., 2002; Kautz et al., 2004; Nandhakumar & Avison, 1999; Robey & Markus, 1984). The significance of support usefulness reaffirms these findings and extends their generalizability to a wider audience of developers.

On the surface, using a methodology for its symbolic usefulness seems benign. However, when developers use a methodology in this manner, they often use the methodology superficially and retrofit documentation to make it appear that a methodology has been used (Fitzgerald et al., 2002). Thus, when used merely as a symbol of professionalism, the capability of the methodology to achieve rational goals, such as improving software quality, may not be realized (Fitzgerald et al., 2002).

The results of this study also indicate that defense usefulness is a significant component of political usefulness, implying that developers believe a methodology is useful for defending against unreasonable user demands. Prior case study research finds that methodologies can insulate developers from unreasonable user demands (Fitzgerald et al., 2002; Kautz et al., 2004). By using a survey, this study increases the generalizability of these findings from case study research to a wider audience of developers.

Used in a defensive manner, however, developers may use a methodology to obstruct additional user requirements (Wastell, 1996). Using a methodology in this manner may be detrimental to user-developer relations (Wastell, 1996). The results of this study should motivate upper-level managers to be wary of developers who use methodologies in such a manner.

Prior research provides conflicting evidence regarding the significance of career usefulness on developer attitude toward a methodology. Riemenschneider et al. (2002) find career consequences, conceptually similar to career usefulness, is not significantly related to developer intention to use a formalized in-house methodology. On the contrary, Johnson et al. (1999) find that career usefulness is a factor of perceived usefulness for a formalized commercial methodology. The significance of career usefulness supports the research by Johnson et al. (1999) and therefore suggests that a commercial methodology may be useful to developers because it enhances career opportunities.

Similar to the other dimensions of political usefulness, using a methodology to gain greater career opportunities may be initially perceived as a positive benefit. However, it is possible for developers to use a methodology to achieve career advancement at the expense of achieving rational goals (Robey & Markus, 1984). For example, the quest for greater career opportunities could motivate developers to use the Rational Unified Process (RUP), rather than an in-house methodology. Thus, developers may recommend using the RUP because learning the RUP is valuable for career advancement, even though it may not result in the most efficient development process (Robey & Markus, 1984).

Perceived ease of use

As shown in Table 4.15 in Chapter 4, results indicate that perceived ease of use has a significant positive effect on attitude (beta=.253, $p<.0005$), implying that developers have a more favorable attitude toward a methodology if they perceive it as easy to use.

The positive effect of perceived ease of use on attitude is somewhat contradictory to prior research as Riemenschneider et al., (2002) find complexity, conceptually similar to perceived ease of use, is not a significant predictor of developer intention to use a methodology. Two possible reasons exist for this discrepancy. First, the methodology investigated in Riemenschneider et al. (2002) was a formalized *in-house* methodology, whereas this study investigated developer attitude toward using a formalized *commercial* methodology. Thus, it is possible that perceived ease of use may have had a different impact due to the different methodologies investigated.

The other possibility is that Riemenschneider et al. (2002) investigated *intention to use* a methodology, whereas this study investigated developer *attitude* toward using a methodology. Perhaps the difference in the dependent variable contributed to the difference in the significance of perceived ease of use. Future research should explore the relevance of perceived ease of use on attitude and intention to use in more detail.

In addition to positively influencing attitude, perceived ease of use also positively influences perceived usefulness (beta=.273, $p<.001$). This finding is consistent with prior research investigating developer intention to use a methodology (Hardgrave et al., 2003) and implies that developers may perceive a methodology as more useful if it is easier to use.

Compatibility

Compatibility was hypothesized to positively influence attitude and perceived usefulness. As indicated by the significant path coefficients listed in Table 4.15 in Chapter 4, compatibility positively influences attitude (beta=.343, $p<.0005$) and perceived ease of use (beta=.544, $p<.0005$).

The significant positive influence of compatibility on attitude suggests developers will have more favorable attitudes toward methodologies they perceive as consistent with the way they like to develop systems. This result corroborates prior research on developer intention to use a methodology (Riemenschneider et al., 2002) and extends the findings of Hardgrave et al. (2003) and Riemenschneider et al. (2002) to a different dependent variable—*attitude*, rather than *intention to use*—and to a formalized *commercial* methodology, rather than an *in-house* methodology.

Although this study finds compatibility positively influences attitude, an analysis of prior research yields conflicting results regarding the significance of compatibility. While theory holds that compatibility positively affects attitude (Taylor & Todd, 1995), research on tool adoption finds that the direct effect of facilitating conditions, conceptually similar to compatibility, on intention to use becomes non-significant when performance expectancy (perceived usefulness) and effort expectancy (perceived ease of use) are included in the theoretical model (Venkatesh et al., 2003).

One possible reason for the difference in the significance of compatibility between a tool adoption context and a methodology adoption context is the degree of change required by a methodology. The adoption of a methodology constitutes a more radical change than the adoption of software development tools and techniques (Orlikowski, 1993; Roberts et al., 1998). Thus, compatibility is more important in methodology adoption than tool adoption due to the greater magnitude of change required by a methodology (Hardgrave et al., 2003; Riemenschneider et al., 2002). The significance of compatibility in this study provides further evidence for this rationale.

The positive effect of compatibility on perceived usefulness substantiates prior research (Chau & Hu, 2001; Hardgrave et al., 2003; Oh et al., 2003; Templeton & Byrd, 2003). This finding implies that developers may perceive a methodology as more useful if it is consistent with existing systems development practices.

Comparison to TAM

An additional analysis was conducted to determine whether the proposed research model explains more variance than a competing model composed of perceived usefulness and its associated items from the TAM (Davis, 1989). Prior research on methodology adoption has defined perceived usefulness as the degree to which using a methodology improves a developer's job performance (Hardgrave et al., 2003). If, however, methodologies provide additional benefits to developers, then limiting the definition of perceived usefulness to beliefs surrounding increased job performance may provide an incomplete representation of what makes a methodology useful to developers. By including political usefulness and its referent dimensions as components of perceived usefulness, it was hypothesized that an expanded perceived usefulness construct would explain more variance than a competing model composed of the perceived usefulness construct and its associated items from the TAM.

As shown in Table 4.20 and in Figure 4.2 in Chapter 4, the competing TAM model explains 2% more variance (r-square) than the proposed research model. Furthermore, the path coefficient for the perceived usefulness construct from the TAM model is stronger--.6531--than the path coefficient for the perceived usefulness construct from the proposed research model--.398.

Measurement error constitutes one possible reason why the competing TAM model explains more variance in attitude than the proposed research model. Most of the items operationalizing the perceived usefulness construct in the TAM are at the individual level and include specific references to "I" or "my" or "me"; for example, "Using this methodology improves the quality of the work I do". On the contrary, most of the items operationalizing the dimensions of perceived usefulness in the research model are purposely phrased at a generic level; for example "Using the Rational Unified Process provides protection from unreasonable demands."

The items operationalizing the dimensions of perceived usefulness were phrased at a generic level for two reasons. First, the items were rewritten to circumvent potential issues with discriminant and convergent validity among the seven dimensions of perceived usefulness. During the first round of card sorting, the judges sorted the items into "personal" and "team" categories, rather than into categories reflecting the dimensions of perceived usefulness. To mitigate this problem, items in product, process, communication, symbolic and defense usefulness were reworded to a generic level. (i.e., removed references to "my", "I", and "me"). Furthermore, during the second round of card sorting, career usefulness and support usefulness failed to exhibit discriminant validity. A possible reason for the lack of discrminant validity was that the items in both of these constructs were phrased at the individual developer level. Thus, the items for support usefulness were rewritten to a generic level. For example, "Using this methodology reduces my anxiety" was reworded to "Using this methodology reduces developer anxiety".

Second, the items were also rephrased to a generic level in an effort to reduce social desirability bias. It is possible that subjects may believe that using a methodology for its symbolic, support, or defense usefulness is socially undesirable. By wording the items at a general level, it was hoped that the effects of social desirability bias could be mitigated.

In research, trade-offs and compromises sometimes have to be made. In this study, the decision was made to write the items used to operationalize the dimensions of perceived usefulness at a generic, rather than individual developer, level. Although this may have reduced social desirability bias and improved the convergent and discriminant validity of the items used to operationalize the dimensions of perceived usefulness, it may have also resulted in less variance explained in attitude than the competing TAM model. Future research should explore these issues in more detail.

Implications

Political usefulness of methodologies

The significance of political usefulness suggests that methodologies can be useful from a political perspective. More specifically, this result indicates that a methodology can be useful for advancing objectives specific to one person or individual to the relative disadvantage of other stakeholders on the development project.

The significance of political usefulness reaffirms the importance of the political roles played by methodologies found in prior case study research (Fitzgerald, 1998b; Fitzgerald et al., 2002). Findings from case study research suffer from low external validity and may be subject to researcher bias (Whitley, 2002). By using a survey, this study extends the findings regarding the political roles played by methodologies to a wider audience of developers. Additionally, the dimensions of perceived usefulness related to the political roles played by methodologies are not widely acknowledged in Information Systems Development research (Fitzgerald et al., 2002) and have never been operationalized or included in a theoretical model. By extracting, defining, and operationalizing these dimensions, this study provides the basis for further empirical research.

