ACCEPTANCE OF SYSTEMS DEVELOPMENT METHODOLOGIES: TESTING A THEORETICALLY INTEGRATED MODEL

by

NANCY A. BONNER

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

August 2008

ACKNOWLEDGEMENTS

First, I would like to acknowledge the continual support and encouragement that my husband, Dave, has provided throughout this long and sometimes arduous process. Without his love and steadfast belief in me, this journey would not have been possible. My children have also been a constant source of love and although they have not always understood exactly what I do when I am away at school, or furiously pounding away at my keyboard, they have always made me feel that they are immensely proud of my accomplishments while at the same time, keeping me grounded.

Although the past few years have been challenging, I have learned a tremendous amount under the guidance of gifted scholars such as Dr. James T.C. Teng, Dr. Sridhar Nerur, Dr. Radha Mahapatra, Dr. Wendy Casper and Dr. Mark Eakin. They have been extremely encouraging and I feel incredibly lucky to have them on my committee. I also want to thank Dr. Greg Frazier for continually striving to ensure that we have an excellent program and for his help in facilitating this process. I would be remiss not to acknowledge the help of two very capable members of our UTA staff, Rita Delmar and Andrea Webb, who are always available to help students navigate the program.

Finally, I cannot say enough to acknowledge the gratitude I feel toward my committee chairs, Dr. Teng and Dr. Nerur, both of whom have spent hours ensuring that my research and resulting dissertation are of the highest quality. They have persistently maintained a high standard and tenaciously worked to ensure that I am putting forth my best effort. Their keen insight and vast knowledge have been invaluable.

July 15, 2008

ABSTRACT

ACCEPTANCE OF SYSTEMS DEVELOPMENT METHODOLOGIES:

TESTING A THEORETICALLY

INTEGRATED MODEL

Nancy A. Bonner, PhD.

The University of Texas at Arlington, 2008

Supervising Professor: Dr. J.T.C. Teng and Dr. S. Nerur

Industry reports continue to portray a dismal picture of software development project success. Software projects continue to be over budget, overdue and lacking in quality and desired functionality. A variety of innovative methodologies and tools have been introduced to aid in improving the software development process. However, there has been reluctance to commit to their usage. The goal of this research is to examine the reasons a software developer makes a commitment to use a given development methodology.

We used a theoretically integrated approach to study this phenomenon, drawing on theories from the fields of marketing as well as psychology. This thesis is in the form of a three essay structure.

In the first essay, we developed an instrument to measure process agility and applied the instrument in an empirical study. The second essay investigated the impact of process agility as well as specific innovation characteristics on a developer's commitment to using the methodology. Applying the psychological theory of Self-Determination, we explored whether there are innate psychological needs that mediate the relationship between these factors and an individual's commitment to using the methodology. The third summarizes the results of the second study for practitioners, providing industry with information to aid in the implementation of new development methodologies and processes.

This research makes significant contributions. First, it conceptualizes, operationalizes, and develops a standard measure of process agility in the context of software development methodologies. Second it applies the process agility instrument to investigate the relationship between the agility of a methodology and a developer's motivation to be committed to the methodology thus presenting empirical support for a positive relationship between process agility and developer motivation. Third, it builds upon prior research by integrating Self-Determination Theory (SDT) into current usage models, thereby providing an explanatory link between Diffusion Theory with its associated factors and individual commitment to usage. This contributes toward a deeper understanding of the underlying reasons for the effects of certain factors on technology implementation as well as making significant progress toward the development of a model for determining individual developers' intention to commit to and support the use of a given software development methodology. Fourth, this research provides insight and helpful diagnostics to facilitate and assist practitioners in their efforts to implement new software development methodologies. Finally, it gives direction for future research in terms of employing the agility measurement instrument to examine other phenomena such as project outcomes as well as providing for further validation of the Software Development Methodology acceptance model.

TABLE OF CONTENTS

Appendix

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Objective

The objective of this research is to investigate the reasons a software developer is committed to a given development methodology. While use of the methodology may be mandatory, a developer's commitment to its use can contribute greatly to a successful implementation. Conversely, a developer's indifference or lack of commitment can lead to an unsuccessful implementation. Therefore, the dependent variable of interest is the developer's commitment to using the methodology.

We used a theoretically integrated approach to study this phenomenon, drawing on theories from the fields of marketing as well as psychology. This thesis is in the form of a 3 essay structure. The next section discusses the motivation for this research, the specific research questions and a general description of the three essays. We conclude with a section summarizing the contributions of this thesis.

1.2 Motivation

Historically, software projects have been over budget, overdue and lacking in quality and desired functionality. According to a 2003 Standish Group Chaos Report, time overruns increased to 82% from a low of 63% in 2000 and research shows only 52% of required features and functions make it to the released product (as compared with 67% in 2000). And, of the $2.5 trillion spent on IT during 1997 – 2001, almost $1 trillion went toward underperforming projects (Benko and McFarland, 2003).

Various potential reasons have been cited for this lack of success, but there is one point of agreement. Something definitely appears to be "broken" in the software development process indicating that there may be a better way to engineer software. A multitude of innovative methodologies and tools have been developed to aid in improving the success rate of software development projects. However, there has been a reluctance to commit to the use of these new methods (Glass, 1999; Hardgrave, 1995; Kemerer, 1992). Therefore, our first research question is: What characteristics and factors are associated with an individual's motivation and commitment to the use of a given development methodology? Secondly, are there innate psychological needs that explain how and why these factors impact commitment?

One factor we explored is the "agility" of the methodology. In recent years, a host of new "agile methodologies" has arrived on the scene. And, although implementation of agile methodologies has been slow, their usage has steadily grown indicating that organizations are increasingly accepting this new approach. In a recent survey, 69% of respondents said that their organizations were using agile methods (Ambler, 2007). The Agile Manifesto specifies four basic tenets of agile software development (AgileAlliance, 2001). It was written in 2001 by the Agile Alliance and captures the essence of a truly agile development process. Specifically, the Agile Manifesto emphasizes individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation and responding to change over following a plan. While the proponents of agile acknowledge that the items on the right are important, they aver that the ones on the left are even more valuable. However, we suggest that there are different "degrees" or "levels" of agility within *any* development methodology. In this research we measured the level of agility of the development methodology being used. To our knowledge, an instrument to measure process agility in the context of software development does not currently exist. Therefore, part of this research entailed conceptualizing the construct, and developing and validating an instrument to measure it. While agility has been examined in areas such as manufacturing, it is yet to be empirically explored in the IS literature.

1.3 Overview of The Three Essays

In the first essay, we developed and validated an instrument to measure process agility. Modeled after Moore and Benbasat's development of the PCI, the measurement development process consisted of 3 stages: 1) item creation, 2) scale development and 3) instrument testing.

In the second essay, we investigated the impact of process agility as well as specific innovation characteristics on a developer's commitment to using the methodology. Applying the psychological theory of Self-Determination, we explored whether there are innate psychological needs that mediate the relationship between these factors and an individual's commitment to using the methodology.

The third essay summarizes the results of the empirical study. In it, we summarized the results of the second study and discussed their implications for practitioners. This essay provides industry with information to aid in the implementation of new development methodologies and processes.

1.4 Contributions of this Thesis

This research makes significant contributions. First, it conceptualizes, operationalizes, and develops a standard measure of process agility in the context of software development methodologies. Second it applies the process agility instrument to investigate the relationship between the agility of a methodology and a developer's motivation to be committed to the methodology thus presenting empirical support for a positive relationship between process agility and developer motivation. Third, it builds upon prior research by integrating Self-Determination Theory (SDT) into current usage models, thereby providing an explanatory link between Diffusion Theory with its associated factors and individual commitment to usage. This contributes toward a deeper understanding of the underlying reasons for the effects of certain factors on technology implementation as well as making significant progress toward the development of a model for determining individual developers' intention to commit to and support the use of a given software development methodology. Fourth, this research provides insight and helpful diagnostics to facilitate and assist practitioners in their efforts to implement new software development methodologies. Finally, it gives direction for future research in terms of employing the agility measurement instrument to examine other phenomena such as project outcomes as well as providing for further validation of the Software Development Methodology acceptance model.

CHAPTER 2

DEVELOPING AN INSTRUMENT TO MEASURE PROCESS AGILITY IN SOFTWARE DEVELOPMENT

2.1 Introduction

A global study conducted by IBM in 2004, revealed that agility and responsiveness are uppermost in the minds of CEOs. In-depth interviews were conducted with 765 CEOs from across the world and the results indicate that they are embracing change and using innovation to their advantage (IBM, 2006). The ability to respond efficiently and effectively to change is a common characteristic of agility, whether discussing an organization, a system or a person. Within the research literature, one can find numerous definitions of the term "agility". Agility has been defined as:

> ….the ability of an organization to sense environmental change and to respond efficiently and effectively to it… (McCoy and Plummer, 2006-Gartner Group);
>
> ...the primary dimension of agility is the ability of a system to respond to change (Arteta and Giachetti, 2004);
>
> …continuous close coordination between business and IT people to respond effectively to constantly changing situations…(Hugos, 2007).

In a software development environment, where user requirements may be continually changing and technology is evolving, truly agile development methodologies could provide an effective approach to increasing the success rate of software projects.

Growing frustration with traditional development approaches led to the articulation of the Agile Manifesto in 2001 (AgileAlliance, 2001). In accordance with the Agile manifesto, agile methodologies emphasize people over process, software over documentation, customer collaboration over contract negotiation and responding to change over following a plan (Cockburn and Highsmith, 2001; Martin, 2002).

Agile approaches encourage developers to "embrace change". Highsmith (2002) opines that "Agilists welcome change as an opportunity to help the customer respond to marketplace turbulence" (Highsmith, 2002). In his book *Agile Software Development Ecosystems*, Highsmith emphasizes the similarities between ecosystems ("organisms and their environment") and the software development environment. This perspective views software systems as being very similar to living things that must adapt to constantly changing surroundings and situations. In such a scenario, planning too far ahead can be counterproductive because constant change will cause long term plans to become obsolete. Rigorous processes and inflexible plans discourage change. While traditional approaches value these processes and technology over people, agile methodologies place a premium on individuals and their interactions (Highsmith, 2002). This emphasis has resulted in agile development processes being associated with increased developer motivation and commitment (Asproni, 2004). Boehm and Turner

(2004), claim that customers, developers and organizational culture have a significant influence on the success of a project.

Boehm (2004) explains that "Plan-driven methods are characterized by a systematic engineering approach to software that carefully adheres to specific processes in moving software through a series of representations from requirements to finished code". In plan-driven methods, comprehensive documentation is required during each phase of the project. Plan-driven methods can be implemented using a waterfall approach where a given phase must be completed prior to beginning the next phase or using an incremental approach where the entire system is designed, but coded and tested incrementally. In either case, it is necessary to provide comprehensive documentation and traceability of requirements. The processes are highly standardized to make them repeatable and amenable to continuous improvement. Typically, developers are skilled in a particular area and assigned tasks within their area of specialization. The project manager plans, estimates, schedules and assigns tasks at the beginning of the project. Any changes to this plan must follow a well-defined change control process. Thereafter, the manager is in charge of project monitoring and control in addition to acting as liaison to the client and upper management (Conger, 1994). Agile methods differ from traditional methods in terms of team structure as well as development process. "A truly agile method must include all of the following attributes: iterative (several cycles), incremental (not deliver the entire product at once), self-organizing (teams determine the best way to handle work), and emergence (processes, principles, work structures are recognized during the project rather than predetermined)" (Boehm, 2004).

Several studies have provided empirical support for an association between agile practices and a higher quality software product (Nosek, 1998, Kessler, 1999, Williams, 2000, Upchurch, 2001). In fact, it has been shown that implementing even just one agile practice can improve quality, satisfaction and productivity without a significant change in cost (Parsons *et al.*, 2007). In this study it was also shown that organizations tend to combine several techniques from various methodologies as opposed to adopting a single methodology (Parsons *et al.*, 2007). This supports the idea that there are different "degrees" or "levels" of agility within any development process. Also, some organizations are habitually process-intensive while others are not. For example, some have embraced CMM or ISO standards and therefore follow rigid steps to carry out their tasks. Hence, you could have different levels of agility even though the same agile techniques are practiced. While the (Parsons *et al.*, 2007) study mentioned above, looked at various combinations of techniques, it would be extremely valuable if an organization could measure the level of agility of the development process being followed. Without this ability, organizations do not really know the extent to which their method is truly "agile". Our research conceptualized, developed and validated an instrument to measure the agility of a given software development methodology.

In the next section, we provide background information associated with the conceptualization of agility in other domains and discuss how it is related to the concept of software development process agility. This is followed by a description of the research method. Next, we describe the instrument development process and present the results of the instrument development portion of the research. In the subsequent sections, we describe a study that validates the predictive ability of the agility measurement instrument. We conclude with a discussion of the findings, limitations, contributions and future research.

2.2 Background

The idea of agility within the context of business was conceptualized in an industry-led study by the Iaccoca Institute. Capabilities of an agile manufacturing enterprise include the ability to introduce new products and respond to customer needs quickly through "innovative management structures, flexible technology and a skill base of knowledgeable workers" (Nagel and Dove, 1991). With agile manufacturing, the focus is shifted from mass production to more customized products. Competitive capabilities considered to be agility drivers have included proactiveness, responsiveness, competence, flexibility, quickness, focusing on the customer and forming a partnership with suppliers. Using these drivers as the basis, (Zhang and Sharifi, 2007) proposed a framework for the implementation of agility as a manufacturing strategy and developed a numerical taxonomy of agility strategies using the framework. While agility originated within manufacturing research, its scope has widened to encompass other aspects of an organization such as agile supply chains (Swafford, 2003) and agile workforces (Breu *et al.*, 2001). Research has included development and validation of instruments to measure these constructs. (Swafford, 2003; Breu *et al.*, 2001). Table 2.1 provides an overview of characteristics of various types of agility.

Table 2.1 Characteristics of Agility in Other Domains

Construct	Characteristics	Citation
Customer Agility	Co-opt customers in exploration and exploitation of innovations opportunities	Sambamurthy, *et al*, 2003
Design Flexibility (in Product Development Agility)	High design flexibility is associated with rapid decisions on critical changes, ability to run quick, test-driven design iterations, embracing change	Ettlie, 1998; Thornke, 1997
Enterprise Agility	Ability to sense environmental change and respond efficiently and effectively to that change	Gartner, 2006
Manufacturing Agility	Responding to change in proper ways and due time; Exploiting changes and taking advantage of them as opportunities	Sharifi and Zhang, 1999
Operational Agility	Ability to accomplish speed, accuracy, and cost economy in the exploitation of innovation opportunities	Sambamurthy, *et al*, 2003
Organizational Agility	Deliver value to the customer, Be ready for change, Value human knowledge and skills; Form virtual partnerships	Katayama and Bennett, 1999
Partnering Agility	Leverage assets, knowledge and competencies of suppliers, distributors, contract manufacturers and logistics providers in exploration and exploitation of innovation opportunities	Sambamurthy, *et al*, 2003
Supply Chain Agility	Promptness with and degree to which firm can adjust its supply chain speed, destinations and volume	Prater, *et al*, 2001
Value Chain Agility	Adapt in a timely manner to a quickly changing competitive environment to provide products and services.	Sharifi and Zhang, 1999
Workforce Agility	Defined 5 capabilities: Intelligence, competencies, collaboration, culture and information systems	Breu, *et al.*, 2001; Oyen, *et al.*, 2001

In reviewing Table 2.1, it is apparent that agility does not have a consistent meaning across domains. Although a common theme is the ability to respond to change, the specific characteristics have been customized to fit the parameters of the construct's environment. Some of the agility construct characteristics from other domains are applicable to software process agility (e.g. rapid decisions on critical changes and test-driven design iterations) and others are not (e.g. ability to adjust supply chain speed). Next we will present a conceptual model of agility developed as a framework for achieving agility in manufacturing organizations. However, parts of it can be applied to the software development process.

Figure 2.1, adapted from Sharifi and Zhang (1999), depicts a conceptual model of agility. The model consists of three main parts, agility drivers, agility capabilities, and agility providers. Agility Drivers are those changes in the business environment that compel the company to seek a new position in order to gain competitive advantage. Conboy and Fitzgerald (2004) (pp. 41-43), identified five categories of agility drivers, (competition, customers, technology, social factors, and overhead), and illustrated their applicability to information systems development. Agile Capabilities provide the essential competencies for responding to changes. Agile development processes strive to deliver working software in short iterations by anticipating and embracing change. Frequent delivery of working software ensures quick responsiveness to changing requirements. Agile development methods are characterized by flexibility of team members' roles and the willingness to improvise and adapt. The Agile Providers are the "means" of achieving the capabilities. Agile development methods are categorized as being Agility Providers (i.e. Practices, Methods, Tools).