Although rational usefulness is a stronger component of perceived usefulness, the significance of the political usefulness and its referent dimensions is troubling because facets of perceived usefulness related to the political roles played by methodologies form a less worthy rationale for methodology use (Fitzgerald, 1994). For example, research suggests that when developers use a methodology to simply demonstrate that systematic development processes are being followed, they may use it in a superficial manner (Fitzgerald, 1998b; Kautz et al., 2004; Nandhakumar & Avison, 1999). Thus, developers may retrofit documentation to make it appear that the methodology has been used (Fitzgerald et al., 2002). Furthermore, when developers use a methodology to alleviate the stress and anxiety resulting from systems development—i.e., support usefulness--they can be prone to goal displacement (Fitzgerald et al., 2002; Wastell, 1996; 1999). By blindly following the methodology, developers may perform unnecessary tasks, thereby slowing the productivity of the development process.

As a final example, using a methodology from a defensive standpoint may result in the disadvantage of users. Prior case study research indicates that methodologies can be used to obstruct user requests for additional system functionality (Wastell, 1996). The significance of defense usefulness implies that developers find a methodology useful for blocking unreasonable user demands—a scenario that could lead to contentious developers-user relations.

Expanding perceived usefulness

Although the competing TAM model explains more variance than the proposed research model, it is hoped that the results of this study encourage researchers to explore the perceived usefulness construct in more detail. The majority of Information Systems research has defined perceived usefulness from a job performance standpoint and has adopted the TAM definition of perceived usefulness—"The degree to which using a technology improves job performance". However, it is clear from the results of this study that a methodology may be useful in ways beyond improving job performance.

Moore & Benbasat (1991) comment that relative advantage, conceptually similar to perceived usefulness, has not been well measured or well explicated in prior research and has become a "garbage can" construct. They further argue that the perceived usefulness construct used in the TAM may be too broadly based. Essentially, methodologies as well as other technologies may be "useful" in many ways. These comments support the basic premise of this study and suggest that the perceived usefulness construct should be explored in greater depth. Overall, although the competing TAM model explains more variance than the proposed research model, expanding and subsequently decomposing the perceived usefulness construct provides a better understanding of what makes a methodology "useful."

Additionally, breaking-out the perceived usefulness into its referent dimensions makes the perceived usefulness construct more managerially relevant. When attempting to engender positive attitudes toward using methodologies, training programs must provide specific ways in which a methodology may be useful. The results of this study suggest that positive attitudes toward methodologies may be facilitated by discussing how methodologies improve the quality of the software (product usefulness), the productivity of the development process (process usefulness) and enhance communication with end users and other members of the development team (communication usefulness).

Comparing and contrasting methodologies

Although prior research has investigated developer intention to use a formalized *in-house* methodology (Hardgrave et al., 2003; Riemenschneider et al., 2002), research has not investigated developer intention to use or attitude toward using a formalized *commercial* methodology. A formalized *commercial* methodology may provide greater career opportunities than a formalized *in-house* methodology because it is used across organizations. Consistent with this reasoning, the significance of career usefulness in this study indicates that developers believe a formalized *commercial* methodology is useful because it affords greater career opportunities.

Although prior research finds that perceived ease of use does not have a significant effect on developer intention to use a methodology (Riemenschneider et al., 2002), results indicate that perceived ease of use positively influences developer attitude toward using a formalized commercial methodology.

The differences in the significance of perceived ease of use and career usefulness provide evidence that developers may perceive commercial and in-house methodologies differently. As a result, researchers should not treat methodologies as homogeneous phenomena (Huisman & Iivari, 2006) and therefore should compare and contrast models of methodology adoption and developer attitude for different methodologies.

Chapter Six

Conclusion

This study investigated developer attitude toward using a formalized commercial methodology. This research also attempted to uncover what makes a formalized commercial methodology useful to developers. The findings indicate that developer attitude is positively influenced by perceived usefulness, compatibility and perceived ease of use. Results further suggest that perceived usefulness is composed of rational usefulness and political usefulness. Rational usefulness, in turn, is composed of product usefulness, process usefulness and communication usefulness; political usefulness is composed of career usefulness, symbolic usefulness, support usefulness and defense usefulness.

Contributions to Information Systems Research

The results of this research make several important contributions to Information Systems research. First, by expanding the perceived usefulness construct to include aspects of perceived usefulness related to the perspective of a methodology as a political process, this research provides a deeper understanding of what makes a methodology useful to developers. Results indicate that methodologies are useful not only for achieving objectives related to the rational perspective, such as increasing system quality, they are also useful for achieving objectives related to the political perspective, such as enhancing career opportunities. These findings should encourage researchers to explore the perceived usefulness construct in more depth for other technologies.

The significance of political usefulness and its referent dimensions also highlight the importance of the political roles played by methodologies. The dimensions of perceived usefulness related to the political roles played by methodologies are not well recognized in Information Systems Development research (Fitzgerald et al., 2002) and have never been operationalized or included in a theoretical model. By defining and operationalizing these dimensions and including them in nomological network, this research supplies the basis for further empirical research on the political roles played by methodologies.

This research also broke-down perceived usefulness into its referent dimensions in order to better understand what makes a methodology useful to developers. By decomposing perceived usefulness, the results of this study illustrate the relative importance of each dimension of perceived usefulness. Results indicate that while rational usefulness is a stronger factor in what makes a methodology useful to developers, political usefulness does play a part in what makes a methodology useful. These results are disconcerting, given that facets of perceived usefulness related to the political roles played by methodologies may lead developers to use a methodology inappropriately (Fitzgerald et al., 2002)

Although prior research finds that career consequences is not a significant explanatory variable for explaining developer intention to use a formalized *in-house* methodology, the results of this study illustrate that career usefulness may be an important aspect of what makes a formalized *commercial* methodology useful to developers. Furthermore, the significance of perceived ease of use also runs contrary to prior research on developer intention to use a methodology (Riemenschneider et al., 2002). These findings underscore the point that researchers should not treat methodologies as homogeneous phenomena (Huisman & Iivari, 2006; Khalifa & Verner, 2000). It is hoped that these results encourage researchers to compare and contrast adoption models and models of developer attitude for different methodologies.

Contributions to Practice

In addition to providing important contributions to research, this study provides several important contributions to practice. Given that methodology use is often mandatory, this study investigated a more relevant dependent variable than intention to use—developer attitude toward using a formalized commercial methodology. Identifying the salient antecedents of developer attitude toward using a methodology constitutes an important research endeavor, given the high resistance to methodologies from developers (Kozar, 1989) and the potential for negative developer attitude to impede the capability of the methodology to achieve rational goals.

Results indicate that perceived usefulness, compatibility and perceived ease of use all positively influence developer attitude toward using a formalized commercial methodology. Training programs, thus, should emphasize how a methodology can be useful to developers, how it is compatible with existing systems development practices and its ease of use in order to foster positive attitudes toward methodology use.

With respect to the dimensions of perceived usefulness, results suggest that the best way to engender positive developer attitude toward methodology use would be to discuss how the methodology will improve the quality of the developed system. Training programs should also discuss how the methodology will improve the productivity of the development process and enhance communication. Although political usefulness and its referent dimensions are significant aspects of what makes a methodology useful, it may be unwise for training programs to frame these aspects as positive benefits of methodology use. Factors of perceived usefulness related to the political roles played by methodologies may not be the "right" reasons for using a methodology (Fitzgerald et al., 2002). Rather, the significance of the political usefulness construct and its referent dimension should serve as a warning sign.

Limitations

In social science research, choices regarding the design of the study must be made. These choices may impose certain limitations on the study. The findings of this study must be judged within the context of its limitations, as discussed below.

Exploratory constructs

The primary limitation of this study revolves around its exploratory nature. The symbolic, support, and defense dimensions of usefulness are new constructs and involve the development of new scales. This research followed the recommendations of Straub (1989), Churchill (1979) and Netemeyer et al. (2003) for construct definition and item development. Care was taken to ensure that the definitions of the constructs were well-grounded in the referent literature.

The construct definitions and associated items for the dimensions of perceived usefulness were subjected to a disciplined development and purification process. The construct definitions were given to developers to evaluate their understandability and appropriateness (Netemeyer et al., 2003). Three rounds of card sorting were subsequently performed with practicing Information Technology professionals to provide preliminary insights into the convergent and discriminant validity of the dimensions of perceived usefulness (Moore & Benbasat, 1991). A pre-test was conducted to ensure that the survey instructions were easy to follow. Finally a pilot study was conducted on 98 Information Technology professionals (Netemeyer et al., 2003). Measures of internal consistency and validity were calculated on the pilot study data; changes were made to the items to enhance the potential of the scales. Overall, a rigorous instrument development process was used to both establish the definitions of the dimensions of perceived usefulness and their associated items.

Sample

A second limitation revolves around the use of a convenience sample. Rather than obtaining a true random sample of developers, this research uses a convenience sample. Furthermore, the final survey was conducted on developers who use the Rational Unified Process; thus, it is not fully known whether the results of this study will generalize to other methodologies. However, given that the Rational Unified Process is a *de facto* industry standard (Ambler, 2005), it seems reasonable to assume that a wide audience of developers may use the Rational Unified Process.

Self-reported data

Another limitation is that this research uses self-reported data. All of the measures for this study are perceived; thus, a limitation of this study is self report biases of participants. Although the measures are perceived, objective measures of usefulness, such as percentage reduction in software defects as a result of using a methodology, may not be as useful for explaining developer attitude toward using a methodology. Even though a methodology may reduce the actual number of software defects, it is possible that developers may still *perceive* that a methodology does not improve software quality. Thus, *perceptions*, rather than objective outcomes, seem more appropriate for explaining developer attitude toward using a methodology.

Response rate

The exact response rate is unknown as individuals receiving the invitation letter containing the hyperlink to the survey could have passed it along to other individuals. It should be noted that non-response bias may have impacted the results.

Recommendations for Future Research

During the progression of this study, numerous additional research questions of interest arose. Addressing some of these questions forms the basis for a fruitful research agenda. This section discusses several avenues for future research.

Compare models of developer attitude for different methodologies

Methodologies may not be uniform phenomena (Huisman & Iivari, 2006; Khalifa & Verner, 2000). For example, the differences between the traditional approach and Agile approaches to software development are large (Nerur, Mahapatra, & Mangalaraj, 2005). As demonstrated by the significant path coefficients for career usefulness and perceived ease of use, the significance of some independent variables for explaining developer attitude toward using a methodology may vary depending upon the methodology investigated. Thus, future research could compare and contrast models of developer attitude for different methodologies.

Compare models of developer attitude for expert and novice developers

A major criticism of methodologies is that they ignore developer-related factors (Fitzgerald, 1998; Fitzgerald et al., 2002). Developer-related factors, however, represent important considerations for methodology design because they influence how a methodology is used. Due to individual differences, such as experience, developers may interpret and hence apply the same methodology differently for a given situation (Fitzgerald et al., 2002; Wynekoop & Russo, 1997).

The level of systems development experience constitutes one of the most important developer-related factors influencing how developers use methodologies (Fitzgerald et al., 2002). Prior research indicates that experienced developers describe methodologies as plans or guides to action, rather than as deterministic rule-sets to be followed rigorously (Kautz et al., 2004; Beynon-Davies & Williams, 2003; Nandhakumar & Avison, 1999; Fitzgerald, 1997). As a result, experienced developers make adaptations to the methodology to accommodate the unique characteristics of the development context. On the other hand, inexperienced developers find methodologies to be useful guides and express a need for methodologies to help them learn the company's development practices (Kautz et al., 2004). Inexperienced developers who use methodologies are more likely to follow methodologies rigorously to relieve their lack of systems development self-efficacy (Fitzgerald, 1997).

Overall, the same methodology may effectively serve the needs of an experienced developer, but fail to make a difference or cause problems for an inexperienced developer (Wynekoop & Russo, 1997). Similarly, an experienced developer might adapt, select or use a methodology differently than an inexperienced developer (Wynekoop & Russo, 1997). These differences suggest that the significance of some independent variables may change based upon a developer's level of systems development experience. Accordingly, future research could use the demographic data collected in this study, such as years of development experience, to compare and contrast models of developer attitude for experienced and inexperienced developers.

Desirable properties of methodologies

A review of the research on methodology use yields only one study (Siau & Tan, 2005) that has attempted to define the characteristics of an effective methodology. Although the study by Siau and Tan (2005) presents an initial attempt to uncover the desirable properties of a methodology-- characteristics of a software development methodology that increase a methodology's effectiveness and practical application--their study consisted of an electronic brainstorming session with a limited sample of practitioners. Additional survey research is needed to examine the views of a wider audience of Information Technology professionals. Furthermore, future research could build a theoretical model of how those desirable properties lead to methodology adoption.

Political usefulness for other technologies

Finally, future research could explore whether other technologies besides formalized commercial methodologies are useful from a political perspective. One example of a technology that may have political usefulness is email. Prior research indicates that email is often used defensively to cover one's position. Markus (1994) investigated email use within organizations and find that employees use email defensively to cover themselves. By using email rather than face-to-face communication, employees were able to construct an audit trail of all communication. By providing an audit trail of conversations, in turn, email use became a way to hold other employees accountable. Overall, future research should investigate the appropriateness of political usefulness for other technologies besides formalized commercial methodologies.

Concluding Comments

Acknowledging the limitations of this study, this research makes several important contributions to Information Systems research and practice. This research represents the first study to break-out the perceived usefulness construct into rational and political usefulness and their referent dimensions as well as the first attempt to explain developer *attitude*, rather than *intention to use*, toward using a formalized *commercial*, rather than *in-house*, methodology.

References

Agarwal, R., & Karahanna, E. (2000). Time flies when you're having fun: Cognitive absorption and beliefs about information technology usage. *MIS Quarterly, 24*(4), 665–694.

Agarwal, R., & Prasad, J. (2000). A field study of the adoption of software process innovations by information systems professionals. *IEEE Transactions on Engineering Management, 47*(3), 295–308.

Agarwal, R., Sambamurthy, V., & Stair, R. M. (2000). Research report: The evolving relationship between general and specific computer self-efficacy—An empirical assessment. *Information Systems Research, 11*(4), 418–430.

Ajzen, I. (1988). *Attitudes, personality, and behavior.* Chicago: Dorsey Press.

Ajzen, I. (1991). The theory of planned behavior? *Organizational Behavior and Human Decision Processes, 50*(2), 179–211.

Ajzen, I., & Fishbein, M. (1980). *Understanding attitudes and predicting social behavior*. Englewood Cliffs, NJ: Prentice-Hall.

Ambler, S. (2005). *A manager's introduction to the Rational Unified Process (RUP).* Retrieved September 30, 2005, from http://www.ambysoft.com/unifiedprocess/ rupIntroduction.html.

Avison, D. E., & Fitzgerald, G. (1995). Information systems development: Methodologies, techniques and tools. London: McGraw-Hill.

Avison, D., & Fitzgerald, G. (2003). Where now for development methodologies? *Communications of the ACM, 46*(1), 78–82.

Bachmann, D., Elfrink, J., & Vanazza, G. (1996). Tracking the progress of e-mail vs. snail-mail. *Marketing Research, 8*(2), 30–35.

Backlund, P., Hallenborg, C., & Hallgrimsson, G. (2003). Transfer of development process knowledge through method adaptation and implementation. *Proceedings of the Eleventh European Conference on Information Systems*.

Bagozzi, R. P. (1985). Expectancy-value attitude models: An analysis of critical theoretical issues. *International Journal of Research Marketing, 2*, 43–60.

Bagozzi, R. P. (1988). The rebirth of attitude research in marketing. *Journal of the Market Research Society, 30*(2), 163–195.

Bahli, B., & Buyukkurt, M. D. (2005). Group performance in information systems project groups: An empirical study. *Journal of Information Technology Education, 4*, 97–113.

Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review, 84*(2), 191–215.

Bandura, A. (1982). Self-efficacy mechanism in human agency. *American Psychologist*, *37*(2), 122–147.

Bandura, A. (1986). *Social foundations of thought and action.* Englewood Cliffs, NJ: Prentice-Hall.

Bandura, A. (1997). *Self-efficacy: The exercise of control.* New York: W.H. Freeman.

Barling, J., &. Beattie, R. (1983). Self-efficacy beliefs and sales performance. *Journal of Organizational Behavior Management, 5,* 41–51.

Bassellier, G., & Benbasat, I. (2004). Business competence of information technology professionals: Conceptual development and influence on IT-business partnerships. *MIS Quarterly, 28*(4), 673–694.

Beath, C. M., & Orlikowski, W. J. (1994). The contradictory structure of systems development methodologies: Deconstructing the IS-user relationship in information engineering. *Information Systems Research, 5*(4), 350–377.

Betz, N. E., & Hackett, G. (1981). The relationships of career-related self-efficacy expectations to perceived career options in college women and men. *Journal of Counseling Psychology, 28*(5), 399–410.

Beynon-Davies, P., & Williams, M. D. (2003). The diffusion of information systems development methods. *Journal of Strategic Information Systems*, *12*(1), 29–46.

Bollen, K. A. (1984) Multiple indicators: Internal consistency or no necessary relationship? *Quality & Quantity,* 18, 377–385.

Bollen, K. A. (1989). *Structural equations with latent variables.* New York: John Wiley & Sons.

Borgatta, E. F., & Bohrnstedt, G. W. (1980). Level of measurement once over again. *Sociological Methods & Research*, *9*(2), 147–160.

Bostrom, R. (1989). Successful application of communication techniques to improve the systems development process. *Information and Management*, *16*(5), 279–275.

Boudreau, M. C., Gefen, D., & Straub, D. (2001). Validation in IS Research: A state-of-the-art assessment. *MIS Quarterly, 25*(1), 1–24.

Brown, S. A., Massey, A. P., Montoya-Weiss, M. M., & Burkman, J. R. (2002). Do I really have to? User acceptance of mandated technology. *European Journal of Information Systems, 11*(4), 283–295.

Bryant, S. M., Hunton, J. E., & Stone, D. N. (2004). Internet-based experiments: Prospects and possibilities for behavioral accounting research. *Behavioral Research in Accounting, 16*(1), 107–129.

Chatzoglou, P. D., & Macaulay, L. A. (1996). Requirements capture and IS methodologies. *Information Systems Journal*, *6*(3), 209–225.

Chau, P. Y. K. (1996). An empirical investigation on factors affecting the acceptance of CASE by systems developers. *Information and Management, 30*(6), 269–280.

Chau, P. Y. K., & Hu, P. J. H. (2001). Information technology acceptance by individual professionals: A model comparison approach. *Decision Sciences, 32*(4), 699–719.

Chin, W. W. (1998a). Commentary: Issues and opinion on structural equation modeling. *MIS Quarterly, 22*(1), vii–xvi.

Chin, W. W. (1998b). The partial least squares approach for structural equation modeling. In G. A. Marcoulides (Ed.), *Modern methods for business research* (295–336). Hillsdale, NJ: Lawrence Erlbaum Associates.