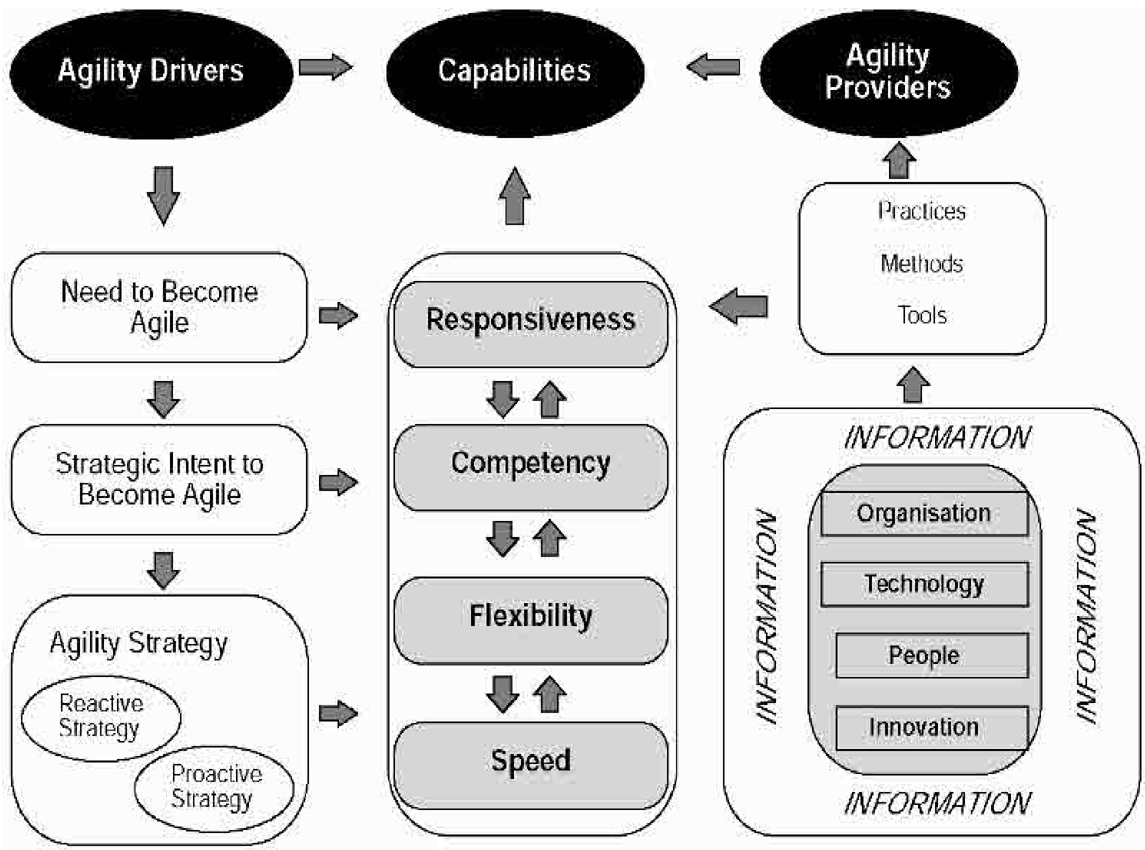


Figure 2.1 Conceptual Model of Agility (adapted from Sharifi & Zhang, 1999)

As mentioned earlier, the Agile Manifesto describes the four basic tenets of agile software development. There are 12 principles behind the Agile Manifesto and these principles describe practices and beliefs that embody these four tenets. The agility measurement instrument is intended to be a generic measure of software development process agility, meaning that we did not want to mirror any specific development methodology (e.g. Scrum or XP). However, we did want to ensure that we were truly capturing aspects of development process agility. The Agile Manifesto and its 12 principles were written by a group of experts in the area of agile software development. Therefore, we chose to use the 12 principles behind the Agile Manifesto as the basis for developing the measurement items.

2.3 Method

2.3.1 Instrument Development

To the extent possible, development of the Process Agility Measurement Scale was modeled after the procedure used by Moore and Benbasat (1991). However, one major difference in their study was the fact that Moore and Benbasat's initial pool of items was taken from existing instruments. To our knowledge, such a scale doesn't currently exist for measuring development process agility. Therefore, it was necessary to create the initial pool of items based on: a) academic and industry literature related to development process agility; and b) the domain knowledge provided by, researchers in the field.

Consistent with the Moore and Benbasat study, the instrument development process consisted of 3 stages: 1) item creation; 2) scale development; and 3) instrument testing. The goal of the item creation step is to ensure content validity. A pool of items was developed based on the following 12 principles behind The Agile Manifesto (http://agilemanifesto.org/principles.html).

1) *Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.*

2) *Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.*

3) *Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.*

4) *Business people and developers must work together daily throughout the project.*

5) *Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.*

6) *The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.*

7) *Working software is the primary measure of progress.*

8) *Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.*

9) *Continuous attention to technical excellence and good design enhances agility.*

10) *Simplicity--the art of maximizing the amount of work not done--is essential.*

11) *The best architectures, requirements, and designs emerge from self-organizing teams.*

12) At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

The process agility construct was identified as a formative construct. Four decision rules to determine if a construct is formative or reflective are outlined in (Petter, *et al*., 2007). According to these four rules, for a construct to be considered formative: 1) the direction of causality is from items to construct, 2) the indicators need not be interchangeable, 3) the indicators need not covary with each other and 4) the nomological net for the indicators may differ. Using the following two characteristics of process agility, we will explicate the reasoning for the formative designation. Agile development processes are characterized by 1) flexibility of team members' roles and relationships, and 2) dynamically evolving requirements specification through continuous feedback from users. First, a change in the degree of flexibility of team members' roles and relationships will result in a reduction in the degree of process agility. Conversely, if the degree of agility changes, it is not necessarily true that the flexibility of roles and relationships would change. Second, it is obvious that the two characteristics described above are not interchangeable; they capture two different features of process agility. Third, although the two aforementioned characteristics may covary, it is entirely possible that they may not. For example, a high degree of flexibility in team members' roles and relationships may be associated with being able to accommodate dynamically evolving requirements better; however, this high degree of flexibility may be present without a high level of dynamically evolving requirements. Fourth, these two attributes of process agility may have different antecedents and consequences. For example, one antecedent of team member role flexibility is the presence of team members cross trained in multiple areas and one antecedent of dynamically evolving requirements is the establishment of continuous communication and feedback between developers and users.

For formative constructs, it is especially critical to establish content validity because they are defined by the dimensions that comprise them (Petter, *et a*l., 2007). The agility construct scale items went through several iterations of review and refinement by academics knowledgeable in survey design as well as software methodologies. Some of the initial items attempted to measure multiple aspects with a single question and were deemed to be too complex. Therefore, these were broken into more than one question to ensure that each item was referring to only one aspect. Conversely, in cases where an aspect of agility may not have been covered thoroughly, new items were added. Any wording that appeared to provide a judgmental assessment of a particular agile principle was changed so that subjects would not be influenced by the perceived "virtues" of a given characteristic. This resulted in a total 14 items. Each item was in the form of a statement and the respondent was asked to indicate their level of agreement based on a seven-point Likert scale, ranging from "strongly disagree" to "strongly agree".

The purpose of the scale development stage is to assess construct validity and ensure items are clearly worded in order to maximize respondents' understanding of the statement. This was accomplished by forming a panel of 4 experts from industry to review the items. These experts had extensive software development methodology experience with a combined total of over 80 years in the software development industry. They rated each item on understandability and sorted them into categories based on their opinion of what each was measuring. There was general agreement in terms of both item clarity and categorization of items. As a result, only minor changes were necessary.

Instrument testing consisted of both a pilot as well as a field test. The purpose of the pilot test was to ensure mechanics of the questionnaire were sufficient and to calculate an initial reliability assessment. Pilot test respondents consisted primarily of developers from industry who closely resemble the target population. Several of the respondents were academics who had prior industry experience. A total of 20 respondents completed the questionnaire which was administered online. The last section of the questionnaire asked respondents to comment on and provide any suggestions they had related to the questionnaire. Based on these comments, minor changes were made related to the wording of 2 of the questions and several of the instructions, the number of questions per screen, and the highlighting of key words in the instructions. An initial assessment indicated adequate discriminant and convergent validity among the items. Therefore, no items were dropped from the survey.

2.3.2 Sample

For the field test, an online survey was conducted via the internet. Respondents were software developers working in teams and using a software development methodology. CIO’s and other top executives from several organizations were contacted via email and telephone and asked if they would be willing to have their developers participate in the survey. The purpose of the study was explained along with a link to the online survey. Respondents could click on the link which opened an introductory page to the survey. The introductory page contained the UTA logo and a statement assuring their privacy. It was stated that participation was purely voluntary.

It is difficult to determine exact response rates for each organization as they were varied; however, the overall number of potential respondents was approximately 8,000, out of which 554 completed the survey. This represents a response rate of 6.92%. However, large chunks of data were missing from 75 of the surveys, rendering them unusable even for purposes of reporting demographic data. Therefore, they were excluded from the analysis, leaving us with a total of 479 usable surveys. Respondents' average number of years with the organization was 2.88. Average number of years of development experience was 6.47. Additional demographic data for the respondents is illustrated in Table 2.2.

Table 2.2 Demographic Profile of Respondents

MEASURE	ITEM	FREQ	PERCENT
Gender	Female	73	15.2
	Male	392	81.8
	No Response	14	2.9
Age	18 – 21 years	1	0.2
	21 – 30 years	261	54.5
	31 - 40 years	184	38.4
	41 – 50 years	15	3.1
	51 – 60 years	7	1.5
	No Response	11	2.3
Education	Associate Degree	4	0.8
	Bachelor Degree	281	58.7
	Master Degree	176	36.7
	Doctorate Degree	1	0.2
	High School Degree	3	0.6
	Other	4	0.8
	No Response	10	2.1
Industry	Banking/Insurance/Financial Service	12	2.5
	Construction/Architecture/Engineering	2	0.4
	Consulting/Business Service	65	13.6
	Government, including Military	2	0.4
	Health Care	4	0.8
	Hotel/Entertainment/Service Industry	4	0.8
	IT/Telecommunications	361	75.4
	Manufacturing	15	3.1
	No Response	14	2.9
Job Title	Business Analyst	14	2.9
	Programmer	24	5.0
	Programmer/Analyst	146	30.5
	Manager/Supervisor	81	16.9
	Project Team Leader	128	26.7
	No Response	86	17.9

The sample does seem to represent a broad selection of different industries, and various age groups. Although a majority of the respondents are male, this is the case for the overall population of programmers/analysts (Craig, *et al.*, 2002; Joshi and Schmidt, 2005). Most hold bachelor's or master's degrees, as expected. Thus, we have good reason to believe that the sample comes from many different segments of the population, and is not a biased sample from a particular segment of the population.

2.4 Results

2.4.1 Data Analysis

Factor analysis examines the interrelationships among items and attempts to explain them in terms of their common underlying dimensions. It is often used in the process of instrument development. An Exploratory Factor Analysis was performed on the 14 items contained in the agility survey. Principal Components Analysis with Oblimin rotation was performed. The agility construct is formative and because the objective of formative constructs is to preserve the unique variance of each measure, Principal Components Analysis is recommended to interpret the factors (Petter, *et al.*, 1995). This is because Principal Components Analysis extracts all of the variance whereas Common Factor Analysis extracts only the shared variance. Oblique rotation was chosen because it allows the factors to correlate whereas orthogonal rotations produce factors that are uncorrelated (Costello and Osborne, 2005). Although it is not a requirement that formative indicators covary, it was expected that there would be some correlation among the items in this study as we are trying to identify underlying dimensions of the agility construct.

Table 2.3 contains the instrument items with their associated means and standard deviations. The Kaiser-Meyer-Olkin (KMO) measure of sampling adequacy is equal to .880. The KMO measure of sampling adequacy is an index for comparing the magnitudes of the observed correlation coefficients to the magnitudes of the partial correlation coefficients. A value close to 1 is an indication that the correlation patterns are "relatively compact" and that a factor analysis would produce "distinct and reliable" factors (Field, 2005). Therefore, our data exhibits a high level of sampling adequacy. Another indicator of the strength of the relationship among variables is Bartlett's test of sphericity. It is used to test the null hypothesis that the variables in the population correlation matrix are uncorrelated. The observed significance level is .0000, therefore it is small enough to reject the null hypothesis. This indicates that the strength of the relationship among the variables is strong.

The Kaiser criterion was applied to the initial solution. This implied that 3 factors with eigenvalues greater than 1 should be extracted. In addition, examination of the scree plot suggested extraction of 3 factors as well. The first 3 factors accounted for a total of 55 percent of the variance. The rotated pattern matrix was used instead of the structure matrix because it represents the unique contribution of each factor to the reconstruction of any variable and provides a better basis for judging simple structure. As a guide to interpreting the weights in the pattern matrix, a .45 cutoff for the loading was required to retain the item with the factor. For sample sizes of 100 or larger (our sample size = 479) a cutoff point ranging from .30 to .50 is recommended with .40 suggesting the item is important and .50 ensuring practical significance. Because factor loading is the correlation of the variable and the factor, the squared loading is the amount of the variable's total variance accounted for by the factor (Hair, *et al.*, 1998). Therefore, a .45 cutoff will result in retention of items that account for at least 20 percent of the variable's total variance. After applying this criterion, item 10 was dropped from the analysis as it did not have a sufficient loading on any of the 3 factors. The remaining items with their associated loadings are shown in Table 2.4. Table 2.5 contains the component correlation matrix. The two factors are correlated at .299. This is sufficiently low for a formative construct.

Table 2.3 Items With Means and Standard Deviations

Item	Mean	Std. Dev.
We frequently develop working software that is tested, integrated and executable as a partial system.	4.91	1.437
Adjustments and refinements to requirements are always welcome at any stage of the development process.	4.60	1.527
We constantly seek users' feedback to shape new requirements and re-prioritize features of the system.	5.08	1.369
Our requirements specification process dynamically evolves through continuous feedback from users.	5.00	1.348
We meticulously document every aspect of the system throughout the development cycle.	2.88	1.360
Our initial system plan consists of minimal, yet essential requirements without complete and detailed specifications.	4.63	1.372
We believe changing requirements are normal and help to enhance the system quality.	4.77	1.482
Developers communicate and collaborate with business people continuously to incorporate their evolving requirements.	4.99	1.368
Our project schedules and estimates are determined up front and are not subject to change.	3.72	1.526
We improvise and experiment with new ways of doing things which may differ from the old routines.	4.94	1.180
The roles and relationships of our team members are flexible and not strictly defined.	4.59	1.459
We don't mind deviating from established processes and procedures as long as we continuously deliver working software.	4.14	1.604
We use short iterations of fixed intervals to quickly design, implement and test a small subset of the requirements.	4.56	1.342
Working software is the primary measure of progress.	4.81	1.354

Table 2.4 Items with Loadings

	1	2	3
Our requirements specification process dynamically evolves through continuous feedback from users.	.823		
We constantly seek users' feedback to shape new requirements and re-prioritize features of the system.	.797		
Developers communicate and collaborate with business people continuously to incorporate their evolving requirements.	.639		
We frequently develop working software that is tested, integrated and executable as a partial system.	.628		
Our initial system plan consists of minimal, yet essential requirements without complete and detailed specifications.	.567		
Adjustments and refinements to requirements are always welcome at any stage of the development process.	.552		
We believe changing requirements are normal and help to enhance the system quality.	.488		
We don't mind deviating from established processes and procedures as long as we continuously deliver working software.		.874	
The roles and relationships of our team members are flexible and not strictly defined.		.620	
Working software is the primary measure of progress.		.563	
We use short iterations of fixed intervals to quickly design, implement and test a small subset of the requirements.		.509	
*Our project schedules and estimates are determined up front and are not subject to change.			.763
*We meticulously document every aspect of the system throughout the development cycle.			.488

*****denotes reverse coded item

Table 2.5 Component Correlation Matrix

Component	1	2	3
1	1.000	.299	-.219
2	.299	1.000	-.089
3	-.219	-.089	1.000

2.4.2 Interpretation of Factors

After examination of the factors, it appears that the first two factors emerge as distinctive components of agility. The first factor appears to capture the dynamic aspect of an agile process that strives to deliver working software in short iterations by anticipating and embracing change, actively involving stakeholders and continually seeking users' feedback to address their evolving needs. This factor is thus labeled "Evolutionary Development" as the items capture the collaborative dimension of agile processes coupled with the adaptability and responsiveness to changing conditions. The second factor is labeled "Process Flexibility", as the items capture the reaction capabilities inherent in agile processes, and team members' roles and the willingness to deviate from established processes in order to deliver working software. It is the ability and willingness to improvise and come up with new solutions on the fly.

While the first two factors emerged as distinctive components of agile development processes and also exhibited high levels of reliability and internal consistency, the third factor did not. Both items that load on it are reverse coded items. We believed that there was sufficient reason to drop it from the scale. First, the component correlation matrix showed it negatively correlated with factors 1 and 2 which should not be the case. Prior to analysis, it was reverse coded so that it should be positively correlated with factors 1 and 2. Secondly, it only has two items and one of the items had a high negative loading on factor 1. Thirdly, it did not add significantly to the total variance explained. Without the third factor, total variance explained is 47%

and with it the total variance explained is 54%. Therefore, the two items loading on factor 3 were dropped from the final scale.

The finalized agility scale consists of 11 items which measure two underlying factors: Evolutionary Development and Process Flexibility.

In the next section we assess the predictive validity of the process agility scale.

2.5 Predictive Validity of the Process Agility Scale

In order to further validate the process agility scale and assess its predictive ability we used it to test hypotheses related to the effect of process agility on a developer's motivation to commit to the use of a given development methodology. It has been theoretically argued that process agility results in higher levels of developer motivation and commitment (Asproni, 2004). However, to the best of our knowledge, this has not been empirically tested.

The next section will present hypotheses for testing the predictive validity of the scale. Then the data analysis and results of this study will be presented.

2.5.1 Testing Predictive Validity of the Scale

The predictive validity of the scale was evaluated by testing its relationship to motivation and commitment (Figure 2.2) which, according to theories and past research findings, should be the outcome of process agility.

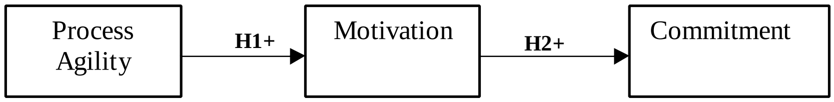


Figure 2.2 Testing the Predictive Validity of the Scale

In their landmark study that investigated motivation of information technology employees, Couger and Zawacki found that information systems workers had the highest growth need strength (Couger and Zawacki, 1980). Assessment of growth need strength was based on Core Job Theory which posits that meaningful jobs, those that challenge and stretch employees, are designed by focusing on the five core job dimensions of skill variety, task identity, task significance, autonomy and feedback. Skill variety refers to the degree to which the job requires a variety of different activities involving utilizing a number of different skills and talents of the employee. Task identity is the degree to which the job allows the completion of a whole and identifiable piece of work and structuring the work so the programmer has increased interaction with the business organization increases task identity. Task significance is the degree to which the job has a major impact on the lives or work of other people. Autonomy is how much the job provides freedom, independence and discretion to the employee. Feedback is related to the amount of information the employee receives about the effectiveness of their performance. These five dimensions are combined to produce a motivating potential score (MPS) and increasing the level of any of the five core job dimensions will result in an increased MPS (Zawacki, *et al.*, 1995). Therefore, jobs that are ranked high in the five core job dimensions should lead to an increase in information systems workers' motivation.

In an agile development environment, team members' roles are flexible and developers are cross trained (Highsmith, 2002; Martin, 2002). Because of this, agile developers' jobs involve a high degree of skill variety. Agile processes contribute to higher task identity because an iterative development environment results in frequent delivery of working software (Beck, 2000; Highsmith, 2002) allowing developers to complete identifiable pieces of work on a regular basis. Because agile developers have an ongoing collaborative relationship with users (Highsmith, 2002; Martin, 2002), they are able to see the impact of their work resulting in increased task significance. Autonomy is increased in an agile environment because developers are given the freedom to be flexible and deviate from established processes in order to deliver working software (Highsmith, 2002; Martin, 2002). And, feedback is increased due to the nature of iterative development and frequent collaboration with the users as well as among the developers (e.g. pair programming). Prior research has provided support for increased individual satisfaction when employees work in groups (Campion, *et al.*, 1993; Forsyth, 1999). Studies have also shown a positive association between the use of agile development methods and job satisfaction (Balijepally, 2006; Mannaro, *et al*, 2004; Williams, *et al*, 2000). And, increased job satisfaction is associated with higher levels of motivation (Couger and Zawacki, 1980; Herzberg, *et al.*, 1959; Locke, 1976).

We posit that the characteristics of agile development processes result in increased developer motivation. Therefore, the following hypothesis relates process agility to increased motivation.

Hypothesis 1 (H1). *Process Agility will be positively related to Motivation.*

Studies conducted in a work environment have shown that one of the consequences of self-determined motivation is a higher level of commitment on the job (Deci and Ryan, 2000; Gagne and Deci, 2005). Further, studies have shown that autonomous motivation is identified with increased affective commitment (Gagne and Deci, 2005). Affective commitment involves an individual identifying with and feeling an emotional attachment to the organization. This type of commitment involves accepting the organization's goals as one's own. Given that our focus is a developer's commitment to using a given methodology, this is of particular interest. In this case, one of the organization's goals would be developer's commitment to using the methodology. Therefore, an increased level of developer motivation would be associated with an increased level of commitment to using the methodology. Thus, we propose the following hypothesis.

Hypothesis 2 (H2). *Higher levels of motivation will be associated with higher levels of commitment.*

2.5.2 Constructs and Measurements

As described earlier, Evolutionary Development and Process Flexibility are two underlying dimensions of Process Agility. Process Agility is defined as an aggregate construct because it represents a composite of these two dimensions.

Dimensions of aggregate constructs are similar to formative measures in that they combine to produce the construct; however, they differ in that they are not observed variables. Rather, they are constructs identified as certain components of the aggregate construct they comprise. These aggregate constructs are usually operationalized by totaling scores on their dimensions (Edwards, 2001). Another way of calculating the aggregate construct is to assign dimension weights based on either correlations among the dimensions or by modeling the path weights from the first-order constructs to the second-order constructs (Edwards, 2001; Pavlou and El Sawy, 2006). In the case of Process Agility, we summed the weights and in doing so, each dimension was equally weighted. This method was chosen because when modeled, the paths were fairly equal (.553 and .590), indicating that each component contributed a relatively equal portion to the aggregate construct. A detailed discussion of this analysis is presented below in the Measurement Validation Section. We used the method of repeated manifest variables whereby the second-order construct of Process Agility was created by using all indicators from each of the two first-order constructs (Chin, 2000). Further assessment of construct validity was accomplished by modeling agility as a second-order formative construct and testing whether it is highly correlated with its indicator (Diamantopoulos and Winklhofer, 2001; Karimi *et al.*, 2007; Pavlou and El Sawy, 2006). For the Process Agility Indicator, two items were included in the questionnaire to measure the overall agility of the development process.

Commitment is defined as the developer embracing the methodology and fully supporting its use. Items for measuring commitment were taken from a previously validated scale (Fedor, D.B., Caldwell, S., and Herold, D. M., 2006). For measuring motivation, we obtained items from the previously validated Situational Motivational Scale (SIMS) that was developed and tested by Guay, *et al.*, 2000.

Table 2.6 below contains the items used for measuring process agility, the process agility indicator, commitment and motivation.

Table 2.6 Agility, Commitment and Motivation Items

Process Agility	Formative	Our requirements specification process dynamically evolves through continuous feedback from users.
		We constantly seek users' feedback to shape new requirements and re-prioritize features of the system.
		Developers communicate and collaborate with business people continuously to incorporate their evolving requirements.
		We frequently develop working software that is tested, integrated and executable as a partial system.
		Our initial system plan consists of minimal, yet essential requirements without complete and detailed specifications.
		Adjustments and refinements to requirements are always welcome at any stage of the development process.
		We believe changing requirements are normal and help to enhance the system quality.
		We don't mind deviating from established processes and procedures as long as we continuously deliver working software.
		The roles and relationships of our team members are flexible and not strictly defined.
		Working software is the primary measure of progress.
		We use short iterations of fixed intervals to quickly design, implement and test a small subset of the requirements.
Process Agility Indicator	Reflective	Overall, our development process is adaptive and responsive to changing user needs.
		In general, our development process is flexible with minimal planning.
Commitment	Reflective	I am doing whatever I can to help this methodology be successful.
		I am fully supportive of this methodology.
		I have tried (or intend to try) to convince others to support this methodology.

Table 2.6 - continued

		I intend to fully support my supervisor in the implementation and/or continued use of this methodology.
Motivation	Formative	Because I think that the methodology is interesting..
		Because I am using the methodology for my own good.
		Because I am supposed to use the methodology.
		There may be good reasons to use the methodology, but personally I don't see any.
		Because I think that using the methodology is pleasant.
		Because I think that using the methodology is good for me.
		Because using the methodology is something that I have to do.
		I use the methodology but I am not sure if it is worth it.
		Because using the methodology is fun.
		It is my personal decision to use the methodology.
		Because I am required to use the methodology
		I don't know; I don't see what using the methodology brings me.

2.5.3 Data Analysis

The sample of 479 software developers was described earlier. In addition to the process agility items, they also completed the items related to motivation and commitment to using and supporting use of the methodology.

The analysis was performed using a Partial Least Squares and Structural Equation Modeling tool PLS-GRAPH Version 3.0, Build 1130 (Courtesy of Dr. Chin, University of Houston). SEM allows for simultaneous examination of the structural component (path model) and measurement component (factor model) in one model.

The data was analyzed for outliers and missing data in the model. Any response with more than 10% of the data missing would be deemed unusable. Based on this criterion all responses were retained. As subscale scores were computed for the motivation construct, a mean substitution was used for any missing values contained within the items measuring motivation. We coded the missing values with a global value of -1 because PLS GRAPH can incorporate missing values. Next, the standardized z-scores were calculated to conduct a univariate outlier analysis for the IVs Process Agility, Evolutionary Development and Process Flexibility. For large sample sizes, Hair *et al.*, (1998) recommends a standardized score value of (+/-) 3 to 4 as a benchmark to identify outliers. Based on this criterion no outliers were detected in the data.

Descriptive statistics for the scale score for the multi-item constructs are shown below in Table 2.7. Most of the scale values show some skewness as well as kurtosis.

However, since normality is not an assumption with PLS this is not a cause for concern (Chin, *et al.*, 2003).

Table 2.7 Descriptive Statistics

	Minimum	Maximum	Mean	S.D.	Skewness Statistic	Kurtosis Statistic
Process Agility	1.73	7.00	4.74	.885	-.412	.400
Process Agility Indicator	1.00	7.00	4.72	1.14	-.612	.426
Evolutionary Development	1.14	7.00	4.87	.99127	-.403	.384
Process Flexibility	1.00	7.00	4.51	1.056	-.630	.748
Commitment	1.00	7.00	5.25	.973	-.471	.923
	Skewness Std error	.114				
	Kurtosis Std. error	.227				

2.5.4 Measurement Validation

First we tested for general reliability of the constructs using PLS internal consistency guidelines (Gefen and Straub, 2005). Further assessment of construct validity was accomplished by modeling agility as a second-order formative construct and testing whether it is highly correlated with its indicator. We also tested to see if the second-order aggregate construct fully mediates the effect of the first-order constructs. Finally, we tested the research model depicted in Figure 2.1 using the aggregate construct for Process Agility. These steps and their results are discussed below.

For the Commitment construct and the Process Agility indicator, we assessed the indicator reliabilities, as well as the convergent and discriminant validity. Indicator reliability and convergent/discriminant validity assessments are not meaningful for formative constructs; therefore the Process Agility, Process Flexibility, Evolutionary Development and Motivation constructs will not be included in these analyses (Bagozzi, 1994; Bollen, 1989). They will be assessed for validity using the procedure described by Diamantopoulos and Winklhofer (2001). This involves a check for collinearity and omitting those indicators that are highly collinear. Highly collinear indicators will inflate the variance explained of the latent variable (Bollen and Lennox, 1991). The highest VIF was less than 3 which is much less than the heuristic of 10. The condition index values were all acceptable.

The loadings of the indicators on their latent constructs were assessed for convergent validity. All indicators loaded with a value above .70, therefore they were all retained. Next, we ran a bootstrap procedure (200 samples) which generated the composite reliabilities, the average variance extracted (AVE) and the t-statistics for the path coefficients. The composite reliability of the construct, indicator loadings and their respective t-values are shown in Table 2.8. A good indicator of internal consistency is exhibited by a composite reliability value above 0.70 (Hair *et al.*, 1009). Because composite reliability considers the actual loadings in the calculations, it is considered better than Cronbach's alpha when measuring internal consistency (Ma and Agarwal, 2007). Our model demonstrates good internal consistency as all values for the constructs are larger than 0.70. According to Fornell *et al.*, 1981, AVE values above 0.5 indicate good convergent validity. Based on these criteria, our data demonstrates that the measurement is internally consistent and exhibits convergent validity.

Table 2.8 Indicator Loadings and Composite Reliabilities

	Composite Reliability	**AVE**	**Loading**	**Std. Error**	**T-Statistic**
Commitment	0.889	0.670			
COMMIT1			0.70	0.059	11.81
COMMIT2			0.94	0.022	42.14
COMMIT3			0.83	0.004	19.57
COMMIT4			0.77	0.004	17.53
Process Agility Indicator	0.830	0.710		0.014	
AGILEOVALL1			0.89	0.014	63.75
AGILEOVALL2			.79	0.034	22.84

Table 2.9 presents an AVE analysis. The square root of the AVE scores is shown by the bolded diagonal elements in the table. The correlations between the constructs are illustrated by the off-diagonal elements. Since the square root of the AVE is higher than any correlations involving the construct our data demonstrates discriminant validity. In other words, the constructs share greater variance with their own measures than with the other constructs in the model.

Table 2.9 AVE and Construct Correlations

	COMMIT	**OVAGILITY**
COMMIT	**0.82**	
AGILITYINDIC	0.347	**0.84**

Figure 2.3 illustrates the results of testing the second-order formative construct to see if it is highly correlated with its indicator. As shown, the correlation between the two indicator items that measured overall process agility with the aggregate second-order construct was 0.70 ($P<0.001$). This shows that the Process Agility aggregate construct is describing what it is intended to measure, exhibiting good construct

validity. Examination of the correlation between the first-order factors (.531) provides support for a formative model because a reflective model would tend to have much higher correlations (frequently above .80) (Pavlou and El Sawy, 2006). The path coefficients from the first-order constructs to the second-order construct (.553 and .590) indicate that each dimension contributes fairly equally to the aggregate construct.

In order to ensure that the second-order construct of process agility is a more parsimonious depiction of the first-order constructs and that its theorized predictive ability on the dependent variable is equal to that of the first-order constructs, we tested whether the process agility construct fully mediates the effect of the Evolutionary Development and Process Flexibility constructs. After controlling for the two first-order constructs, the Process Agility measure is the only significant predictor of developer motivation. This confirms that it fully mediates the impact of Evolutionary Development and Process Flexibility.

To further validate the agility scale, we took the highest average scores for agility and looked at respondents' descriptions of their methodology. We also took the lowest average scores for agility and looked at the respondents' descriptions. In both cases, the methodology descriptions provided, matched the level of agility as measured by the instrument. For example, the following methodology descriptions were reported by two respondents, the first having a very high average score for agility with the second having a very low average score for agility.

> "We are using a highly adapted/streamlined version of SCRUM, with a focus on communication and rapid response to changing conditions (gameplay feedback, design changes, etc.). Our development process also adopts a few ad-hoc concepts from XP like informal pair programming."

> "The relationship of each stage to the others can be roughly described as a Waterfall, where the outputs from a specific stage serve as the initial inputs for the following stage."

In combination, the above tests provide support for the second-order formative construct, Process Agility, in addition to verifying its construct validity.

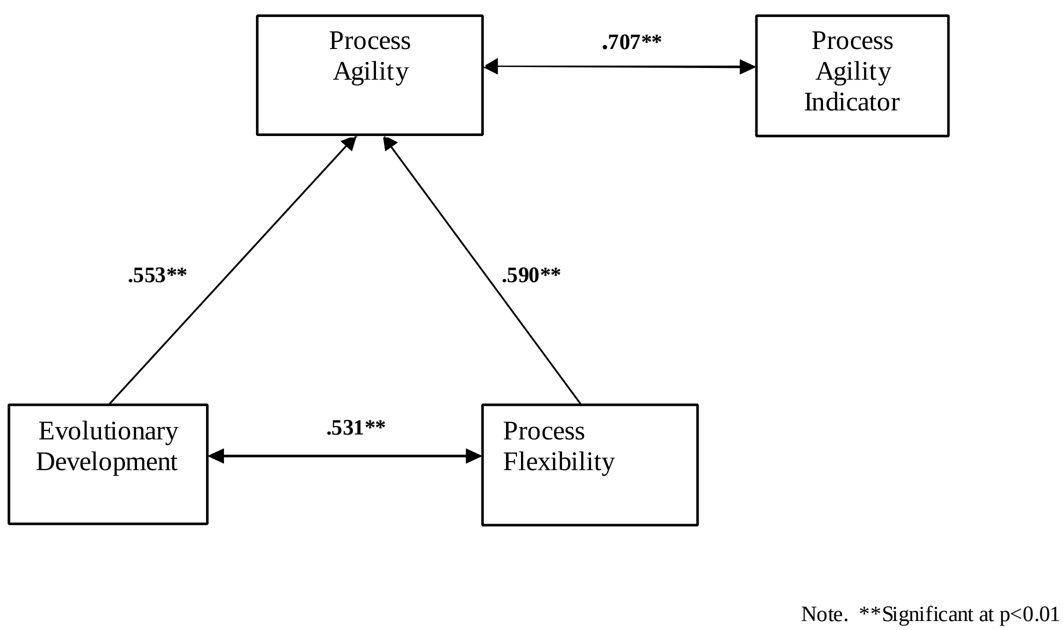
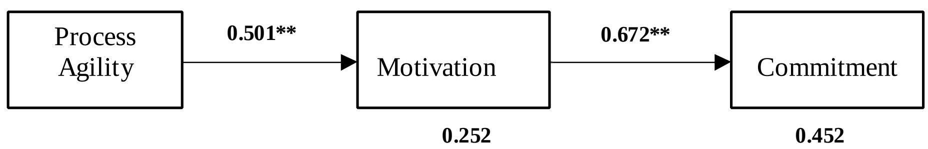


Figure 2.3 Second-Order Formative Construct of Process Agility

2.5.5 Hypothesis Tests

The model to assess the predictive ability of Process Agility is shown in Figure 2.4. Path coefficients with their corresponding t-values were taken from the Bootstrap output. Both paths are significant (p<.001). Support for the Hypotheses is summarized in Table 2.10.



Note. **Significant at p<0.01

Figure 2.4 Predictive Validation Results

Table 2.10 Summary of Hypothesis Testing

	Hypothesis	**Result**
H1	*Process Agility will be positively related to Motivation.*	Supported
H2	*Higher levels of motivation will be associated with higher levels of commitment.*	Supported

There was a strong positive relationship between Process Agility and developer motivation. And, the positive relationship between motivation and commitment shown in prior studies was supported. Thus, we have strong evidence to support the predictive validity of the Process Agility scale.

2.6 Discussion and Conclusions

In this research, we conceptualized, developed and validated an instrument to measure software development process agility. Process Agility is a multidimensional second-order formative construct comprised of two first-order constructs. These two first-order constructs or dimensions of process agility were labeled Evolutionary Development and Process Flexibility. The predictive capability of Process Agility was assessed to determine its ability to predict a developer's motivation and commitment to use a given development methodology. Process Agility had a strong positive relationship to motivation. We also ran the predictive model with the dimensions, Evolutionary Development and Process Flexibility, as two separate independent variables. The following diagram illustrates the magnitude of the relationships.