Chin, W. W. (2000). Partial least squares for researchers: An overview and presentation of recent advances using the PLS approach. Retrieved November 13, 2006, from http://www.bauer.uh.edu/plsgraph/plstalk.pdf.

Chin, W. W., & Gopal, A. (1995). Adoption intention in GSS: relative importance of beliefs. *Data Base for Advances in Information Systems, 26*(2), 42–64.

Chin, W. W., & Newsted, P. R. (1999). Structural equation modeling analysis with small samples using partial least squares. In R. Hoyle (Ed.), *Statistical strategies for small sample research* (307–341). Thousand Oaks, CA: Sage Publications.

Chin, W. W., Marcolin, B. L., & Newsted, P. R. (1996). A partial least squares latent variable modeling approach for measuring interaction effects: Results from a Monte Carlo simulation study and voice mail emotion/adoption study. In J. I. DeGross, S. Jarvenpaa, & A. Srinivasan (Eds.), *Proceedings of the 17th International Conference on Information Systems* (21–41).

Churchill, G. (1979). A paradigm for developing better measures of marketing constructs. *Journal of Marketing Research, 16*(1)*,* 64–73.

Coderre, G., Mathieu, A., & St-Laurent, N. (2004). Comparison of the quality of qualitative data obtained through telephone, postal and email surveys. *International Journal of Market Research, 46*, 347–357.

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Earlbaum Associates.

Cohen, J. (1969). *Statistical power for the behavioral sciences*. New York: Academic Press.

Cohen, J., & Cohen, P. (1983). *Applied multiple regression/correlation analysis for the behavioral sciences* (2nd ed.). Hillsale, NJ: Erlbaum.

Compeau, D. R., & Higgins, C. A. (1995a). Computer self-efficacy: Development of a measure and initial test. *MIS Quarterly*, *19*(2), 189–211.

Compeau, D. R., & Higgins, C. A. (1995b). Application of social cognitive theory to training for computer skills. *Information Systems Research, 6*(2), 118–143.

Couper, M. (2000). Web surveys: A review of issues and approaches. *Public Opinion Quarterly, 64*(4), 464–494.

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly, 13*(3), 319–340.

Davis, F. D., Bagozzi, R. P., & Warshaw, P. R. (1992). Extrinsic and intrinsic motivation to use computers in the workplace. *Journal of Applied Social Psychology, 22*(14)*,* 1111–1132.

DeVillis, R. F. (1991). *Scale development: Theory and applications.* Newbury Park, CA: Sage Publications.

Diamantopoulos, A., & Siguaw, J. A. (2006). Formative versus reflective indicators in organizational measure development: A comparison and empirical illustration. *British Journal of Management, 17*(4)*,* 263-282.

Dishaw, M. T., Strong, D. M., & Bandy, D. B. (2002). Extending the task-technology fit model with self-efficacy constructs. *Proceedings of the Eighth Americas Conference on Information Systems* (1021–1027).

Duggan, E. W. (2004). Silver pellets for improving software quality. *Information Resources Management Journal, 17*(2), 1–21.

Eisenhardt, K. (1989). Building theories from case study research. *Academy of Management Review, 14*(4), 532–550.

Ewusi-Mensah, K. (1997). Critical issues in abandoned information systems development projects. *Communications of the ACM*, *40*(9), 74–80.

Ferratt, T. W., & Short, L. E. (1988). Are information systems people different? An investigation of how they are and should be managed. *MIS Quarterly, 12*(3), 427–443.

Fichman, R. (1992). Information technology diffusion: A review of empirical research. In J. DeGross, J. Elam, & J. Becker (Eds.), *Proceedings of the 13th Annual International Conference on Information Systems* (195–206).

Fitzgerald, B. (1994). Whither systems development: time to move the lamppost? In C. Lissoni et al. (Eds.) *Proceedings of Second Conference on Information Systems Methodologies* (371-380). Swindon: BCS Publications.

Fitzgerald, B. (1996). Formalized systems development methodologies: A critical perspective. *The Information Systems Journal, 6*(1), 3–23.

Fitzgerald, B. (1997). The use of systems development methodologies in practice: A field study. *Information Systems Journal*, *7*(3), 201–212.

Fitzgerald, B. (1998a). An empirical investigation into the adoption of systems development methodologies. *Information & Management, 34*(6), 317–328.

Fitzgerald, B. (1998b). An empirically-grounded framework for the information systems development process. In R. Hirschheim, M. Newman, and J. deGross (Eds.) *Proceedings of the International Conference on Information Systems* (103–114).

Fitzgerald, B., Russo, N. L., & Stolterman, E. (2002). *Information systems development: Methods in action.* London: McGraw-Hill.

Floyd, F.J., & Widaman, K. (1995). Factor analysis in the development and refinement of clinical assessment instruments. *Psychological Assessment, 7*(3)*,* 286–299.

Fornell, C., & Larker, D. (1981). Evaluating structural equation models with unobservable variables and measurement error. *Journal of Marketing Research, 18*(1)*,* 39–50.

Franz, C. R., & Robey, D. (1984). An investigation of user-led system design: rational and political perspectives. *Communications of the ACM, 27*(12)*,* 1202–1209.

Gefen, D., Straub, D., & Boudreau, M. (2000). Structural equation modeling and regression: Guidelines for research practice. *Communications of the Association for Information Systems, 4*(7)*,* 1–57.

Gefen, D., & Straub, D (2000). A practical guide to factorial validity using PLS-GRAPH: Tutorial and annotated example. *Communications of the Association for Information Systems, 16,* 91–109.

Glass, R. L. (1996). Through a glass, darkly. *Data Base Advances, 27*(1), 14–16.

Goodhue, D., Lewis, W., & Thompson, R. (2006). PLS, small sample size, and statistical power in MIS research. *Proceedings of the 39th Hawaii International Conference on System Sciences*.

Gopal, A., Bostrom, R., & Chin, W. (1992). Applying adaptive structuration theory to investigate the process of group support systems use. *Journal of Management Information Systems, 9*(3), 45–69.

Hardgrave, B. C., & Johnson, R. A. (2003). Toward an information systems development acceptance model: The case of object-oriented systems development. *IEEE Transactions on Engineering Management, 50*(3), 322–336.

Hardgrave, B. C., Wilson, R. L., & Eastman, K. (1999). Toward a contingency model for selecting an information systems prototyping strategy. *Journal of Management Information Systems, 16*(2), 113–136.

Hardgrave, B. C., Davis, F. D., & Riemenschneider, C. K. (2003). Investigating determinants of software developers' intentions to follow methodologies. *Journal of Management Information Systems, 20*(1), 123–151.

Hardy, C. J., Thompson, J. B., & Edwards, H. M. (1995). The use, limitations and customization of structured systems development methods in the United Kingdom. *Information and Software Technology, 37*(9), 467–477.

Harter, D. E., Krishnan, M. S., & Slaughter, S. A. (2000). Effects of process maturity on quality, cycle time, and effort in software development. *Management Science, 46*(4), 451–466.

Hartwick, J., & Barki, H. (1994). Explaining the role of user participation in information system use. *Management Science, 40*(4), 440–465.

Herbsleb, J., Carleton, A., Rozum, J., Siegel, J., & Zubrow, D. (1994). *Benefits of CMM-based software process improvement: Initial results.* Pittsburgh, PA: Carnegie Mellon University, Software Engineering Institute.

Hirschheim, R. A., Klein, H-K., & Lyytinnen, K. (1995). *Information systems development and data modeling: Conceptual and philosophical foundations*. New York: Cambridge University Press.

Huisman, M., & Iivari, J. (2002). The individual deployment of systems development methodologies. *Lecture Notes in Computer Science, 2348,* 134–150.

Huisman, M., & Iivari, J. (2006). Deployment of systems development methodologies: Perceptual congruence between IS managers and systems developers. *Information & Management, 43*(1)*,* 29–49.

Hung, S-Y., & Chang, C-M. (2005). User acceptance of WAP services: test of competing theories. *Computer Standards & Interfaces, 27*(4), 359–370.

Igbaria, M., Parasuraman, S., & Badawy, M. K. (1994). Work experiences, job involvement, and quality of work life among information systems personnel. *MIS Quarterly*, *18*(2), 175–201.

Iivari, J. (1996). Why are CASE tools not used? *Communications of the ACM, 39*(10), 94–103.

Iivari, J., Hirschheim, R., & Klein, H. K. (2000). A dynamic framework for classifying information systems development methodologies and approaches. *Journal of Management Information Systems, 17*(3), 179–218.

Jaccard, J. & Choi K.W. (1996). *LISREL approaches to interaction effects in multiple regression.* Thousand Oaks, CA: Sage Publications

Jayaratna, N. (1994). *Understanding and evaluating methodologies*. London: McGraw-Hill.

Johnson, R. A., Hardgrave, B. C., & Doke, E. R. (1999). An industry analysis of developer beliefs about object-oriented systems development. *The Data Base for Advances in Information Systems, 30*(1), 47–64.

Jones, G. R. (1986). Socialization tactics, self-efficacy, and newcomers. Adjustments to organizations. *Academy of Management Journal, 29*(2), 262–279.

Kautz, K., Hansen, B., & Jacobsen, D. (2004). The utilization of information systems development methodologies in practice. *Journal of Information Technology Cases and Applications, 6*(4), 1–20.

Khalifa, M., & Verner, J. M. (2000). Drivers for software development usage. *IEEE Transactions on Engineering Management, 47*(3), 360–369.

Kozar, K. A. (1989). Adopting systems development methods: An exploratory study. *Journal of Management Information Systems, 5*(4), 73–86.