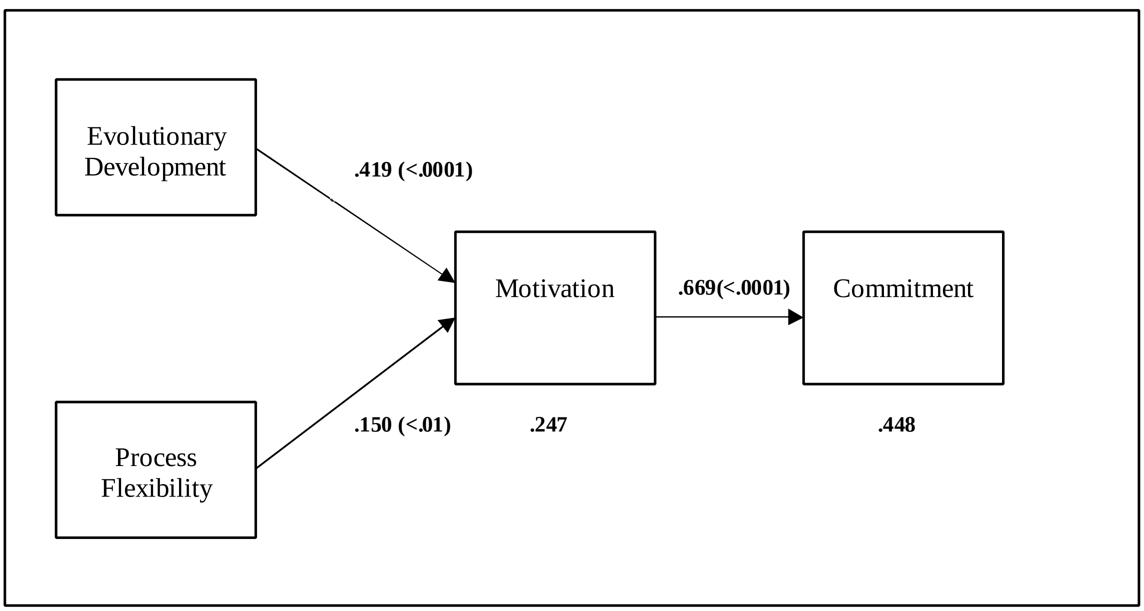


Figure 2.5 Predictive Model with Dimensions

Evolutionary development entails delivery of working software in short iterations in response to user feedback to address their evolving needs. It is characterized by adaptability and responsiveness to changing conditions. Process Flexibility captures the reaction capabilities inherent in agile processes, flexibility in team members' roles and their willingness to deviate from established processes and improvise in order to deliver working software. While both dimensions of Process Agility had a significant effect on motivation, Evolutionary Development had a stronger impact than Process Flexibility. One explanation for the relatively weak effect of Process Flexibility could be that Evolutionary Development is the core substance of the agile concept, and Process Flexibility is the supporting one. Evolutionary Development is necessary in order to fully realize the motivational impact of the reaction capabilities. In other words, reaction capabilities are fairly passive until they are utilized. And, in process agility, Evolutionary Development is the dimension that determines how and to what extent the capabilities are used. It would seem that full utilization of the capabilities and increased demand would result in enhancement of the capabilities. For example, if the Evolutionary Development process put additional demands on the developers in terms of learning new skills, this would increase the level of Process Flexibility through expanded skill variety, leading to increased motivation.

Another explanation for the difference in levels of effect is that a review of the five job dimensions that support the growth need strength associated with increased motivation illustrates that Evolutionary Development supports the majority of the five dimensions. In the Couger and Zawacki (1980) studies, investigating motivation of information technology (IT) professionals, it was found that IT professionals had the highest growth need strength and this growth need strength was based on five job dimensions. In reviewing the five job dimensions, (skill variety, task identity, task significance, autonomy and feedback), used to assess growth need strength and determine level of motivation, both Evolutionary Development and Process Flexibility entail practices that support growth needs. However, Evolutionary Development provides support for a three of the five dimensions. Higher task identity is supported by the iterative development environment which results in frequent delivery of working software allowing developers to complete identifiable pieces of work on a regular basis. Another characteristic of Evolutionary Development, continuous collaboration with users, allows developers the opportunity to see the impact of their work resulting in increased task significance. The iterative nature of Evolutionary Development combined with the ongoing communication between developers and users, creates a continual feedback loop thus, resulting in a high level of feedback. Process Flexibility supports the two job dimensions of autonomy and skill variety. The flexibility of team members' roles and the associated cross-training increases the skill variety of developers' jobs. Because developers are given the freedom to be flexible and deviate from established processes, they experience increased autonomy.

2.6.1 Limitations

One limitation in all studies using self-report surveys is the risk of common method bias. In order to mitigate this risk we have conducted the Harman's one-factor test and the results indicated that common method variance could not explain a significant part of the variance explained by the model. An exploratory factor analysis was conducted on all multiple items measures. Factors were extracted with eigenvalues greater than one. The unrotated solution generated 6 factors. One factor explained 25.7% of the variance and the remaining factors explained less than 11%. Total variance explained was 54.2%. After rotation, the percentage of variance explained by each factor was less than 11. Based on this analysis, no single factor was an overwhelming contributor to the variance. Therefore, we believe that common method variance does not pose a significant threat to measurement validity to this study.

Another possible limitation, related to generalizability, is that approximately 80% of our respondents were from one country with the remaining 20% being spread among various countries. However, no significant differences have been found in cross cultural studies related to the high growth needs attributed to information technology workers (Couger and Motiwalla, 1985; Couger, *et al.*, 1990). Furthermore, when culture was used as a control variable, it had no significant impact on the results.

2.6.2 Contribution to Research

In this study we conceptualized, developed and validated an instrument to measure the agility of a software development process. In doing so, we have made significant progress toward developing a standard measure of software development process agility that can be used to measure the agility of any software development methodology. In confirming the instrument's predictive ability, our results suggest a positive association between agile methods and developer motivation and commitment. To our knowledge this is the first time the impact of process agility on developers' motivation and commitment has been empirically investigated. Development and validation of an instrument to measure the agility of software development processes will facilitate research on agile methodologies. We have provided a valuable tool which can be used by researchers interested in investigating variables related to the agility of software development methodologies.

2.6.3 Contribution to Practice

Prior research has provided empirical support for an association between agile practices and increased software product quality (Nosek, 1998, Kessler, 1999, Williams, 2000, Upchurch, 2001). Implementing just one agile practice has been shown to improve quality, satisfaction and productivity without a significant change in cost (Parsons *et al*., 2007). However, without a valid agility measurement tool, it is virtually impossible to determine if the methodology in use is indeed "agile". This research has supplied an instrument that can be used by industry to assess the agility of an organization's systems development methodology. In addition to confirming predictive validity of the instrument, this study provided support for a positive relationship between agile processes and developer motivation and commitment.

2.6.4 Future Research

This study developed and validated an instrument to measure process agility within the context of software development. Results of its application found support for a positive association between process agility and motivation. It would be useful to investigate this phenomenon further in terms of examining the underlying reasons for this association. Within the domain of psychology, Self-Determination Theory (Deci and Ryan, 1995; 2000) is a well developed theory that has been tested in multiple domains. It examines intrinsic and extrinsic motivation and the impacts of social contexts or environmental factors on intrinsic motivation. Self-Determination Theory postulates that all humans have three innate psychological needs – autonomy, competence and relatedness – and that fulfillment of these needs is positively related to higher levels of self-determined motivation. The relationship between process agility and motivation could be examined through the lens of Self-Determination theory to investigate whether process agility plays a role in fulfilling certain needs

Having an instrument to measure the level of process agility of a given software development process opens up a host of possibilities in terms of investigating how the agility level impacts various success factors. For example, does increased agility result in higher software product quality? How does the level of agility affect user satisfaction? Do increased levels of agility result in higher levels of learning?

It has been argued that the migration to agile methodologies entails changes in technology, people, their roles and work habits, as well as in the communication and authority structures of the organization (Nerur, *et al.*, 2005). Because of this, a complete transition from traditional methodologies to agile methodologies is considered by many to represent a radical change, one that should perhaps be done in a piecemeal fashion (Copeland, 2001). The software process agility measurement instrument could be employed in studies to examine the degree of change required when moving from one level of agility to another.

CHAPTER 3

THE ACCEPTANCE OF AGILE DEVELOPMENT METHODOLOGIES BY SOFTWARE PROFESSIONALS: TESTING A THEORETICALLY INTEGRATED MODEL

3.1 Introduction

As organizations pour increasing amounts of resources into software development projects that, more often than not, come in over budget, overdue and deficient in terms of both quality and functionality (Benko and McFarland, 2003; McDonald, 2001), it becomes increasingly important to investigate ways to improve software project success. One approach is to improve the software development process. A variety of new and innovative methods and tools designed to improve the process have been introduced and research studies examining these new methods have shown promising results in terms of increased quality, satisfaction and productivity (Nosek, 1998; Kessler, 1999; Parsons *et al*., 2007; Williams, 2000; Upchurch, 2001). In the past, organizations and their developers have been reluctant to commit to their use (Glass, 1999; Hardgrave, 1995; Hartwick and Barki, 1994). However, the rapid ascent of agile methods (Ambler, 2007) over the past few years indicates that a lot of organizations are in the process of adopting (or have recently adopted) these methodologies. Yet, to the best of our knowledge, no study has examined the factors that influence the acceptance of such methodologies. In particular, does the level of agility provide sufficient motivation to developers to be committed to their methodology in use? Migrating from one methodology to another, sometimes, requires a significant behavioral change and if developers are not committed to using the methodology, they may not be willing to exert the effort required to make such a change. Having knowledge regarding how and why specific factors impact a developer's acceptance of the methodology is extremely valuable to organizations implementing new methodologies. If a developer does not commit and support usage of the methodology, it could result in a failed implementation or not fully realizing the benefits of the methodology. Awareness of these factors, arms organizations with the tools they need to be proactive and develop an effective implementation strategy.

This research identified factors that cause developers to embrace or resist new methodologies by turning to theories from both psychology and marketing and their usage in technology acceptance. Specifically, we examined the commitment to usage of system/software development methodologies, thus, contributing toward a deeper understanding of the underlying reasons for the effects of certain factors on technology implementation as well as making significant progress toward the development of a model for determining individual developers' intention to commit to and support the use of a given software development methodology.

Whereas the majority of prior technology acceptance research has focused primarily on acceptance of products, this research is concerned with acceptance of processes. Relatively little research has been focused on "reporting of new models for examining the intention to adopt system/software development methodologies"

(Toleman, *et al.*, 2004). According to (Reimenschneider, *et al.*, 2002), future research is needed to integrate the findings of the various models as well as identify and incorporate additional determinants of methodology acceptance. Another difference in this research is that it looked at commitment to usage. Most prior technology acceptance models have had self-reported usage as the dependent variable (Lee, *et al.*, 2003). When developers are committed to using a given methodology, they have "embraced" its usage and are dedicated to making it successful. We suggest that even if an organization mandates the usage of a given methodology, a developer's commitment (or lack thereof) to its success has a significant impact on the implementation outcome. We used the Self-Determination Theory as a basis for investigating whether there are certain innate psychological needs mediating the relationship between the factors and an individual's commitment to usage. Methodology adoption and use constitute a third-order change that involves people, technology, tasks, and structure (O'Hara, *et al.*, 1999). The extant literature falls short of providing a good explanation for the success/failure of software approaches. Our model integrates constructs from Innovation Diffusion with Needs Fulfillment from Self-Determination Theory to provide insights into what might motivate developers to be committed to a methodology. By examining these factors and their impact on commitment from the social-psychological perspective, we have opened up the "black box" and revealed the underlying mechanism at work in these relationships.

In summary, the objectives of this study are to:

1) Identify factors that motivate developers to commit to (or resist) the usage of a given software development methodology.

2) Determine how and why these factors impact commitment to usage.

Fulfilling these objectives would make the following contributions:

1) Extend usage models to include psychological mediators from Self-Determination Theory, thus providing an explanatory link between Diffusion Theory with its associated factors and individual commitment to usage.

2) Contribute toward a deeper understanding of the underlying reasons for the effects of certain factors on technology implementation.

3) Make significant progress toward development of a model for determining individual developers' intention to commit to and support the use of a given software development methodology.

4) Provide insight and helpful diagnostics to facilitate and assist practitioners in their efforts to implement new software development methodologies.

The next section of this article provides the theoretical background for our research. This is followed by a discussion of the research model and its corresponding hypotheses. Next, we define the control variables, provide a description of the research methodology and present the results of the study. The last section includes a discussion of the study with limitations and contributions as well as suggestions for future research.

3.2 Theoretical Background

This research is grounded in Rogers' Diffusion of Innovations Theory (Rogers, 1962) and Deci and Ryan's Self-Determination Theory (Deci and Ryan, 1985).

Figure 3.1 illustrates the hypothesized role of specific factors in the fulfillment of certain innate psychological needs as postulated by Self-Determination Theory. "General Innovation Process Characteristics" were taken from Diffusion of Innovations Theory. Specific development process characteristics refer to characteristics of the development methodology currently being used by the developer.

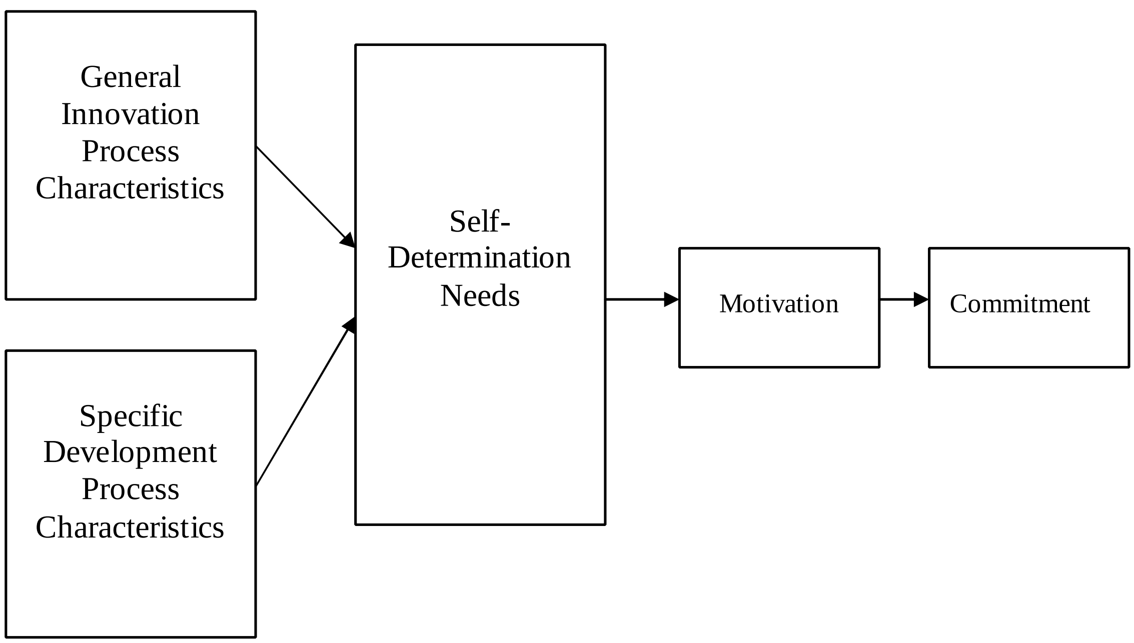


Figure 3.1 Research Framework

3.2.1 Diffusion Theory

In his often cited book, *Diffusion of Innovations,* Rogers defines an innovation as "an idea, practice, or object that is perceived as new by an individual or other unit of adoption." He goes further to say that it does not matter if the idea is "objectively" new as perceived by the lapse of time since its original development; if the idea, practice or object seems new to the individual, it is considered an innovation.

Rogers also describes the innovation-decision process as having 5 stages:

1) Knowledge – when the "decision –making unit" learns of the innovation's existence and understands how it functions;

2) Persuasion - when the decision-making unit forms an attitude toward the innovation;

3) Decision – when the decision-making unit works on activities that lead to an adoption or rejection choice;

4) Implementation – when the idea is put into use;

5) Confirmation – when the individual looks for reinforcement of the decision already made and the prior decision may be reversed at this point.

Therefore, the innovation-decision process does not end when the innovation is adopted. During the implementation stage it is very common for "re-invention" to occur. This refers to the fact that quite often an innovation is changed or modified during the process of its adoption and implementation. Because of this re-invention, it may be better to measure adoption at the implementation stage (Rogers, 2003).

In his model (see Figure 3.2), Rogers defined 5 categories of variables believed to determine rate of adoption. They are:

1) perceived attributes of innovations,

2) type of innovation-decision,

3) communication channels,

4) nature of the social system,

5) extent of change agent's promotion efforts.

Of these 5 types, the perceived attributes of innovations have "been most extensively investigated" and "they have been found to explain about half of the variance" in rates of adoption (Rogers, 2003). These 5 "perceived attributes of innovations" are:

1) relative advantage – the degree to which an innovation is perceived as better than the idea it supersedes,
2) compatibility – the degree to which an innovation is perceived as being consistent with the existing values, past experiences, and needs of potential adopters,
3) complexity – the degree to which an innovation is perceived as difficult to understand and use,
4) trialability – the degree to which an innovation may be experimented with on a limited basis,
5) observability – the degree to which the results of an innovation are visible to others.

Basically, Rogers postulates that innovations perceived by individuals as having less complexity but greater relative advantage, compatibility, trialability, and observability will be adopted more rapidly than other innovations.

The second type of variables, innovative-decision concerns the amount of autonomy the adopter has in terms of decision to adopt. Adoption is defined as optional, determined by group consensus or determined by a small number of powerful individuals. Communication channels refer to the type of communication conduit used to spread information about the innovation (i.e., mass media, interpersonal).

Nature of the social system is the fourth type of variable. This deals with its norms, degree of network interconnectedness, opinion leadership (degree to which an individual can influence others in adopting) and consequences of adopting.

The final type of variable is the extent of change agents' promotion efforts.