Lamantia, J. (2006). Analyzing card sort results with a spreadsheet template. Retrieved on February 27, 2007, from http://www.boxesandarrows.com.

Lankton, N., & McKnight, D. H. (2007). Using expectation disconfirmation theory to predict technology trust and usage continuance intentions. Unpublished manuscript.

Lazar, J., & Preece, J. (1999). Designing and implementing web-based surveys. *Journal of Computer Information Systems, 39*(4), 63–67.

Lee, J., & Kim, S. (1992). The relationship between procedural formalization and MIS success. *Information & Management, 22*(2), 89–111.

Lee, J., & Truex, D. (2000). Exploring the impact of formal training in ISD methods on the cognitive structure of novice information systems developers. *Information Systems Journal, 10*(4)*,* 347–367.

Leonard-Barton, D. (1988). Implementation characteristics of organizational innovations. *Communication Research, 15*(5), 603–631.

Lohmoller, J.B. (1989). *Latent variable path modeling with partial least squares*, Heidelberg: Physica-Verlag.

Lyytinen, K. (1987). Different perspectives on information systems: Problems and solutions. *ACM Computing Surveys, 19*(1), 5–46.

Marakas, G., Yi, M., & Johnson, R. (1998). The multilevel and multifaceted character of computer self-efficacy: Toward clarification of the construct and an integrative framework for research. *Information Systems Research, 9*(2), 126–163.

Markus, M. L. (1983). Power, politics, and MIS implementation. *Communications of the ACM, 26*(6), 430–444.

Markus, M. L. (1994). Finding a happy medium: explaining the negative effects of electronic communication on social life at work. *ACM Transactions on Information Systems, 12*(2), 119–149.

Markus, B. H., Selby, V. C., Niaura, R. S., & Rossi, J. S. (1992). Self-efficacy and the stages of exercise behavior change. *Research Quarterly for Exercise and Sport, 63*(1), 60–66.

Mathieson, K. (1991). Predicting user intentions: Comparing the technology acceptance model with the theory of planned behavior. *Information Systems Research, 2*(3), 173–191.

Melone, N.P. (1990). A theoretical assessment of the user-satisfaction construct in information systems research. *Management Science, 36*(1), 76–91.

Moon, J. and Kim, Y. (2001). Extending the TAM for a World-Wide Web context. *Information & Management, 38*(4), 217–230.

Moore, G. C. and Benbasat, I. (1991). Development of an instrument to measure the perceptions of adopting an information technology isnnovation. *Information Systems Research, 2*(3), 192–222.

Nah, F. H., Tan, X., and Teh, H. S. (2004). An empirical investigation on end-user's acceptance of enterprise systems. *Information Resources Management Journal, 17*(3), 32–53.

Nandhakumar, J., & Avison, D. E. (1999). The fiction of methodological development: A field study of information systems development. *Information Technology & People, 12*(2), 176–191.

Necco, C. R., Gordon, C. L., & Tsai, N. W. (1987). Systems analysis and design: current practices. *MIS Quarterly, 11*(4), 461–476.

Nerur, S., Mahapatra, R.., & Mangalaraj, G. (2005). Challenges of migrating to Agile methodologies. *Communications of the ACM, 48*(5), 73–78.

Netemeyer, R. G., Bearden, W. O., & Sharma, S. C. (2003). *Scaling procedures. Issues and applications.* Thousand Oaks, CA: Sage Publications.

Nunnally, J. C. (1978). *Psychometric theory* (2nd ed.). New York: McGraw-Hill.

Oh, S., Ahn, J., & Kim, B. (2003). Adoption of broadband internet in Korea: The role of experience in building attitudes. *Journal of Information Technology, 18*(4), 267–280.

Orlikowski, W. J. (1993). CASE Tools as organizational change: Investigating incremental and radical changes in systems development. *MIS Quarterly, 17*(3), 309–340.

Palvia, P., & Nosek, J. T. (1990). An empirical evaluation of system development methodologies. *Information Resources Management Journal, 3*(3), 23–32.

Pajares, F. (2002). *Self-efficacy beliefs in academic contexts: An outline.* Retrieved October 30, 2006, from http://www.des.emory.edu/mfp/efftalk.html.

Pedersen, P. E. (2005). Adoption of mobile internet services: An exploratory study of mobile commerce early adopters. *Journal of Organizational Computing and Electronic Commerce, 15*(3), 203–222.

Pedhazur, E. J., & Schmelkin, L.P. (1991). *Measurement, design and analysis: An integrated approach*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Raghavan, S. A., & Chand, D. R. (1989). Diffusing software engineering methods. *IEEE Software, 6*(4), 81–90.

Rai, A., Patnayakuni, R., & Seth, N. (2006). Firm performance impacts of digitally enabled supply chain integration capabilities. *MIS Quarterly, 30*(2), 225–246.

Ram, S., & Jung, H-S. (1991). Forced adoption of innovations in organizations: Consequences and implications. *Journal of Product Innovation Management, 8*(2), 117–126.

Rawstorne, P., Jayasuriya, R., & Caputi, P. (1998). An integrative model of information systems use in mandatory environments. In R. Hirschheim, M. Newman, & J. I. DeGross (Eds.), *Proceedings of the International Conference on Information Systems* (325–350). Helsinki, Finland: Association for Information Systems.

Riemenschneider, C. K., Hardgrave, B. C., & Davis, F. D. (2002). Explaining software developer acceptance of methodologies: A comparison of five theoretical models. *IEEE Transactions on Software Engineering 28*(12), 1135–1145.

Ringle, C. M., Wende, S., & Will A. (2005). SmartPLS 2.0 (beta), www.smartpls.de.

Roberts, T. L., Jr., Gibson, M. L., Fields, K. T., & Rainer, R. K., Jr. (1998). Factors that impact implementing a system development methodology. *IEEE Transactions on Software Engineering, 24*(8), 640–649.

Robey, D., & Markus, L. (1984). Rituals in information system design. *MIS Quarterly*, *8*(1), 5–15.

Rogers, E. (1983). *Diffusions of innovations.* (3rd ed.). New York: The Free Press.

Rogers, E. (1995). *Diffusion of innovations*. (4th ed.). New York: The Free Press.

Russo, N. L., Hightower, R., & Pearson, J. M. (1996). The failure of methodologies to meet the needs of current development environments. In N. Jayaratna, & B. Fitzgerald (Eds.), *Lessons learned from the use of methodologies, proceedings of* *the fourth conference of the British Society Information Systems Methodologies Specialist Group* (387–394). Cork, Ireland: University College of Cork.

Russo, N. L., & Stolterman, E. (2000). Exploring the assumptions underlying information systems methodologies. *Information Technology & People, 13*(4), 313–327.

Russo, N. L., Wynekoop, J. L., & Walz, D. B. (1995). The use and adaptation of system development methodologies. In M. Khosrowpour (Ed.), *Proceedings of the 1995 International Resources Management Association International Conference* (162). Atlanta GA.

Sauer, C., & Lau, C. (1997). Trying to adapt systems development methodologies—a case-based exploration of business users' interests. *Information Systems Journal*, *7*(4), 255–275.

Siau, K., & Tan, X. (2005). Evaluation criteria for information systems development methodologies. *Communications of the AIS*, *16*, 860–876.

Shih, Y. (2004). The use of a decomposed theory of planned behavior to study Internet banking in Taiwan. *Internet Research, 14*(3), 213–223.

The Standish Group. (2004). 2004 CHAOS demographics and project resolution. Retrieved November 11, 2006, from http://www.standishgroup.com/ sample_research/PDFpages/q3-spotlight.pdf.

Staples, D. S., & Seddon, P. (2004). Testing the technology-to-performance chain model. *Journal of Organizational and End User Computing, 16*(4), 17-36.

Straub, D. (1989). Validating instruments in MIS research. *MIS Quarterly, 13*(2), 147–169.

Sultan, F., & Chan, L. (2000). The adoption of new technology: The case of object-oriented computing in software companies. *IEEE Transactions on Engineering Management, 47*(1), 106–126.

Taylor, S., & Todd, P. A. (1995). Understanding information technology usage: A test of competing models. *Information Systems Research, 6*(2), 144–176.

Temme, D., Kreis, H. & Hildebrandt, L. (2006). PLS path modeling—A software review. Retrieved July 20, 2007, from http://sbf649.wiwi.hu-berlin.de.

Templeton, G. F., & Byrd, T. A. (2003). Determinants of the relative advantage of a structured SDM during the adoption stage of implementation. *Information Technology and Management, 4*(4), 409–428.

Thompson, R. L., Higgins, C. A., & Howell, J. M. (1991). Personal computing: Toward a conceptual model of utilization. *MIS Quarterly, 15*(1), 125–143.

Thompson, R., Compeau, D., & Higgins, C. (2006). Intentions to use information technologies: An integrative model. *Journal of Organizational and End User Computing, 18*(3), 25–46.

Tittle, C. R., & Hill, R. J. (1967). Attitude measurement and prediction of behavior: An evaluation of conditions and measurement techniques. *Sociometry, 20*(2), 199–213.

Toleman, M., Ally, M., & Darroch, F. (2004). Aligning adoption theory with agile system development methodologies. *Proceedings of the 8th Pacific-Asia Conference on Information Systems* (458-471).

Townsend, J. T., & Ashby, F. G. (1984). Measurement scales and statistics: The misconception misconceived. *Psychological Bulletin*, *96*(2), 394–401.

Velleman, P. F., & Wilkinson, L. (1993). Nominal, ordinal, interval, and ratio typologies are misleading. *The American Statistician*, *47*(1), 65–72.

Venkatesh, V., & Davis, F. D. (1996). A model of the antecedents of perceived ease of use: Development and test. *Decision Sciences, 27*(3), 451–481.

Venkatesh, V., & Davis, F. (2000). A theoretical extension of the technology acceptance model: Four longitudinal studies. *Management Science, 46*(2), 186–204.

Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. *MIS Quarterly 27*(3), 425–478.

Vessey, I., Jarvenpaa, S. L., & Tractinsky, N. (1992). Evaluation of vendor products: CASE tools as methodology companions. *Communications of the ACM, 35*(4), 90–105.

Vitalari, N. (1985). Knowledge as a basis for expertise in systems analysis: An empirical study. *MIS Quarterly, 9*(3), 221–241.

Wallace, L., Keil, M., & Rai, A. (2004). How software project risk affects project performance: An investigation of the dimensions of risk and an exploratory model. *Decision Sciences, 35*(2), 289–321.

Ward, K. W., Brown, S.A., & Massey, A. P. (2005). Organisational influences on attitudes in mandatory system use environments: a longitudinal study. *International Journal of Business Information Systems, 1*(1-2), 9–30.

Wastell, D. (1996). The fetish of technique: methodology as a social defense. *Information Systems Journal, 6,* 25–40.

Wastell, D. (1999). Learning dysfunctions in information systems development: Overcoming the social defenses with transitional objects. *MIS Quarterly, 23*(4), 581–600.

Wastell, D. G., & Newman, M. (1993). The behavioral dynamics of information system development: a stress perspective. *Accounting, Management and Information Technologies, 3,* 121–148.

Whitley, B. (2002). *Principles of research in behavioral science.* (2nd ed.). New York: McGraw-Hill.

Wing, H., & Nelson, C. E. (1972). The perception of personality through trait sorting: A comparison of trait sampling techniques. *Multibehavioral Research,* 269–274.

Wold, H. (1982). *Soft modeling the basic design and some extensions in systems under indirect observation: Causality, structure, prediction* (volume 2), K.G. Joreskog and H. Wold (Eds.), North-Holland, Amsterdam, 1–54.

Wynekoop, J., & Russo, N. (1997). Studying system development methodologies: an examination of research methods. *Information Systems Journal. 7*(1), 47–65.

Yi, M., & Davis, F. (2003). Developing and validating an observational learning model of computer software training and skill acquisition. *Information Systems Research, 14*(2), 146–169.

Zuboff, S. (1988). *In the age of the smart machine*. New York: Basic Books.

Appendix A Round One Card Sorting Results

Table 1 Item Placement Ratios

	Actual Categories										
Target Construct	No. of Items	Product	Process	Comm	Career	Symbolic	Support	Defense	N/A	Total	Tgt %
Product	9	18	8	0	0	0	0	0	1	27	67%
Process	17	8	17	0	9	3	0	0	0	37	46%
Communication	12	0	0	42	2	6	0	0	1	51	82%
Career	10	0	0	0	30	0	0	0	0	30	100%
Symbolic	10	0	4	0	2	14	0	0	1	21	67%
Support	10	0	6	0	0	0	13	0	1	20	65%
Defense	12	6	3	0	0	2	1	14	1	27	52%

Table 2 Judges' Labels for Categories

	Judges	
Target Construct	**A**	**B**
Product	Quality Assurance	N/A
Process	Process Flow, Timeliness	Time management, Cost control, Design & Development
Communication	Communication	Internal team dynamics
Career	Personal Gain	Personal/Professional Growth
Symbolic	Team Performance	Communication with stakeholders
Support	Emotions/Feelings	Individual proof of concept
Defense	Job Security	CYA/negative/poor performance/results

Appendix B Round Two Card Sorting Results

Table 1 Item Placement Ratios

	Actual Categories										
Target Construct	No. of items	Product	Process	Comm	Career	Symbolic	Support	Defense	N/A	Total	Tgt %
Product	9	30	15	0	0	0	0	0	7	52	58%
Process	17	29	55	14	4	17	3	12	1	135	41%
Communication	12	0	6	96	0	6	0	6	0	114	84%
Career	10	0	0	0	39	0	10	0	1	50	78%
Symbolic	10	2	11	2	3	15	1	14	1	49	31%
Support	10	0	8	0	16	9	25	1	0	59	42%
Defense	12	6	12	4	1	2	1	36	0	62	58%

Table 2 Item Placement Ratios (75% placement in the target constructs)

	Actual Categories										
Target Construct	No. of items	Product	Process	Comm	Career	Symbolic	Support	Defense	N/A	Total	Tgt %
Product	6	24	9	0	0	0	0	0	4	37	65%
Process	11	15	43	9	2	11	2	9	1	92	47%
Communication	12	0	6	96	0	6	0	6	0	114	84%
Career	10	0	0	0	39	0	10	0	1	50	78%
Symbolic	1	0	0	0	0	3	0	1		4	75%
Support	6	0	2	0	11	6	18	0	0	37	49%
Defense	8	1	3	1	0	1	0	30	0	36	83%

Table 3 Judges' Labels for Categories

	Judges			
Target Construct	**A**	**B**	**C**	**D**
Product	Better system	Benefit to system	Quality Assurance	Control of process, Requirements capture, System maintenance
Process	Better system	Benefit to project	Documentation, Design, Development, Maintenance, Make sure requirements are defined, Scheduling	Documentation, Control of process, Pleasing the Project Manager, Systems development efficiency, Schedule related, System maintenance
Communication	Communication	Benefit/impact on users, Benefit to project	Communication	Quality of Communication, Frequency of Communication
Career	Promotion	Benefit to developer	Boost confidence of team members	Increasing value
Symbolic	Validation	Benefit to company, Benefit to project	Stay competitive	External appearances
Support	Validation	Benefit to developer	N/A	Stress
Defense	CYA	Benefit to project, Benefit for project post-mortem	Provide protection	Protection

Appendix C Round Three Card Sorting Results

Table 1 Item Placement Ratios

		Actual Categories									
Target Construct	No. of Items	Product	Process	Comm	Career	Symbolic	Support	Defense	N/A	Total	Tgt %
Product	8	64	15	0	0	0	0	0	3	82	78%
Process	9	17	62	8	0	1	0	0	12	100	62%
Communication	4	4	4	33	0	0	0	0	0	41	80%
Career	9	1	0	0	79	0	9	0	1	90	88%
Symbolic	10	3	2	4	4	66	1	8	11	99	67%
Support	9	0	0	2	8	1	81	1	3	96	84%
Defense	11	3	5	3	1	16	4	74	7	113	65%

Table 2 Judges' (Developers) Labels for Categories

	Developer Judges		
Target Construct	A	B	D
Product	ROI	Quality of deliverable	Impact on system quality, Schedule Performance
Process	Control-BP, Planning	Schedule efficiency	Impact on system quality, Schedule Performance
Communication	Communication	Communication	Schedule Performance
Career	General team factors	Developer--Promotion	Self-promotion/growth
Symbolic	CYA/Contract	Giving impression that you're using a method	Perception of why I'm using a methodology
Support	General team factors	Developer-Anxiety	Confidence that developer is doing the right thing
Defense	CYA/Contract	Defending your decisions/justification	Defense when sitting down with customer.

Table 3: Judges' (Students) Labels for Categories

	Student Judges					
Target Construct	**A**	**B**	**C**	**D**	**E**	**F**
Product	Types of Systems	Efficiency	Development: Quality	General-System	Development process for efficient systems	System being developed
Process	Development Process, Time/Length	Meeting schedule, Breaking down the development process	Development: Process organization and control	Development, Speed, User Demands	Project cycle time/project cycle	Development process
Communication	Communication	Communication	Interaction: Internal communication	Communication	User participation/user communication, Communication within development team	Communication with stakeholders
Career	Personal Attributes	Benefits	Individual benefits/detriments	Career	Personal benefit	Skills
Symbolic	Appearances to Outside/Third Parties	Using rigorous process	Interaction: Demonstration	Decisions	Reliability	Other's viewpoint
Support	Effects on developers	Scope creep	Effects on developer moods	Stress	Developer anxiety	Developer psychology
Defense	Defensive Guard	Decision validation	Interaction: Criticism	Gone wrong, Decisions	Defense for decisions made	Dealing with decisions, Defending against criticism and user demands

Appendix D Pilot Study Items

Product Usefulness	
PROD1	Using this methodology helps to develop more efficient systems.
PROD2	Using this methodology helps to develop better systems.
PROD3	Using this methodology helps to develop more reliable systems.
PROD4	* Using this methodology does not result in higher quality systems.
PROD5	Using this methodology helps to develop higher quality systems.
PROD6	Using this methodology helps to develop more maintainable systems.
PROD7	Using this methodology helps to develop systems with more functionality.
Process Usefulness	
PROC1	Using this methodology helps to break down the development process into manageable parts.
PROC2	Using this methodology helps to keep the development project on schedule.
PROC3	*Using this methodology reduces the likelihood of delivering the system on time.
PROC4	Using this methodology improves schedule performance.
PROC5	Using this methodology helps to meet scheduled delivery dates.
PROC6	Using this methodology results in a more efficient development process.
PROC7	Using this methodology improves control over the systems development process.
PROC8	Using this methodology increases the productivity of the development process.
Communication Usefulness	
COMM1	Using this methodology improves communication with the user's manager(s).
COMM2	Using this methodology improves communication with managers.
COMM3	Using this methodology improves communication with users.
COMM4	Using this methodology improves communication with other team members.
Career Usefulness	
CAREER1	Many job opportunities are available for me as a result of using this methodology.
CAREER2	Using this methodology increases my opportunity to gain job security.
CAREER3	Knowing this methodology is a valuable skill.
CAREER4	Using this methodology increases my opportunity for preferred work assignments.
CAREER5	Using this methodology increases my chances for pay increases.
CAREER6	Using this methodology increases my chances for promotion.
CAREER7	Using this methodology improves my marketability.
CAREER8	Using this methodology puts me on the "cutting edge" in my field.
CAREER9	Using this methodology increases my flexibility of changing jobs.
Symbolic Usefulness	
SYMB1	Using this methodology makes it look like to others that the systems development process is controlled.
SYMB2	Using this methodology gives the appearance to others that professional systems development practices are being used.
SYMB3	Using this methodology gives the appearance to others that industry standards are being used.
SYMB4	Using this methodology provides an appearance to others that credible systems development practices are being used.
SYMB5	Using this methodology indicates to others that sound development practices are being used.
SYMB6	This methodology is a useful marketing tool.