Rogers provides an extremely comprehensive set of variables that appears to cover all cases. However, of these 5 categories or types of variables, some would seem to be more applicable to software methodology diffusion than others. For example, as stated above, a major portion of diffusion research is focused on the perceived innovation characteristics. Perceived innovation characteristics can apply to any innovation, although the pertinence of specific characteristics may vary. Type of innovation-decision is also relevant to the adoption of development methodologies. Moore and Benbasat's (1991) instrument to measure perceptions of adopting an information technology innovation includes a variable they define as "voluntariness". Voluntariness refers to the degree of autonomy an individual has in terms of using the innovation. This is very similar in meaning to Rogers' innovation-decision and has been shown to be significant when measuring developer acceptance of software methodologies (Riemenschneider, *et al.*, 2002). However, the third category, communication channels does not seem as applicable. Adopting a new development methodology can change the very nature of an individual's work process. This represents a more radical change than adopting a new product. Because of this, the typical effort required to assimilate the new process is going to be greater than the typical effort required to learn to use a new product. Therefore, the communication channel will be closer to interpersonal than mass media. A new software process is usually communicated to potential adopters via training classes and/or on the job training. Nature of the social system and extent of change agent's promotion efforts, Rogers' fourth and fifth types of variables, seem similar in that they both involve an individual's or group of individuals' ability to influence potential adopters. Since software development involves teams or groups working together and methodologies define how the team members interact and collaborate, nature of the social system and members' influence on each other would seem to be very applicable to software process innovation.

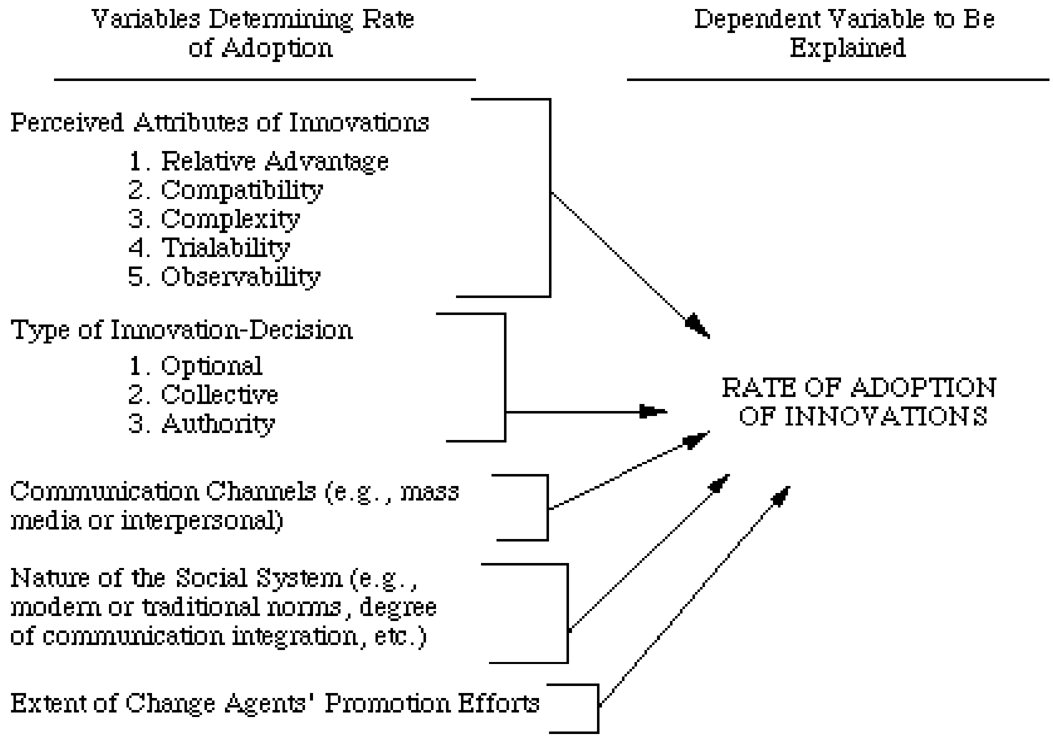


Figure 3.2 Variables Influencing an Innovation's Rate of Adoption
(Rogers 2003, 1995, 1983, 1971, 1962)

In reviewing Rogers work on diffusion of innovation, it is evident that his coverage of innovations included processes as well as products. However, most information technology research efforts appear to be aimed at developing measurement instruments and models for new product innovations rather than process innovations (i.e., Perceived Characteristics of Innovating, (Moore and Benbasat, 1991); Technology Acceptance Model, (TAM1, TAM2, Davis, 1989)). In their work on developer acceptance of methodologies, (Riemenschneider, *et al.*, 2002) compared five theoretical models in order to investigate why "individual developers accept or resist" adopting new methodologies. They used elements from the technology acceptance model (TAM1,TAM2), perceived characteristics of innovating (PCI), the theory of planned behavior (TPB) and the model of personal computer utilization (MPCU). While various studies have been conducted applying one or two of these models, the (Riemenschneider, *et al.*, 2002) study is the first to simultaneously test all five models. It is also one of the few studies designed to specifically measure individual intention to adopt a software development methodology. There are 12 distinctively defined constructs covered by the models. This includes allowance for those that overlap. For example, the construct labeled "usefulness" in TAM and TAM2 is labeled "relative advantage" in PCI, "attitude" in TPB and "job fit" in MPCU. However, all of these items basically measure the same construct. After testing the constructs from all five models, only four variables proved to be significant in predicting individual intention to adopt. They were usefulness, subjective norm, voluntariness, and compatibility. No single model possessed all four of the significant variables. By combining these five models, the authors were able to provide insights into methodology adoption. This study found that variables significant in product innovation are not necessarily significant in process innovation and vice versa. For example, ease of use and perceived behavioral control (internal and external) previously found to be significant in tool studies were found to be nonsignificant in the methodology study (Riemenschneider *et al.*, 2002). These findings suggest that as "the behavioral domain changes from tool use to methodology use, there is a reduction in the relevance of how easy or hard the behavior is to perform and whether one possesses adequate internal or external resources to perform it." Furthermore, there is "an increase in the relevance of subjective normative pressure to perform the behavior, the perception of a formal mandate and the compatibility of the target behavior with individuals' current ways of performing their work" (Reimenschneider *et al.,* 2002).

In a study examining process transformation from a mainframe COBOL-based to a Client Server C-based environment, Agarwal and Prasad (2000), looked at beliefs about relative advantage, ease of use, compatibility and voluntariness regarding usage intentions. They found relative advantage, ease of use and compatibility significantly related to attitude/intention to use.

Hardgrave and Johnson (2003), developed and tested an information systems development acceptance model within an object oriented systems development context. They found subjective norm, organizational usefulness and internal perceived behavioral control to be significantly associated with methodology intention to use the new ISD process. A gap does exist in that there appears to be no integrative comprehensive model specifically designed to measure the individual intention to adopt new software methodologies. See Table 3.1 for a summary of these models.

Of the five attributes defined in Rogers' diffusion theory, only three have been included: complexity, compatibility and relative advantage. In Tornatzky and Klein's meta-analysis consisting of 75 innovation characteristics studies, these three characteristics (i.e., complexity, compatibility and relative advantage) were found to consistently influence innovation adoption and/or implementation. Although complexity was found to be insignificant in Riemenschneider's methodology study (2002), we have chosen to include it here because of its persistence in numerous prior studies. However, it does seem conceivable that the shift from tool use to methodology use reduces the relevance of how easy or difficult a behavior is, and increases the importance of social influence factors. And, in performing a particular behavior as prescribed by a given methodology, ease of use may be exemplified by compatibility with current work behaviors, attitudes and values. Therefore, it is possible that within the domain of methodology usage, the compatibility construct tends to also encompass aspects related to ease of use. However, there have not yet been enough replications of this phenomenon to warrant its exclusion from the model.

As mentioned earlier, adoption of a new methodology potentially changes the very nature of an individual's work behavior and interaction with other individuals, both inside and outside of the team. It can also change the entire organizational structure of the development group. Therefore, social factors become more significant in software process innovation. Social Influence has been included in the research model. It is characterized by nature of the social system from Rogers' Diffusion model.

Table 3.1 Summary of Methodology Usage Models

Study	Context	Significant Predictors
Agarwal and Prasad, 2000	Process transformation from a mainframe COBOL-based to a Client Server C-based environment	Relative Advantage, Ease of Use, Compatibility
Riemenschneider, *et al.*, 2002	Adoption of a new systems development methodology	Usefulness, Subjective norm, Voluntariness, Compatibility
Hardgrave and Johnson, 2003	Object oriented systems development	Subjective norm, organizational usefulness, internal perceived behavioral control

3.2.2 Self-Determination Theory

Every stage of a methodology requires a large amount of human intelligence and intervention and might at times cross boundaries of established norms. A methodology may be considered a radical departure from the current way of doing things and this introduces challenges related to human factors leading to acceptance or rejection. Therefore, it seems that in order to commit to a given methodology, an individual would need to have a fairly high level of internal motivation to do so. Self-Determination Theory (SDT) proposes that every human being has three innate psychological needs that, when satisfied, produce enhanced self-motivation. These three needs are: competence, autonomy and relatedness (Deci and Ryan, 2000). Competence refers to an individual's feeling of being proficient at the activity in which they are currently engaged. It is associated with a high level of self-efficacy. Autonomy involves choice and acting from an individual's integrated sense of self. Relatedness is an individual's need to belong and feel connected to others (Deci & Ryan, 2000).

SDT is focused on the role of needs fulfillment in motivating individuals. When the three needs (autonomy, competence, relatedness) are satisfied, the result is enhanced self-motivation and mental health. Conversely, when these needs are thwarted, the result is decreased motivation and well-being. SDT has been validated within the context of health care, education, work, sport, religion and psychotherapy (Deci and Ryan, 2000). See Table 3.2 for a sampling of studies based on SDT. The theory assumes a continuum from intrinsic to extrinsic to amotivation. The following four levels of extrinsic motivation are based on an individual's degree of internalization of external factors.

- External – compliance is based on external rewards and punishment
- Introjected – self-control; internal rewards and punishment
- Identified – conscious valuing and accepting as personally important
- Integrated – identified regulations are "fully assimilated to the self" (Deci and Ryan, 2002)

As the external factors become more internalized, the extrinsic motivation more closely resembles intrinsic motivation. While intrinsically motivated behaviors originate within the self, actions that are extrinsically motivated "can become self-determined as individuals identify with and fully assimilate their regulation" (Deci and Ryan, 2000). It has also been shown through accumulated research that the "commitment and authenticity characteristic of intrinsic motivation and integrated extrinsic motivation" occur most when individuals get support for competence, autonomy and relatedness.

Vallerand (1997) has proposed that different levels of generality of an integrated motivational sequence exist where environmental factors influence perceptions of competence, autonomy and relatedness. These perceptions determine to what extent an individual exhibits self-determined motivation which then leads to outcomes. Vallerand's motivational sequence is shown below:

Environmental Factors → Psychological Mediators → Self-determined Motivation → Consequences

It has been shown that one of the consequences of self-determined motivation is a higher level of commitment on the job (Deci and Ryan, 2000; Gagne and Deci, 2005). This is of particular interest since our focus is a developer's commitment to using a given methodology. Malhotra and Galletta (2005) propose a multidimensional model of commitment within the context of volitional systems adoption. They define identification, internalization and compliance as different dimensions of commitment. Compliance refers to public conformity without really privately accepting the behavior, whereas identification and internalization denote public conformity combined with increasing levels of private acceptance. They found that identification and internalization had a positive effect on both initial use and continued use, whereas compliance commitment had a negative effect on both initial use and continued use. Within our study, commitment encompasses the identification and internalization dimensions of commitment.

Table 3.2 Sampling of Studies Using Self-Determination Theory

Study/Context	Independent Variables Examined	Needs Satisfaction and Dependent Variables
Baard and Deci, 2004/ Work	Perceived autonomy support of work climate; Autonomous causality orientation	Increased need satisfaction lead to higher job performance; better psychological adjustment
Chirkov and Ryan, 2001/ Academic cross cultural (Russia vs. United States)	Perceived Teacher autonomy support; Perceived Parental autonomy support	Increased autonomy support lead to increased well being for both Russian and American students

Table 3.2 - continued

Deci, *et al.*, 2001/ Work cross cultural (Bulgaria vs. United States)	Managerial autonomy support; Supervisory autonomy support; Environment support	Increased support lead to increased engagement, decreased anxiety and increased self esteem in both Bulgarian and American employees
Edmunds, *et al.*, 2007/ Exercise Class	Supportive Teaching Style: -Autonomy supportive, -Well structured, -Interpersonally involved	Supportive teaching style positively influenced behavioral, cognitive and affective responses to exercise
Grolnick and Ryan, 1989/ Academic	Parental Autonomy support	Increased parental autonomy support positively influence self-regulation, competence and adjustment variables
Niemiec, *et al.*, 2006/ College	Perceived need support of Parents	Increased need support lead to increased psychological well-being and decreased psychological ill-being
Pelletier, *et al.*, 2001/ Sports team	Coach's autonomy support; Coach's control	Increased autonomy support and decreased controlling behaviors lead to increased self-determined motivation

Table 3.2 - continued

Reis, *et al.*, 2000/ Lab Experiment	Autonomy, competence and relatedness support	Need fulfillment was positively associated with well-being
Standage, *et al.*, 2003/ Physical Education	Mastery Climate (support hard work, task mastery); Origin Climate (autonomy supportive); Performance Climate (focus on competition)	Mastery Climate increased autonomy leading to increased physical intention; Origin Climate increased all needs leading to increased physical intention; Performance Climate was weakly and negatively related to need satisfaction
Williams, *et al.*, 2006/ Health Care	Perceived Autonomy support	Increased perceived autonomy support lead to increased autonomy and competence resulting in compliance with treatment program (taking of medication and cessation of smoking)

Using Diffusion Theory and SDT as our base, the next section describes the research model and corresponding hypotheses.

3.3 Research Model and Hypotheses

In the previous section, our review of SDT included Vallerand's motivational sequence originating with environmental factors which lead to consequences mediated by need fulfillment and motivation. Our theoretical discussion also included innovation factors which, in this case, refer to characteristics of a given systems development methodology. In the research model, these characteristics act as the factors experienced by the user when engaging in processes prescribed by the methodology. The study will investigate the user's perception of these factors in terms of their effect on the motivational sequence.

First, we will provide the research model followed by a definition of the constructs and a description of the hypotheses.

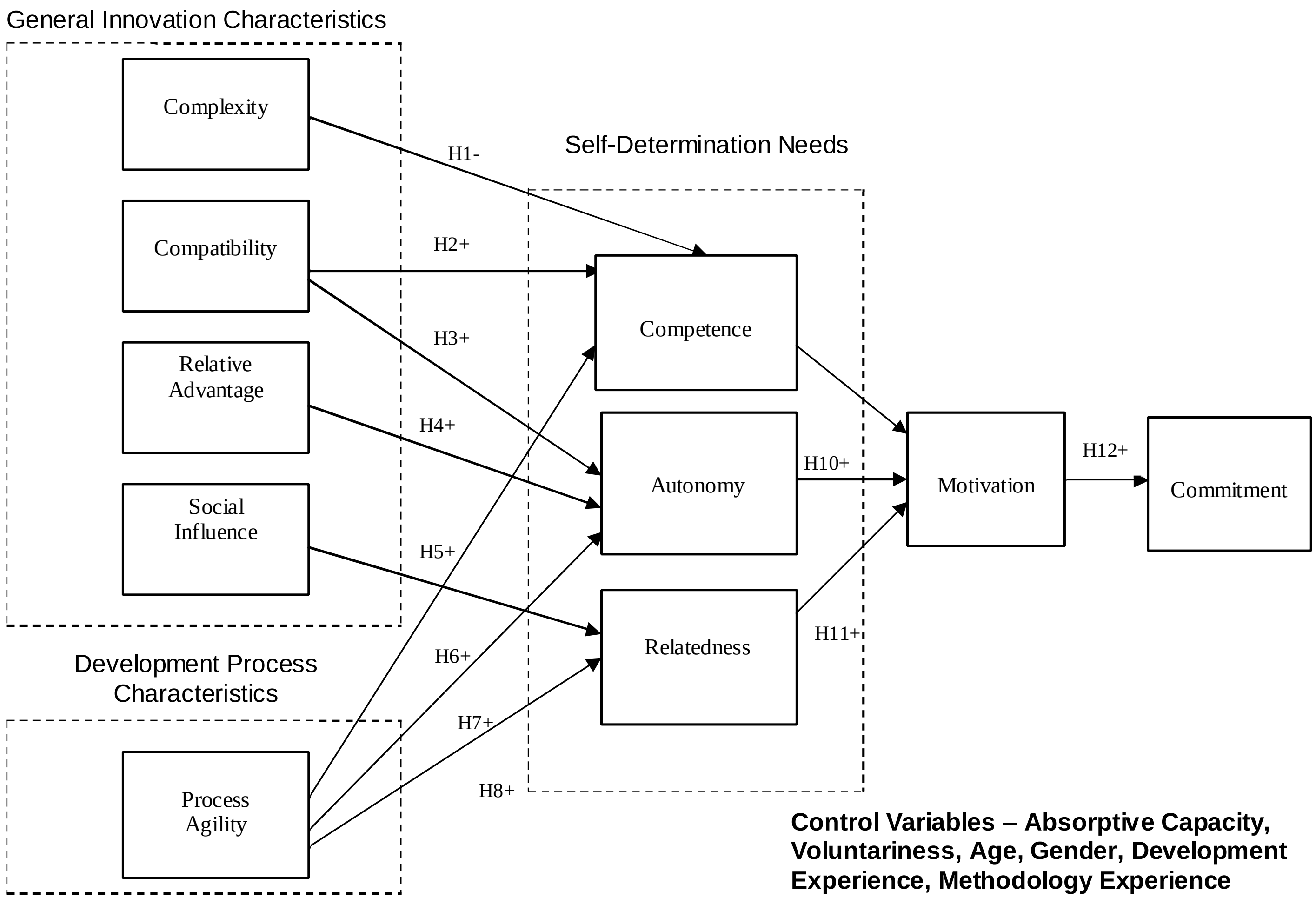


Figure 3.3 Research Model

3.3.1 The Independent Variables: Complexity, Compatibility, Relative Advantage, Social Influence and Process Agility

Complexity is a reflection of the extent to which the software developer perceives the methodology to be difficult to understand and use (Riemenschneider, *et al.*, 2002). Compatibility is how much the individual feels the methodology "fits in" with their way of "doing things". Consistent with Karahanna *et al.*'s (2006) conceptualization of compatibility as a multidimensional construct, we include three dimensions: compatibility with prior experience, compatibility with preferred work style and compatibility with values. Relative advantage refers to the degree to which individuals think the methodology aids or facilitates them in improving their job performance.