SYMB7	Using this methodology provides confidence to others that the systems development project will succeed.
SYMB8	This methodology gives the impression to others that accepted systems development practices are being used.
SYMB9	Using this methodology demonstrates to others that systematic development processes are being used.
Support Usefulness	
ANX1	Using this methodology reduces developer stress during systems development.
ANX2	Using this methodology reduces developer nervousness during systems development.
ANX3	Using this methodology helps developers deal with the pressures of systems development.
ANX4	Using this methodology reassures developers that they are using sound development practices.
ANX5	Using this methodology relieves developer apprehension during systems development.
ANX6	Using this methodology increases developer confidence during systems development.
ANX7	*Using this methodology reduces developer confidence during systems development.
ANX8	Using this methodology helps developers cope with the stress of systems development.
ANX9	Using this methodology reduces developer anxiety during systems development.
Defense Usefulness	
DEF1	If a design decision is wrong, using this methodology shows why it was decided that way.
DEF2	If the software project fails, having used this methodology provides protection against criticism.
DEF3	Using this methodology guards against complaints that a formal systematic process was not followed.
DEF4	Using this methodology protects against unreasonable deadlines.
DEF5	This methodology can be used to authorize decisions made during systems development.
DEF6	Using this methodology defends against unreasonable demands.
DEF7	If a decision made during systems development turns out to be wrong, this methodology can be used to provide protection from blame.
DEF8	Using this methodology can defend against criticism that a formal systems development process was not used.
DEF9	Using this methodology protects against unreasonable demands for additional system functionality.
DEF10	This methodology can be used to justify decisions made during systems development.
DEF11	This methodology can be used to validate decisions made during systems development.
DEF12	If a decision made during systems development turns out to be wrong, this methodology can be used to show the rationale for the decision.
DEF13	Using this methodology provides protection from unreasonable demands.
Perceived Ease of Use	
PEOU1	Learning this methodology was easy for me.
PEOU2	*It took too long to learn how to use this methodology to make it worth the effort.
PEOU3	I think this methodology is clear and understandable.
PEOU4	It was easy for me to become skillful at using this methodology.
PEOU5	Using this methodology does not require a lot of mental effort.
PEOU6	I find this methodology easy to use.
PEOU7	Using this methodology does not take too much time from my normal duties.
PEOU8	This methodology is not cumbersome to use.
Compatibility	
COMPAT1	Using this methodology is compatible with all aspects of my work.
COMPAT2	This methodology is compatible with the way I develop systems.
COMPAT3	Using this methodology is completely compatible with my current situation.
COMPAT4	Using this methodology fits into my work style.

COMPAT5	Using this methodology fits well with the way I work.
COMPAT6	*Using this methodology would be incompatible with the way I like to work.
Self Efficacy	
SE1	I could develop an information system if there was no one around to tell me what to do as I go.
SE2	I could develop an information system if I had never developed one like it before.
SE3	I could develop an information system if I could call someone for help if I got stuck.
SE4	I could develop an information system if someone else helped me get started.
SE5	I could develop an information system if I could have a lot of time to create it.
SE6	I could develop an information system if someone showed me how to do it first.
SE7	I could develop an information system if I had built a similar information system in the past.
SE8	I could develop an information system if I had seen someone develop one before developing one myself.
SE11	I could develop an information system if this methodology was not available to tell me what to do.
SE12	I could develop an information system if I had only this methodology for reference.
SE14	I could develop an information system without following this methodology.
Perceived Usefulness (TAM)	
PU1	Using this methodology increases my productivity.
PU2	Using this methodology improves my job performance.
PU3	Using this methodology enhances the quality of my work.
PU4	The advantages of using this methodology outweigh the disadvantages.
PU5	Using this methodology makes it easier to do my job.
PU6	This methodology is useful in my job.
Voluntariness	
VOL1	*My superiors expect me to use this methodology.
VOL2	My boss does not require me to use this methodology.
VOL3	*The use of this methodology is required for my job.
VOL4	Although it might be helpful, using this methodology is certainly not mandatory in my job.
VOL5	My use of this methodology is voluntary (as opposed to being required).
Attitude	
ATT1	Using this methodology is a good idea.
ATT2	I think it is very good to use this methodology for developing systems.
ATT3	I am enthusiastic about using this methodology.
ATT4	In my opinion, it would be very desirable to use this methodology for developing systems.
ATT5	Using this methodology is a wise idea.
ATT6	It would be much better for me to use this methodology for developing systems.
ATT7	Using this methodology is beneficial.
ATT8	*Using this methodology is a foolish idea.
ATT9	I like the idea of using this methodology to perform my job.
ATT10	*Using this methodology is unpleasant.

***Denotes reverse-coded item**

Appendix E Survey Items

Product Usefulness	
PROD1	Using the Rational Unified Process helps to develop more efficient systems.
PROD2	Using the Rational Unified Process helps to develop better systems.
PROD3	Using the Rational Unified Process helps to develop more reliable systems.
PROD5	Using the Rational Unified Process helps to develop higher quality systems.
PROD6	Using the Rational Unified Process helps to develop more maintainable systems.
Process Usefulness	
PROC2	Using the Rational Unified Process helps to keep the development project on schedule.
PROC4	Using the Rational Unified Process improves schedule performance.
PROC5	Using the Rational Unified Process helps to meet scheduled delivery dates.
PROC6	Using the Rational Unified Process results in a more efficient development process.
PROC8	Using the Rational Unified Process increases the productivity of the development process.
Communication Usefulness	
COMM1	Using the Rational Unified Process improves communication with the user's manager(s).
COMM2	Using the Rational Unified Process improves communication with managers.
COMM3	Using the Rational Unified Process improves communication with users.
COMM4	Using the Rational Unified Process improves communication with other team members.
Career Usefulness	
CAREER2	Using the Rational Unified Process increases my opportunity to gain job security.
CAREER5	Using the Rational Unified Process increases my chances for pay increases.
CAREER6	Using the Rational Unified Process increases my chances for promotion.
CAREER7	Using the Rational Unified Process improves my marketability.
CAREER9	Using the Rational Unified Process increases my flexibility of changing jobs.
Symbolic Usefulness	
SYMB5	Using the Rational Unified Process indicates to others that sound development practices are being used.
SYMB10	Using the Rational Unified Process shows others that professional development practices are being used.
SYMB11	Using the Rational Unified Process shows others that industry standard systems development practices are being used.
SYMB12	Using the Rational Unified Process indicates to others that systematic systems development practices are being used.
SYMB13	Using the Rational Unified Process signifies to others that accepted systems development practices are being used.
Support Usefulness	
ANX1	Using the Rational Unified Process reduces developer stress during systems development.
ANX3	Using the Rational Unified Process helps developers deal with the pressures of systems development.
ANX5	Using the Rational Unified Process relieves developer apprehension during systems development.

ANX8	Using the Rational Unified Process helps developers cope with the stress of systems development.
ANX9	Using the Rational Unified Process reduces developer anxiety during systems development.
Defense Usefulness	
DEF10	The Rational Unified Process can be used to justify decisions made during systems development.
DEF12	If a decision made during systems development turns out to be wrong, the Rational Unified Process can be used to show the rationale for the decision.
DEF13	Using the Rational Unified Process provides protection from unreasonable demands.
DEF5	The Rational Unified Process can be used to authorize decisions made during systems development.
DEF6	Using the Rational Unified Process defends against unreasonable demands.
DEF9	Using the Rational Unified Process protects against unreasonable demands for additional system functionality.
Perceived Ease of Use	
PEOU1	Learning the Rational Unified Process was easy for me.
PEOU3	I think the Rational Unified Process is clear and understandable.
PEOU4	It was easy for me to become skillful at using the Rational Unified Process.
PEOU5	Using the Rational Unified Process does not require a lot of mental effort.
PEOU6	I find the Rational Unified Process easy to use.
PEOU7	Using the Rational Unified Process does not take too much time from my normal duties.
PEOU8	The Rational Unified Process is not cumbersome to use.
Compatibility	
COMPAT1	Using the Rational Unified Process is compatible with all aspects of my work.
COMPAT2	The Rational Unified Process is compatible with the way I develop systems.
COMPAT3	Using the Rational Unified Process is completely compatible with my current situation.
COMPAT4	Using the Rational Unified Process fits into my work style.
COMPAT5	Using the Rational Unified Process fits well with the way I work.
Self Efficacy	
SE1	I could develop an information system if there was no one around to tell me what to do as I go.
SE11	I could develop an information system if the Rational Unified Process was not available to tell me what to do.
SE12	I could develop an information system if I had only the Rational Unified Process for reference.
SE14	I could develop an information system without following the Rational Unified Process.
SE2	I could develop an information system if I had never developed one like it before.
SE3	I could develop an information system if I could call someone for help if I got stuck.
SE4	I could develop an information system if someone else helped me get started.
SE5	I could develop an information system if I could have a lot of time to create it.
SE6	I could develop an information system if someone showed me how to do it first.
SE7	I could develop an information system if I had built a similar information system in the past.
SE8	I could develop an information system if I had seen someone develop one before developing one myself.
Perceived Usefulness (TAM)	
PU1	Using the Rational Unified Process increases my productivity.
PU2	Using the Rational Unified Process improves my job performance.
PU3	Using the Rational Unified Process enhances the quality of my work.
PU4	The advantages of using the Rational Unified Process outweigh the disadvantages.