Social Influence is characterized by nature of the social system from Rogers' Diffusion model. This construct is labeled as subjective norm in most technology acceptance models (e.g., TAM2 and TPB). It refers to "the degree to which people think that others who are important to them think they should perform the behavior" (Riemenschneider, 2002).

In addition to general innovation factors and social factors, specific characteristics of a given process could also have an impact on an individual's commitment to its use. For this study, we have included process agility. Within the domain of software development, a methodology's agility has become a primary focus because of the rapidly changing development environment. Agile methodologies are designed to aid the developer in responding quickly and effectively to these changes. Process agility has been conceptualized as comprising two dimensions: Evolutionary

Development and Process Flexibility. Evolutionary Development captures the dynamic aspect of an agile process that strives to deliver working software in short iterations by anticipating and embracing change, actively involving stakeholders and continually seeking users' feedback to address their evolving needs. It embodies the collaborative dimension of agile processes coupled with the adaptability and responsiveness to changing conditions. Process Flexibility encompasses the reaction capabilities inherent in agile processes. It is characterized by flexibility of team members' roles and the willingness to deviate from established processes in order to deliver working software. It is the ability and willingness to improvise and come up with new solutions on the fly.

3.3.2 The Mediators: Competence, Autonomy and Relatedness

As discussed in the previous section, competence, autonomy and relatedness are the three needs postulated by SDT to lead to increased motivation. Competence refers to the feeling of being proficient at the activity in which the individual is currently engaged. It is associated with a high level of self-efficacy. According to Deci and Ryan, "autonomy involves being volitional or acting from one's integrated sense of self and endorsing one's action". It does not necessarily mean being separate from or independent of others. Relatedness is an individual's need to feel "belongingness and connectedness" to others (Deci and Ryan, 2000).

3.3.3 The Outcome Variables: Motivation and Commitment

In the previous section, we discussed the continuum from intrinsic to extrinsic to amotivation proposed by SDT as well as a definition for each of the four levels of extrinsic motivation defined by SDT. It was also noted that fulfillment of the three needs described above lead to higher levels of self-determined motivation which in turn lead to a higher level of commitment on the job. In terms of the methodology, commitment refers to the developer "embracing" usage of the methodology and feeling an attachment or sense of obligation to support and promote its successful implementation.

3.3.4 Hypotheses

Complexity was defined as the developer's perception of how hard or easy a methodology is to follow or learn. The less complex a system is, "the greater should be the user's sense of efficacy and personal control regarding his or her ability to carry out the sequences of behavior needed" (Davis *et al.*, 1989). By contrast, a more complex system should result in a lower sense of efficacy regarding its use. Self-efficacy beliefs refer to people's judgments of their capabilities to execute courses of action that may be required to achieve a specific type or level of performance (Bandura, 1978). Self-efficacy beliefs have been shown to affect behaviors of people in that if they have a low feeling of self-efficacy regarding the ability to achieve the desired outcome, they will not expend as much effort to perform the action and in fact, may not even try at all (Bandura, 1986). A feeling of competence is achieved when individuals are able to feel that they have succeeded at tasks that are optimally challenging and were able to achieve the desired outcomes (Baard, *et al.*, 2004). However, if individuals have low self-efficacy regarding their ability to follow the methodology, they may not expend much effort in trying to follow the methodology. This decreased effort may result in failure to achieve desired outcomes leading to lower feelings of competence. Conversely, higher efficacy regarding use of the methodology could lead to increased effort which may result in success in achieving desired outcomes. In this scenario, the individual would experience a higher level of competence. The following hypothesis states this negative relationship between complexity and competence.

Hypothesis 1 (H1). *Perceived complexity of the development methodology will be negatively related to the level of competence experienced while using the methodology.*

Compatibility was defined as the degree to which developers feel that the methodology fits in with their way of doing things. As mentioned earlier, compatibility has been reconceptualized as having multiple dimensions and it was found that compatibility with prior experience had an impact on ease of use (Karahanna *et al.*, 2006). That is, if individuals felt that the system was compatible with their prior work experience, they felt that it was easier to use. So if developers felt that the methodology was compatible with their past work experience, they would find it easier to use and have greater amounts of efficacy with regard to its use. The increased feeling of efficacy, as discussed above, could result in increased effort which may contribute to a high level of success in achieving desired outcomes. Achievement of desired outcomes leads to increased feelings of competence. This positive relationship is the basis of our next hypothesis.

Hypothesis 2 (H2). *Perceived compatibility of the development methodology will be positively related to the level of competence experienced while using the methodology.*

One dimension of compatibility defined by Karahanna *et al.*, in the 2006 study is compatibility with values. This refers to compatibility with the individual's dominant value system. On the continuum from extrinsic to intrinsic motivation, the 3rd level, identified regulation, involves a conscious "valuing" and accepting something as personally important. Deci and Ryan (2000), observe that, "As people internalize regulations and assimilate them to the self, they experience greater autonomy in action." If an individual's values happen to be compatible with the principles of the methodology, the internalization process will be facilitated. They will feel as if they have a choice and are making a conscious decision to use the methodology. This feeling of choice and endorsement of one's actions leads to increased feelings of autonomy. The following hypothesis relates to this relationship between compatibility and autonomy.

Hypothesis 3 (H3). *Perceived compatibility of the development methodology will be positively related to the level of autonomy experienced while using the methodology.*

Much like the relationship between compatibility and autonomy described above, perceived relative advantage will facilitate internalization, thus causing the individual to experience greater autonomy. If the individual believes that the methodology helps them to improve their job performance, they will value its use and accept it as "personally important" (Ryan and Dec, 2000). Once this internalization has occurred, the individual experiences feelings of autonomy because they look at their use of the methodology as a choice they have made. Because they now feel that it is personally important to them, they are making a conscious choice regarding its use. Therefore, we hypothesize this positive relationship between relative advantage and competence.

Hypothesis 4 (H4). *Perceived relative advantage of the development methodology will be positively related to the level of autonomy experienced while using the methodology.*

Social influence was defined previously as how much people believe others, “who are important to them think they should perform the behavior” (Riemenschneider, 2002). In order for this to occur, however, relatedness must be present. If an individual does not feel a belongingness or connectedness to the entity exerting the influence, it will not have the desired impact. In a study of hockey players it was discovered that the coach’s influence was only effective when the players felt a positive sense of relatedness with the coach (Vallerand, 2000). When the coaches put an emphasis on sportsmanlike conduct (i.e., reducing violence), the players self-reported violence was only decreased for players who felt a relatedness to their coaches.

Hypothesis 5 (H5). *Social Influence will be positively related to the level of relatedness experienced while using the methodology.*

When agile development methods are used, developers are cross-trained so that they are able to work on multiple facets of the project. This increases their efficacy regarding the ability to achieve desired outcomes related to the goal of producing working software. And, agile development is characterized by evolutionary development which ensures that evolving requirements and changes are continually accommodated. This is accomplished through an iterative, incremental development process that involves active engagement of customers (Beck, 2000; Cockburn, 2001; Highsmith, 2004; Martin, 2002). Feedback and communication are key elements of an evolutionary development process (Ambler, 2000; Boehm & Turner, 2004; Highsmith, 2002, 2004), thus guiding the efforts of the developers. Developers quickly get a sense of what works and what doesn't and are continually learning and adjusting their behaviors. This process, therefore, enhances their skills and heightens their sense of competence. We suggest that this leads to developers' feeling an increased sense of competence. This positive relationship is stated in the next hypothesis.

Hypothesis 6 (H6). *Process Agility of the development methodology will be positively related to the level of competence experienced while using the methodology.*

Agile methods also prescribe that developers be empowered to make decisions without having to go through a bureaucratic approval process (Highsmith, 2002). There is not a rigid hierarchy or chain of command as agile leaders act more as coaches facilitating the development process (Chin, 2004; Cockburn and Highsmith, 2001). Feelings of autonomy are increased in an agile environment because developers are given the freedom to be flexible and deviate from established processes in order to deliver working software (Highsmith, 2002; 2004). Flexibility of team members' roles and relationships also gives developers an increased sense of autonomy in making decisions regarding how they accomplish their job. Therefore, we propose that process agility results in an increased level of developer autonomy.

Hypothesis 7 (H7). *Process Agility of the development methodology will be positively related to the level of autonomy experienced while using the methodology.*

Relatedness refers to a feeling of belongingness and connectedness to others. Agile development is characterized by continuous collaboration between developers and users. It emphasizes individuals and interactions (Highsmith, 2002) over processes and procedures. Because team members' roles and relationships are flexible and teams are self-organizing it is critical that developers work closely together. We propose that this emphasis on collaboration and interaction results in higher feelings of relatedness.

Hypothesis 8 (H8). *Process Agility of the development methodology will be positively related to the level of relatedness experienced while using the methodology.*

The following three hypotheses regard the perception of need fulfillment and its relationship to motivation. Self-Determination theory has been well tested in multiple contexts providing empirical support for the relationship between the perception of need fulfillment and motivation. Several of these studies are mentioned below.

In a study of 328 secondary school students, it was found that perceived competence in a physical education environment was positively related to increased motivation to participate (Standage *et al.*, 2003). A study of 5th and 6th grade students examined levels of competence and motivation in a competitive sports environment. It was found that performance based awards lead to increased feelings of competence and motivation for those who won the awards and decreased feelings of competence and motivation for those not winning awards (Vallerand, *et al.*, 1986). Perceived competence was shown to be related to motivation to participate in an exercise program in a study involving 723 school children (Biddle, *et al*., 1999).

Therefore we propose the following hypothesis.

Hypothesis 9 (H9). *Perceived competence while using the methodology will be positively related to motivation.*

Perceived autonomy support from physical education coaches was found to be positively associated with increased motivation to participate in physical education activities (Standage *et al*., 2003). In another study involving competitive swimmers, it was found that increased levels of coach's autonomy support were associated with increased motivation (Pelletier, *et al*., 2001). Two studies examining the relationship between adolescents' perceptions of support for autonomy from their parents and autonomous self-regulation for pursuing college found support for a positive relationship between autonomy and motivation (Niemiec *et al*., 2006).

The next hypothesis addresses this association between autonomy and motivation.

Hypothesis 10 (H10). *Perceived autonomy while using the methodology will be positively related to motivation.*

In a secondary school study of 328 students, higher levels of relatedness were associated with increased motivation to participate in physical education activities (Standage, *et al*., 2003). Two studies examining the relations of adolescents' perceptions of support for relatedness from their parents and autonomous self-regulation for pursuing college found support for a positive relationship between relatedness and motivation (Niemiec *et al*., 2006). In a study examining the effect of young gymnasts perception of support for the need for relatedness, it was found that perceived support for relatedness lead to increased motivation (Gagne, *et al*., 2003).

Therefore, we expect to find a positive relationship between relatedness and motivation.

Hypothesis 11 (H11). *Perceived relatedness while using the methodology will be positively related to motivation.*

As noted earlier, studies have shown a positive relationship between motivation and commitment within a work setting (Deci and Ryan, 2000; Gagne and Deci, 2005; Mowday *et al*., 1979). In addition, studies have found that autonomous motivation is identified with increased affective commitment (Gagne and Deci, 2005). Affective commitment is associated with an individual identifying with and feeling an emotional attachment to the organization. One consequence of this type of commitment is an individual accepting the organization's goals as one's own. Since we are concerned with a developer's commitment to using a given methodology, this is of particular interest. If developer usage of the methodology is an organizational goal, a higher level of developer motivation would result in a higher level of commitment to using the methodology. Therefore, we propose the following hypothesis related to motivation and commitment.

Hypothesis 12 (H12). *Higher levels of motivation will be associated with higher levels of commitment to usage of the methodology.*

3.4 Control Variables

Based on prior studies and the focus of our research, we chose to use the following six control variables: voluntariness, age, gender, development experience, experience with methodology and absorptive capacity. Because of the nature of our sample, we added culture as another control variable.

Voluntariness is whether use of the innovation is mandatory. It is very similar in meaning to Rogers' innovation-decision and has been shown to be significant when measuring developer acceptance of software methodologies (Riemenschneider, *et al.*, 2002). It has also been shown that subjective norm had a significant effect on behavioral intention in mandatory settings but not in voluntary settings (Hartwick and Barki, 1994). Voluntariness, age gender and experience have all been used as moderators in prior acceptance studies (Venkatesh, *et al.*, 2003). We controlled for these variables.

Absorptive capacity refers to the organization's ability to gain new knowledge or insights, identifying and acquiring internal as well as external knowledge and assimilating it into the processes and products of the organization. We suggest that it is necessary to control for an organization's level of absorptive capacity to ensure that the individual's level of commitment is due to the methodology's characteristics and not attributable to the organization's ability to assimilate new knowledge into its product development processes.

3.5 Research Methodology

3.5.1 Data Collection

The data was collected using an online survey accessed via an internet link. Subjects were software developers of all levels currently working in a software development capacity and using a software development methodology. There was no specific criteria related to the type of methodology and, in fact, in some cases they were using a combination of methods.

Prior to conducting the full scale study, the online survey was piloted with a total of 20 respondents. Most of the respondents were developers from industry and closely resembled the population of interest. Several of the respondents were academics with prior industry experience. The questionnaire had an area at the end asking for comments regarding the survey. Based on these comments, minor changes were made related to the wording of 2 of the questions and several of the instructions, the number of questions per screen and the highlighting of key words in the instructions. A preliminary assessment indicated adequate discriminant and convergent validity among the items. Therefore, no items were dropped from the survey.

We contacted CIO's and other top executives from several organizations via email and telephone. They were asked if they would consider sponsoring the survey by asking their developers to participate. They were informed about the general objective of the survey and assured that all responses would be kept confidential. If they agreed, they were provided with a link to the online survey. When respondents clicked on the link, it opened an introductory page to the survey. The introductory page displayed the

UTA logo and a paragraph assuring their privacy and stating that participation was voluntary.

Because of the nature of online surveys, it is difficult to determine exact response rates for each organization. However, it is estimated that the overall number of subjects in the potential pool was approximately 8,000 and 554 surveys were completed. Therefore, we estimate a response rate of 6.92%. Large amounts of data were missing from 75 of the surveys, making it necessary to exclude them from the analysis. In fact, so much data was missing that we are unable to report a sufficient amount of demographic data regarding those responses. After dropping the 75 responses, our final sample size is 479.

Demographic information is presented in Table 3.3 below. The sample does seem to represent a wide selection of different industries, and a variety of age groups. Although a majority of the respondents are male, this is the case for the overall population of programmers/analysts (Craig, *et al.*, 2002; Joshi and Schmidt, 2005). As expected, most hold bachelor's or master's degrees. Therefore, we have good reason to believe that the sample comes from many different segments of the population, and is not a biased sample from a particular segment of the population. As illustrated, a variety of methodologies is represented. In order to allow for respondents that may be using a combination of methodologies, subjects were allowed to select more than one type.

Table 3.3 Demographic Profile of Respondents

MEASURE	ITEM	FREQ	PERCENT
Methodology	Adaptive Software Development	12	2.5
	Cleanroom	3	0.6
	CMM Integration	197	41.1
	Crystal	3	0.6
	Dynamic Systems Development	6	1.3
	Feature Driven Development	7	1.5
	Joint Application Development	19	4.0
	Lean Development	16	3.3
	Personal Software Process	7	1.5
	Rapid Application Development	43	9.0
	Rational Unified Process	114	23.8
	Scrum	10	2.1
	Spiral	28	5.8
	CMM for Software	50	10.4
	Team Software Process	11	2.3
	Waterfall	193	40.3
	eXtreme Programming	20	4.2
	Other	60	12.5
Country	North America	33	5.9
	India	382	79.7
	Canada	11	2.3
	England	15	3.1
	Bulgaria	16	3.3
	China	17	3.5
	Costa Rica	3	0.6
	France	1	0.2
	Germany	1	0.2
Gender	Female	73	15.2
	Male	392	81.8
Age	18 – 21 years	1	0.2
	21 – 30 years	261	54.5
	31 - 40 years	184	38.4
	41 – 50 years	15	3.1
	51 – 60 years	7	1.5
Education	Associate Degree	4	0.8
	Bachelor Degree	281	58.7
	Master Degree	176	36.7
	Doctorate Degree	1	0.2
	High School Degree	3	0.6

Table 3.3 - continued

	Other	4	0.8
Industry	Banking/Insurance/Financial Service	12	2.5
	Construction/Architecture/Engineering	2	0.4
	Consulting/Business Service	65	13.6
	Government, including Military	2	0.4
	Health Care	4	0.8
	Hotel/Entertainment/Service Industry	4	0.8
	IT/Telecommunications	361	75.4
	Manufacturing	15	3.1
Job Title	Business Analyst	14	2.9
	Programmer	24	5.0
	Programmer/Analyst	146	30.5
	Manager/Supervisor	81	16.9
	Project Team Leader	128	26.7
Organization Size	Ranged from 6 to 600,000 Mean = 23,583		
Department Size	Ranged from 2 to 50,000 Mean = 2,110		
Percentage of organization Using Methodology	Less than 10%	9	1.9
	10 – 20	9	1.9
	20 – 30	22	4.6
	30 – 40	21	4.4
	40 – 50	27	5.6
	50 – 60	43	9.0
	60 – 70	67	14.0
	70 – 80	81	16.9
	80 – 90	57	11.9
	More than 90	118	24.6
Amount of Tool Support	Ranged from 1 = Minimal to 7 = Full Mean = 4.65		
Years with Organization	Ranged from Less than 1 to 28 Mean = 2.88		
Years Development Experience	Ranged from Less than 1 to 35 Mean = 6.47		

Table 3.3 - continued

Months Using Methodology	Ranged from Less than 1 to 120 Mean = 29.5 months Less than 1 - 12 months = 127 (26.5%) 12 to 24 months = 139 (29%) (over half (55%) less than 2 years) 24 – 36 months = 74 (15.4%) 36 – 48 months = 50 (10.4%) 48 – 60 months = 34 (7.1%) Over 60 months = 32 (6.7%)		
Project Duration (in months)	Ranged from 1 to 168 Mean = 18.99 months		
Number of Team Members	Ranged from 1 to 500 Mean = 34.6		

3.5.2 Construct Operationalization

Appendix A contains the survey items along with their respective constructs, and indicates if the construct is formative or reflective. In Table 3.4 below, we list the source for the items for each of the constructs.

Table 3.4 Constructs and Measurements

CONSTRUCT	MEASUREMENT
Complexity, Relative Advantage, Social Influence, Voluntariness	questionnaire from Riemenschneider, *et al.*, 2002
Compatibility with prior experience, preferred work style and values	Karahanna, Agarwal and Angst, 2006
Process Agility	Process Agility Measurement Scale (developed by the authors, see Appendix B)
Competence, Autonomy Relatedness	(INS) Intrinsic Need Satisfaction scale from Baard, Deci and Ryan, 2004
Levels of Motivation	Situational Motivation Scale from Guay, *et al.*, 2000
Commitment	Fedor, D. B., Caldwell, S., and Herold, D. M., 2006
Absorptive Capacity	Pavlou and El Sawy, 2006

The Process Agility Measurement Scale was developed as part of this research. Refer to Appendix B for a summary of how the Process Agility Measurement Scale was developed and validated. All other scales have been adapted from previous studies. The scales for complexity, relative advantage, social influence and voluntariness were taken directly from Riemenschneider *et al.*, 2002. Measures for the various dimensions of compatibility were developed and validated by Karahanna, Agarwal and Angst (2006).

The items for competence, autonomy and relatedness measured by the Intrinsic Need Satisfaction scale are well established and have been widely tested. SDT is a well established theory and has been extensively tested in various areas such as education, work, sports, family relationships and health. Scales to measure the basic needs are available and have been thoroughly validated. The Basic Need Satisfaction at work scale (Deci *et al.*, 2001) with 21 items was modified and a shorter version of 12 items was used. The Situational Motivational Scale (SIMS) was developed and tested by Guay, *et al.*, 2000. It assesses the individual's current motivation in relation to a specific activity. The questionnaire is comprised of 12 items that measure four subscales (Intrinsic motivation, autonomous (identified) regulation, controlled (introjected/external) regulation and amotivation). Consistent with the literature on weighting these dimensions (Vallerand, 1997, Grouzet *et al.*, 2004), we assigned weights of +2, +1, -1 and -2 to intrinsic motivation, identified regulation, external regulation and amotivation, respectively. Therefore, a positive weight is indicated for self-determined forms of motivation and a negative weight for non self-determined forms of motivation. Grouzet *et al.*, (2004) combined the four subscale scores into one motivation index. Vallerand *et al.*, (1997) used a prior version of the motivation scale and developed four indices for the different dimension of motivation and treated handled them as reflective indicators of a single motivation construct. In both cases, it was reported that this was done due to sample size restrictions. And, while these are valid approaches, we are not restricted by sample size. Therefore, used the weighted items as direct formative indicators of motivation allowing retention of the information in all of the indicators.

Commitment was measured using items validated and tested in (Fedor, *et al.*, 2006). These items were developed as a measure of change commitment.

The measurement items for absorptive capacity were taken from the New Product Development Dynamic Capabilities Instrument validated and tested by Pavlou and El Sawy, 2006.

3.6 Data Analysis

3.6.1 Analysis Method

Our research model has several mediators and tests multiple complex relationships. Because of this, structural equation modeling (SEM) is a suitable technique. Partial Least Squares (PLS) was chosen because it handles both formative and reflective indicators, whereas other SEM techniques do not. In addition, PLS is prediction oriented and it does not assume multivariate normality unlike SEM techniques such as LISREL. PLS Graph Version 3, Build 1130 (Courtesy of Dr. Chin, University of Houston) was the software tool used to conduct the analysis.

3.6.2 Data Screening

First, an exploratory analysis was performed on the data looking for outliers and missing data in the model. Responses with more than 10% of the data missing would be deemed unusable. Applying this criterion we retained all responses. Since we are computing subscale scores for the motivation construct, a mean substitution was used for any missing values contained within the items measuring motivation. Because PLS GRAPH can incorporate missing values, all other variables were coded with a global value of -1. We calculated standardized z-scores in order to conduct a univariate outlier analysis for the IV's Complexity, Compatibility, Relative Advantage, Social Influence and Process Agility. When dealing with large sample sizes, Hair, *et al.*, (1998) recommends a standardized score value of (+/-) 3 to 4 as a benchmark to identify outliers. Using this criterion, we found 2 responses that had an outlier in the Relative Advantage variable. However, they were only slightly outside of the acceptable range (-4.04 and -4.21). Upon inspection of the 2 cases, the items did contain valid values. Both cases were retained.

Table 3.5 below contains descriptive statistics for the scale score for the multi-item constructs. Most of the scale values show some skewness as well as kurtosis. However, since normality is not an assumption with PLS this is not a cause for concern (Chin, *et al.*, 2003).

Table 3.5 Descriptive Statistics

	Minimum	Maximum	Mean	S.D.	Skewness Statistic	Kurtosis Statistic
Autonomy	2.14	7.00	4.64	.692	.249	.936
Competence	2.83	7.99	5.08	.849	.061	-.913
Relatedness	3.13	7.00	5.06	.787	.122	-.424
Commitment	1.00	7.00	5.25	.973	-.471	.923
Complexity	1.00	6.33	2.92	.930	.550	.978
Compatibility	1.50	7.00	4.65	.833	.486	.625
Relative Advantage	1.00	7.00	5.27	1.00	-.962	1.992
Social Influence	1.00	7.00	5.02	1.13	-.802	1.182
Process Agility	1.73	7.00	4.74	.885	-.412	.400
	Skewness Std error	.114				
	Kurtosis Std. error	.227				

3.6.3 Control Variable Analysis

The seven control variables: Absorptive Capacity, Age, Culture, Development Experience, Experience using the Methodology, Gender and Voluntariness were entered into the model as IV's on the first level of DV's of autonomy, competence and relatedness. There is not a general consensus as to how to handle control variables in PLS, but in covariance based approaches such as LISREL, entering them as IV's on the first level of DV's is the common practice.

The control variable, "Months using the methodology", was significantly related to autonomy. Voluntariness was found to be negatively related to relatedness. And, Absorptive Capacity was significantly related to autonomy, competence and relatedness. These control variables were kept in the model. All other control variable relationships were found to be insignificant with a two tailed t-test and were dropped from the model.

3.6.4 Common Method Bias

Common method bias poses a threat to the validity of our conclusions and is of particular concern when dealing with self-reported measures. In order to assess the presence of common method bias, we conducted Harman's one-factor test (Podsakoff *et al.*, 1984). An exploratory factor analysis was conducted on all multiple items measures. Factors were extracted with eigenvalues greater than one. The unrotated solution generated 17 factors. One factor explained 24.3% of the variance and the remaining factors explained less than 10%. Total variance explained was 66.8%. After rotation, the percentage of variance explained by each factor was less than 10. Based on this analysis, there was no single factor which explained a substantial amount of the variance. Therefore, we believe that common method variance does not pose a significant threat to measurement validity to this study.

3.6.5 Measurement Model

We can evaluate the measurement model by assessing the indicator reliabilities, the convergent validity of the indicators for each construct, and the discriminant validity. We have four formative constructs: process agility, social influence, motivation and the control variable, voluntariness. Since it is not meaningful to calculate reliabilities of the indicators and convergent/discriminant validity for formative constructs, they are excluded from the measurement model analysis (Bagozzi, 1994; Bollen, 1989). In order to assess the validity of the measurement for these formative constructs, we used the procedure outlined by Diamantopoulos and Winklhofer (2001). This procedure entails checking for collinearity and excluding those indicators that are highly collinear as they will inflate the variance explained by the latent variable (Bollen and Lennox, 1991). The highest VIF among the three formative constructs was 3.8, which is much less than the heuristic of 10. The condition index values were all acceptable. There was a fairly high correlation between SOCINFL1 and SOCINFL2 (-0.677). However, it was decided that these indicators would be retained because dropping one would result in a 2 indicator measure, possibly losing a significant dimension of the construct.

To assess convergent validity for the reflective constructs, we analyzed the loadings of the indicators on their latent constructs. This resulted in 15 indicators with loadings less than .70 being dropped from the model. Three were dropped from compatibility (COMPATEXP2, COMPATVAL1, COMPATVAL2), four from autonomy (AUTON2R, AUTON4R, AUTON6, AUTON7R), three from competence (COMPET1R, COMPET5R, COMPET6R), one from complexity (COMPL3R) and four from relatedness (RELAT2, RELAT3R, RELAT6R, RELAT7R). After we dropped these indicators from the model, a bootstrap procedure (200 samples) was run to generate the composite reliabilities, the average variance extracted (AVE) and the t-statistics for the path coefficients. Table 3.6 presents the composite reliability of the construct, indicator loadings and their respective t-values. Composite reliability considers the actual loadings while calculating indicators and is considered superior to Cronbach's alpha in terms of measuring internal consistency (Ma and Agarwal, 2007).

According to Hair *et al.*, (1998), a composite reliability value greater than 0.70 is a good indicator of internal consistency. All of the values for the constructs are above 0.70 indicating that our model has good internal consistency. An AVE value greater than 0.5 indicates good convergent validity (Fornell *et al.*, 1981). All AVE values are above 0.5. Therefore, our data indicates that the measurement model is internally consistent and exhibits convergent validity.

Table 3.6 Indicator Loadings and Composite Reliabilities

	Composite Reliability	**AVE**	**Loading**	**Std. Error**	**T-Statistic**
Compatibility	0.861	0.676			
COMPATEXP1			0.69	0.046	14.93
COMPATWORK1			0.87	0.019	46.08
COMPATWORK2			0.88	0.025	34.61
Commitment	0.899	0.690			
COMMIT1			0.75	0.036	20.38
COMMIT2			0.89	0.011	33.79
COMMIT3			0.83	0.024	33.79
COMMIT4			0.84	0.024	34.59
Autonomy	0.772	0.531			
AUTON1			0.76	0.033	22.75
AUTON3			0.73	0.043	18.86
AUTON5			0.68	0.040	16.67
Competence	0.895	0.580			
COMPET2			0.68	0.056	12.13
COMPET3			0.78	0.032	24.00
COMPET4			0.80	0.025	31.62
Complexity	0.914	0.680			
COMPL1R			0.84	0.27	31.42
COMPL2R			0.84	0.22	37.33
COMPL4R			0.89	0.012	70.21
COMPL5R			0.75	0.040	18.72
COMPL6R			0.77	0.035	22.14
Relative Advantage	0.937	0.713			
RELADV1			0.89	0.013	64.13
RELADV2			0.91	0.011	81.27
RELADV3			0.86	0.020	42.51

Table 3.6 - continued

RELADV4			0.83	0.029	27.83
RELADV5			0.71	0.048	14.96
RELADV6			0.83	0.024	33.46
Relatedness	0.848	0.583			
RELAT1			0.79	0.021	36.93
RELAT4			0.76	0.031	24.42
RELAT5			0.71	0.043	16.65
RELAT8			0.76	0.037	20.42

The AVE analysis is presented in Table 3.7. The bolded diagonal elements in the table represent the square root of the AVE scores. The off-diagonal elements are the correlations between the constructs. The square root of the AVE is significantly higher than any correlations involving the construct. Basically, all constructs share greater variance with their own measures than with the other constructs in the model. Therefore, our data exhibits discriminant validity.

Table 3.7 AVE and Construct Correlations

	COMPAT	**COMMIT**	**AUTON**	**COMPET**	**COMPLEX**
COMPAT	**0.82**				
COMMIT	0.61	**0.83**			
AUTON	0.37	0.41	**0.72**		
COMPET	0.36	0.43	0.61	**0.76**	
COMPLEX	-0.67	-0.6	-0.36	-0.34	**0.82**
RELADV	0.67	0.77	0.4	0.44	-0.66
RELATE	0.31	0.4	0.64	0.66	-0.3
	RELADV	**RELAT**			
RELADV	**0.84**				
RELAT	0.39	**0.76**			

3.6.6 Hypothesis Tests

Next we evaluated the structural model. Bootstrap output from PLS Graph was obtained in order to get path coefficients with their associated t-values. Results of the Hypotheses tests are summarized in Table 3.8 and shown in Figure 3.4.

Table 3.8 Summary of Hypothesis Tests

	Hypothesis	Result
H1	Complexity will be negatively related to the level of competence.	Supported
H2	Compatibility will be positively related to the level of competence.	Supported
H3	Compatibility will be positively related to the level of autonomy.	Supported
H4	Perceived relative advantage will be positively related to autonomy.	Supported
H5	Social Influence will be positively related to relatedness.	Supported
H6	Process Agility will be positively related to competence.	Supported
H7	Process Agility will be positively related to autonomy.	Supported
H8	Process Agility will be positively related to relatedness.	Supported
H9	The perception of competence while using the methodology will be positively related to motivation.	Supported
H10	The perception of autonomy while using the methodology will be positively related to motivation	Supported
H11	The perception of relatedness while using the methodology will be positively related to motivation.	Supported
H12	Higher levels of motivation will be associated with higher levels of commitment.	Supported

The control variables, number of months using the methodology, voluntariness and absorptive capacity, were all significant. Experience with the methodology was positively related to autonomy indicating that the longer a developer had worked with the methodology the higher their feelings of autonomy. Voluntariness was negatively related to relatedness meaning that if the developer was not required to use the methodology they had lower feelings of relatedness. There was a strong relationship between the developer's perception of their organization's absorptive capacity and their feelings of competence, autonomy and relatedness. Therefore, if a developer perceived that their organization did a good job of assimilating new knowledge into the processes and products of the firm, developers had higher feelings of competence, autonomy and relatedness.

A sizeable amount of the variance in competence, autonomy and relatedness was explained. SDT's posited relationship between the three needs and motivation was validated as was the positive relationship between motivation and commitment. All hypotheses for the relationships between the general innovation characteristics and the needs were supported. Process agility had strong positive relationships with all three needs.

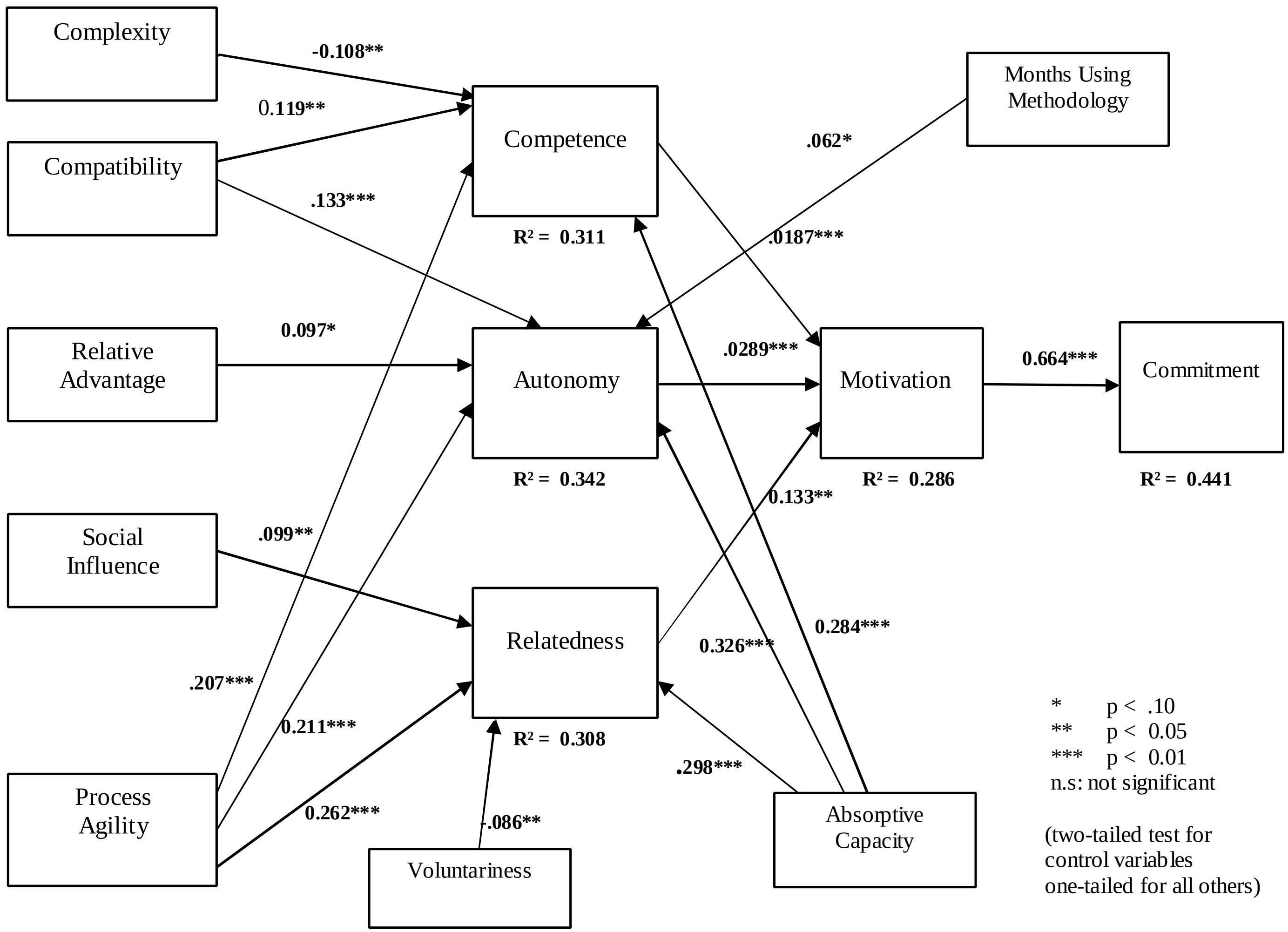


Figure 3.4 Structural Model

3.7 Discussion and Conclusions

In this research, we examined the impact of general innovation process characteristics and specific development process characteristics on a developer's commitment to use a given development methodology. As part of this examination, we investigated how and why these factors impact commitment to usage. All hypothesized paths were found to be significant. Our study confirmed the strong positive relationship between motivation and commitment found in prior research (Deci and Ryan, 2000; Gagne and Deci, 2005). Although competence, autonomy and relatedness all had a significant impact on motivation, autonomy had the strongest effect and relatedness had the weakest. The major impact of autonomy on motivation is not altogether surprising, considering that autonomy has been identified as the most important of the three needs in the integration process (Gagne and Deci, 2005). And, integration represents the highest degree of internalization of external factors (Deci and Ryan, 2002) thus, leading to the highest level of self-determined motivation. The significant, but, relatively weak relationship between relatedness and motivation is consistent with prior research on motivating IT professionals. Prior research in this area has shown that as compared to other professions, IT personnel have the lowest social needs strength. Social needs strength is defined as, the degree to which an individual needs to interact with others," (Couger and Zawacki, 1980). Similarly, relatedness is the degree to which individuals need to feel a connectedness or belongingness to others.