PU5	Using the Rational Unified Process makes it easier to do my job.
PU6	The Rational Unified Process is useful in my job.
Voluntariness	
VOL1	*My superiors expect me to use the Rational Unified Process.
VOL2	My boss does not require me to use the Rational Unified Process.
VOL3	*The use of the Rational Unified Process is required for my job.
VOL4	Although it might be helpful, using the Rational Unified Process is certainly not mandatory in my job.
VOL5	My use of the Rational Unified Process is voluntary (as opposed to being required).
Attitude	
ATT1	Using the Rational Unified Process is a good idea.
ATT10	*Using the Rational Unified Process is unpleasant.
ATT5	Using the Rational Unified Process is a wise idea.
ATT7	Using the Rational Unified Process is beneficial.
ATT8	*Using the Rational Unified Process is a foolish idea.

***Denotes reverse-coded item**

VITA

David L. Henderson III

Virginia Polytechnic Institute and State University

Pamplin College of Business • Accounting and Information Systems

EDUCATION

Virginia Tech, Blacksburg, VA **Fall 2007**
Ph.D., Business, Accounting
George Washington University, Washington, DC **May 2000**
M.S., Information Systems Technology
Mary Washington College, Fredericksburg, VA **May 1995**
B.A., Economics; B.S., Business Administration

TEACHING EXPERIENCE

Virginia Tech, Blacksburg, VA **August 2003–Present**
Teaching Assistant

- Taught one section of **Managerial Accounting.** Lectured on managerial accounting topics such as variance analysis, overhead allocation, and budgeting.
- Taught multiple sections of **Introduction to Business Information Systems**. Lectured on major topics in the field of information systems such as computer networking, electronic commerce, database development, systems analysis and design, and programming.
- Taught and managed all aspects (selected book and developed syllabus, lecture notes, and homework projects) of **Object-Oriented Systems Design**. Lectured on technical topics such as object-oriented design, systems architecture, database design, and user-interface design.
- Taught one section of **Introduction to Personal Computers.** Lectured on various Microsoft Excel topics including Visual Basic for Applications (VBA) and pivot tables.

Trinity University, Washington, DC **July 2002–August 2004**
Adjunct Instructor

- Taught and managed all aspects (selected book; developed syllabus, lecture notes, and homework projects; and secured guest speakers) of **Decision Support Systems** class for adult and undergraduate students**.** Lectured on topics such as the types of decision support systems; competitive advantage and decision support systems; how to implement decision support systems, how to use Visual Basic to develop decision support systems.
- Taught and managed all aspects (selected book; developed syllabus, lecture notes, and homework projects; and secured guest speakers) of **Information Technology Project Management** class for adult and undergraduate students. Lectured on topics such as using Microsoft Project to develop schedules; earned value management; critical path analysis; and cost estimation.
- Taught and managed all aspects (selected book and developed syllabus, lecture notes, and homework projects) of **Introduction to Microsoft Office 2000** class for adult and undergraduate students. Conducted hands-on lectures covering Microsoft Excel, Microsoft Word, and Microsoft PowerPoint.

George Washington University, Washington, DC **September 2001–June 2002**
Adjunct Instructor

- Taught and managed all aspects (selected book and developed syllabus, lecture notes, and homework projects) of **Introduction to Visual Basic** class for adult students. Lectured on topics such as active data objects, arrays, loops, variables, and case structures.
- Taught and managed all aspects (selected book and developed syllabus, lecture notes, and homework projects) of **Structured Systems Analysis and Design** class for adult students. Lectured on topics such as data-flow diagrams, entity-relationship diagrams, and systems development methodologies.

RESEARCH

Research interests

Software metrics • software development methodologies • requirements engineering • Sarbanes-Oxley

Dissertation

Explaining Developer Attitude Toward Using Formalized Commercial Methodologies: Decomposing Perceived Usefulness
Advisors: Dr. Steven Sheetz and Dr. France Belanger

Under review

Henderson, D., Sheetz, S., Wallace, L. (2006). *Understanding and Explaining Software Metric Use.* Submitted to *Communications of the ACM.*

Conference presentations

- Sheetz, S., **Henderson, D.,** Wallace, L. (2005). *Understanding Manager and Developer Perceptions of the Relative Advantage, Compatibility, and Complexity of Function Points and Source Lines of Code.* **Nominated for best paper.** Presented at the *Eleventh Americas Conference on Information Systems* (2005).
- Reed, T., Lin, H., **Henderson, D.,** Jenkins, G. (2006). *Sarbanes-Oxley (SOX) Implementation at a Fortune 50 Company: Perceptions from the Field.* Presented and received **"Best Student Paper"** at the *2006 Accounting Information Systems (AIS) Educators Conference.*
- **Henderson, D.** (2007). *Explaining Developer Attitude Toward Using Formalized Commercial Methodologies:Decomposing Perceived Usefulness.* Presented and received **"Best Student Paper"** at the *2007 Accounting Information Systems (AIS) Educators Conference.*

Working papers

- Tegarden, D., Sheetz, S., **Henderson, D.** *Using Causal Mapping to Reveal Cognitive Diversity within an Academic Department.*
- Sheetz, S., **Henderson, D.,** Wallace, L. *Understanding the Relative Advantage, Compatibility, and Complexity of Function Points Versus Source Lines of Code: A Comparison of Developers and Managers.*
- Sheetz, S., **Henderson, D.,** Wallace, L. *Why do Function Points Users Adopt Function Points?*

Service to research community

- Reviewed two papers for the 2005 and 2006 Hawaii International Conference on System Sciences
- Reviewer for *Information Technology and People* journal
- Reviewer and program committee member for the International Information Business Management Association's 2005 and 2006 conferences
- Reviewed two papers for the 2006 International Conference on Information Systems (ICIS)
- Reviewed two papers for 2007 Americas Conference on Information Systems (AMCIS)

Invited presentation

Understanding Manager and Developer Perceptions of the Relative Advantage, Compatibility, and Complexity of Function Points and Source Lines of Code. Virginia Commonwealth University (2005).

HONORS AND AWARDS

- “Best Student Paper” Award (2006 and 2007 AIS Educators Conference)
- AMCIS Doctoral Consortium (2006)
- Best Paper Nomination for *Understanding Manager and Developer Perceptions of the Relative Advantage, Compatibility, and Complexity of Function Points and Source Lines of Code* (AMCIS 2005)
- Phi Kappa Phi Honor Society (April 2006)
- Floyd A. Beams Accounting Scholarship (2004, 2006)
- John E. Peterson Accounting Fellowship (2005)

UNIVERSITY SERVICE

Pamplin College of Business Research Committee

PROFESSIONAL MEMBERSHIPS

Association for Information Systems • American Accounting Association

PROFESSIONAL EXPERIENCE

Electronic Data Systems, Washington, DC **September 2001–August 2003**

Advanced Systems Engineer

- Served as project manager for second phase of internal accounting system designed to support Immigration and Naturalization Service (INS) contract.
- Managed all aspects of the development effort. Developed project schedules, data migration, cost estimates, requirements, and relational database design.

Siebel Systems, Reston, VA **November 2000–September 2001**

Siebel Consultant

- Served as lead consultant for application and unit testing of Siebel software implementation for BellSouth.
- Created detailed sales and marketing reports. Defined requirements via Joint Application Design sessions with end-users.

Electronic Data Systems, Herndon, VA **April 1995–November 2000**

Systems Engineer ***April 2000–November 2000***

- Served as project manager on a development effort for an internal accounting system designed to support a contract with the INS.
- Managed all aspects of the development effort including developing project schedules, cost estimates, defining requirements, relational database design, programming and testing.

Technical Associate ***December 1998–April 2000***

- Served as project manager on a development effort for an internal accounting system designed to support a contract with the INS.
- Operated and maintained existing accounting system for a contract with the INS.

Business Analyst ***October 1997–December 1998***

- Performed Y2K analysis and remediation on INS legacy financial systems.
- Graduated in top 10% of Systems Engineering Development (SED) Program and selected as Assistant Instructor for next class.

Financial Analyst ***April 1995–October 1997***

- Served as lead General and Administrative and Overhead expenses financial analyst.
- Analyzed and monitored monthly financial outlook and budget data.

COMPUTER SKILLS

Languages

Visual Basic 6.0 • Active Server Pages • ASP.Net • Visual Basic.Net • Visual Basic for Applications (VBA) • JavaScript • Java • SQL • T-SQL • HTML

Database and Reporting Applications

Microsoft Access (2000, XP) • SQL Server (7.0, 2000) • Active Reports • Crystal Reports