There was a strong positive relationship between compatibility and autonomy, providing support for the idea that if developers believe that the methodology fits in with the way they prefer to work, they feel as if using the methodology is a conscious choice and will experience increased feelings of autonomy. Compatibility with past work experience has been shown to have a positive impact on ease of use (Karahanna et al., 2006) and, if individuals find the methodology easier to use, they will have greater amounts of self-efficacy regarding its use. Higher self-efficacy beliefs lead to greater efforts (Bandura, 1986), increasing the chance of success, resulting in increased feelings of competence. Consistent with this supposition, we found a positive relationship between compatibility and competence. Conversely, complexity was found to be negatively related to competence. If developers perceived the methodology to be difficult to use, they experienced lower levels of self-efficacy regarding its use, leading to less effort, decreasing their chance of success and resulting in diminished feelings of competence. Although, relative advantage had a significant effect on autonomy, it was relatively weak as compared to the other innovation characteristics. Our operationalization of compatibility did not include compatibility with needs, however, it is still possible that there was a confounding effect between the constructs of relative advantage and compatibility as suggested by (Karahanna, *et al.*, 2006). If a methodology is compatible with a developer's needs, by definition, it can also be considered to provide a relative advantage. On the other hand, our finding could be attributable to the explanation put forth by Reimenschneider, *et al.,* (2002), that usefulness or the relative advantage of adoption of a methodology takes longer to realize than that of tool studies. This explanation is less likely, however, because our sample represents varying amounts of experience with the methodology. Social influence was shown to have a significant effect on relatedness. This supports the need for developers to have a feeling of belongingness or connectedness to individuals in order for those individuals to influence their behavior regarding use of the methodology.

Process agility had a much higher impact on the three needs than any of the innovation factors. The effect of process agility on competence, autonomy and relatedness was significant at the (.01) level. While the innovation characteristics do have a significant effect on the needs, they are each fairly unidimensional with the exception of compatibility. Process agility, however, has the potential to permeate virtually every aspect of the developer's work environment. It impacts vital aspects of their job from how they interact with co-workers to how their roles are determined. In an agile development environment, developers are cross-trained so that they can work on various facets of the project. This increases their efficacy regarding their ability to achieve desired outcomes in terms of producing working software. Since agile development is characterized by evolutionary development accomplished by frequent delivery of small releases, the process involves actively engaging customers. Key elements of this evolutionary process are feedback and communication which serve to guide the efforts of the developers. This continual feedback loop quickly gives developers a sense of what works and what doesn't, resulting in continuous learning and adjusting of behaviors, thereby enhancing their skills and increasing their feelings of competence. Process agility is also characterized by flexibility in terms of developers' roles and processes. Agile methods advise that developers be empowered to make decisions without having to go through a rigid chain of command (Highsmith, 2002). Developers experience increased levels of autonomy in an agile environment because they are given freedom to deviate from established procedures in order to deliver working software (Highsmith, 2002; 2004). In an agile development environment, developers experience increased feelings of relatedness because agile methods are characterized by continuous collaboration between developers and users. This constant communication coupled with the self-organizing nature of agile teams where developers must work closely together, increase developers' feelings of relatedness.

Of the control variables, absorptive capacity had a significant (.01) positive effect on all three needs, voluntariness had a significant negative effect (.05) on relatedness and months using the methodology had a significant (.10) positive, yet weak, effect on autonomy. Absorptive capacity refers to an organization's ability to acquire new knowledge, internally as well as externally, and assimilate it into their processes and products. Its significant effect on all three needs suggests focusing on increasing the absorptive capacity of the development organization could have a far reaching effect. Voluntariness is whether use of the methodology is mandatory. We found that if the developer was not required to use the methodology they had lower feelings of relatedness. It is possible that if the developer is not required to use the methodology, there is less group cohesiveness resulting in a decreased sense of connectedness or belongingness. And, the longer a developer had been using the methodology, the higher their feelings of autonomy. This suggests that time may play a role in the internalization process. Another possible explanation is that the more experience the developer has with the methodology, the more comfortable they feel acting on their own behalf in terms of free choice and decision-making. Since these important control variables all have a significant influence on the dependent variables, removing these effects prevents them from contaminating the significant impact of the independent variables (complexity, compatibility, relative advantage, social influence and process agility), on the dependent variables. Thus, we can conclude with a high degree of confidence that our findings regarding the hypothesized relationships are solely caused by the independent variables.

3.7.1 Limitations

This study involved self-report surveys. One limitation of self-report studies is the risk of common method bias. In order to mitigate this risk we have conducted the Harman's one-factor test and the results indicated that common method variance could not explain a significant part of the variance explained by the model. No single factor contributed to a majority of the variance. Results of the rotated solution resulted in all factors contributing less than 10% to the total variance explained.

In terms of generalizability, one possible limitation is that approximately 80% of our respondents were from one country with the remaining 20% being spread among various countries. However, no significant differences have been found in cross cultural studies related to the high growth needs attributed to information technology workers (Couger and Motiwalla, 1985; Couger, *et al.*, 1990). And, cross cultural studies using Self-Determination theory have shown support for the relationship between the three needs and motivation in various cultures (Chirkov, *et al.*, 2003; Deci, *et al*., 2001). Additionally, we added culture as a control variable to the model and it did not have a significant effect.

3.7.2 Contributions

This study has made valuable contributions in the area of software development and specifically in the area of systems development methodology implementation. Using theories from both psychology and marketing, we developed a model to predict developers' intention to commit to using a given Systems Development Methodology. This included identifying certain general innovation characteristics and specific development process characteristics that significantly impact developer motivation and commitment. For methodology implementation in general, and agile methods in particular, this study has suggested that agile development methods lead to increased developer motivation and commitment. Applying the theory of Self-Determination, we found support for the mediating effect of three innate psychological needs on developers' intention to commit to using a given systems development methodology.

Researchers have called for additional research to integrate findings of the various models as well as identify and integrate additional determinants of methodology acceptance (Reimenschneider, *et al*., 2002). While most studies have focused on acceptance of products, this is one of the relatively few studies concerned with acceptance of processes and specifically examination of the intention to adopt systems development methodologies (Toleman, *et al*., 2004). Another distinguishing factor of this research is that it looked at commitment to usage, while most prior technology acceptance models have had behavioral intention and self-reported usage as the dependent variables (Lee, *et al.*, 2003). Commitment to usage represents a more proactive measure than intention. When developers are committed to using a given methodology, they have "embraced" its usage and are dedicated to making it successful. While effectiveness of a methodology is certainly important, if developers do not accept the methodology and commit to its use, the benefits of the methodology will not be fully realized. This study has contributed toward providing clarification for the underlying reasons of the effects of certain factors on a developers' acceptance of and commitment to using the methodology. A major strength of this study is that we have examined usage through the theoretical lens of Self-Determination Theory. To our knowledge, this is the first time that Self-Determination Theory has been applied within the domain of software development methodology acceptance. While TAM (Davis, 1989) looked at the cognitive process underlying the adoption decision, Self-Determination Theory explores the general psychological/affective factors for adopting an innovative process.

Another major contribution of this research is the investigation of the role of process agility on developer motivation and commitment. Agile development is rapidly gaining acceptance with a recent survey reporting that 69% of respondents work in organizations that are currently using agile methods (Ambler, 2007). However, the role of agility in the acceptance/rejection of methodologies has never been explored. This study investigated the impact of the methodology's level of agility on a developer's motivation and commitment to use and support the methodology. We have built upon prior research by integrating Self-Determination Theory (SDT) into current usage models, thereby providing an explanatory link between the specific development process characteristics and individual commitment to usage.

For practitioners, we have provided valuable insights into the underlying factors that influence a developer's commitment to using the methodology. This information will enable organizations to develop more effective strategies for choosing and implementing development methodologies. Our findings are welcome news to the practitioners of agile methodologies, since, indeed, they have positive and beneficial effects on developers' self-determination needs, which should lead to higher motivation and commitment. These outcomes, in turn, should lead to better morale and greater productivity. In addition to the agility of the methodology, we have found that certain general innovation characteristics are associated with a developer's commitment to using the methodology. For example, the significance of complexity and compatibility on user acceptance, emphasize the importance of training. Having developers try the methodology prior to its implementation may increase their feelings of competence in terms of its use. And, if developers become familiar with the methodology, it should have a positive impact on compatibility. In order to improve a developer's perception of relative advantage, introduction strategies should provide salient examples of how it will improve their job performance. The impact of social influence confirmed the importance of having a champion or change agent to support implementation of the methodology. However, it is important that the champion be someone to whom the developers have a degree of relatedness. The results of this study indicate that even incorporating some of the agile tenets may have a positive impact on developer motivation. Finally, organizations should focus on increasing their absorptive capacity as our results show that absorptive capacity has a significant association with increased levels of competence, autonomy and relatedness, leading to increased developer motivation and commitment to using and supporting the methodology.

3.7.3 Future Research

In terms of future studies, we would encourage researchers to further validate the model in different contexts with a variety of methodologies. Our findings should be replicated and extended. Any additional determinants of commitment found to be significant in acceptance studies should be added to the model and tested. For example, in our study, we examined the impact of four general innovation characteristics and one specific methodology characteristic. However, there are probably more methodology characteristics that could be added as independent variables to determine how they impact developer motivation and commitment. In a similar vein, other outcome variables of interest could be investigated such as, quality of software produced, time needed to complete the project and developers satisfaction.

It would be beneficial to conduct a longitudinal study designed to examine a developer's level of commitment over time to see if level of commitment changes based on how long the developer has been using the methodology. Also, specifically investigating the impacts of agile methodologies over time would provide insight into the sustainability of their benefits in terms of characteristics such as continuous learning.

While this research addressed the characteristics of the methodology, it would also be of interest to investigate the relationship between process agility and leadership style. For example, a manager's leadership style could impact the agility of a given process regardless of the intended agility of the methodology. How does leadership style affect developer motivation in the context of agile methodologies? Self-Determination Research has shown a relationship between manager's autonomy supportive behaviors and fulfillment of the three needs (Deci, *et al.*, 2001). This same phenomenon has been studied in a sports team environment, where coach's leadership style in terms of autonomy supportive or controlling was shown to impact motivation (Pelletier, *et al.*, 2001). Self-Determination Theory could be used to examine leadership style in a software development environment to determine how it impacts developer motivation.

It has been suggested that agile methodologies may not be the best approach to use in all software development situations (Boehm and Turner, 2004). For example, a large complex project may not be suitable for using agile methods. Studies to measure the relationships between level of agility, project outcomes and various characteristics of projects would provide insight as to when it is best to use agile methods. Effectiveness of agile methods on projects with various characteristics would provide insight as to when it is best to apply agile methods.

CHAPTER 4

EMBRACING CHANGE: WHAT MOTIVATES SOFTWARE PROFESSIONALS?

4.1 Introduction

Earlier this year, InformationWeek reported the filing of a lawsuit against SAP for the "complete failure" of a $100 million software implementation. The client, Waste Management Inc., claims that SAP was misrepresenting their product by presenting demos of "fake mock-up simulations" of software with "false functionality"(Weier, 2008). Of the $2.5 trillion spent on IT during 1997 – 2001, almost $1 trillion went toward underperforming projects (Benko and McFarland, 2003). Based on the history of software development project success, it becomes increasingly apparent that the job of creating high quality software within a given budget and timeframe is a difficult prospect. Many methods have been tried from the very rigid traditional approaches to the very flexible agile approaches. And, we have found that there is not one standard approach that works in every situation. In fact, it is recommended that development organizations use a combination of practices that work best for a given set of project characteristics (Boehm and Turner, 2004). Research studies, examining some of the new agile methodologies, have shown promising results in terms of increased quality, satisfaction and productivity (Nosek, 1998; Kessler, 1999; Parsons *et al.*, 2007; Williams, 2000; Upchurch, 2001). By incorporating some of these agile practices, organizations may be able to reap some of the benefits of this approach.

Different methodologies will exhibit varying degrees of agility on a continuum from "very rigid" to "very flexible". With that said, software developers are sometimes reluctant to embrace the change required to implement new methods. Therefore, it would be beneficial if we could figure out what motivates them to embrace or resist a given methodology. It should be clarified that when we use the term "embrace" within this context we are referring to a developer committing to the use of the methodology in terms of being dedicated to its successful implementation and being supportive of its use. While an organization can and typically does mandate the methodology that is to be used, we believe that it is the developer's commitment (or lack thereof) to its success that has a significant impact on the implementation outcome. Therefore, identifying factors that motivate developers to commit to usage of the chosen methodology could provide valuable insights into ways of successfully introducing and employing these new methods.

In order to explore these questions, an empirical study was conducted involving 479 software developers from various organizations. The following sections will provide an explanation of the study as well as present the results and discuss implications for software development managers.

4.2 What *Does* Motivate Software Professionals?

Past research has shown that programmers have a high need for work that is challenging. A landmark study exploring what motivates information technology (IT) employees, found that IT personnel have a very high need for jobs that provide opportunities to grow. It was found that meaningful jobs, those that challenge and stretch employees, rank high on dimensions such as autonomy and feedback.

Job dimensions found to motivate IT personnel align well with a psychological theory known as Self-Determination Theory (SDT). The idea is that every human being has basic psychological needs, and fulfillment of these needs leads to psychological well being and higher levels of self-determined motivation. The result of this increased motivation has been shown to lead to higher levels of commitment on the job. The three needs from SDT are autonomy, competence and relatedness. Autonomy is associated with acting from one's sense of self and involves choice. Competence is defined as individuals feeling that they have succeeded at tasks that are optimally challenging. And, Relatedness is characterized as a feeling of belongingness or connectedness to others.

4.3 Which Methodology Characteristics Impact Commitment?

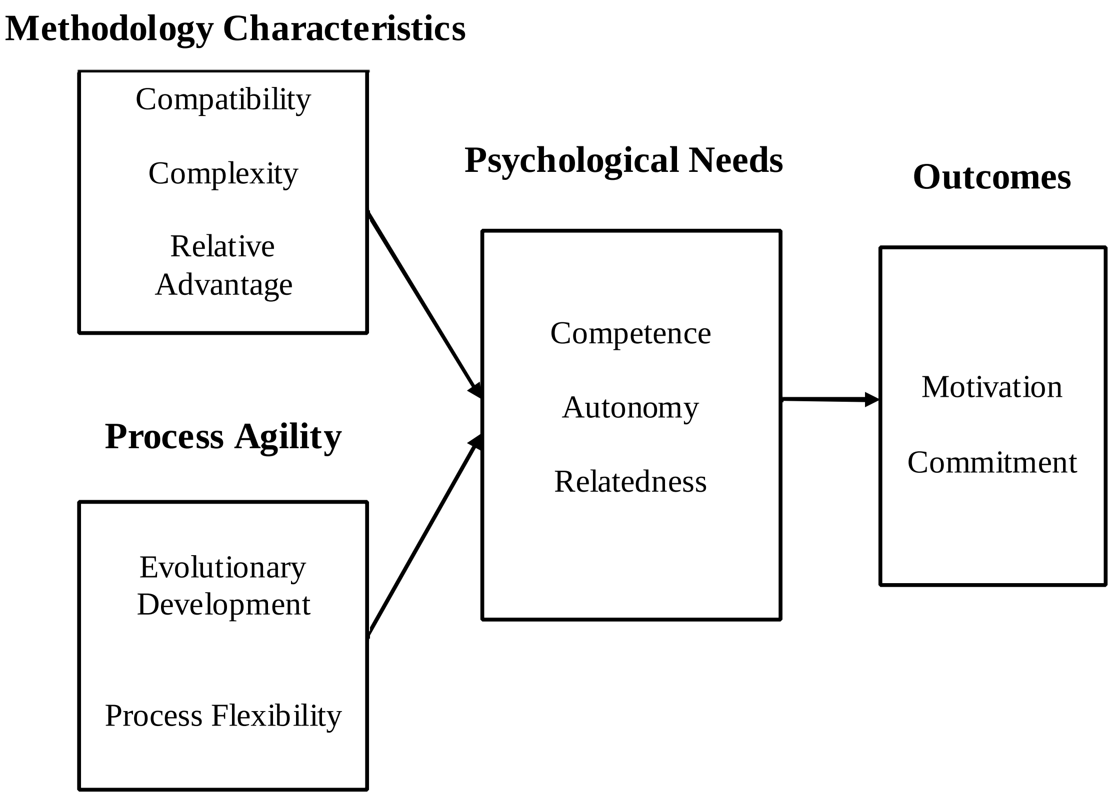


Figure 4.1 Factors Influencing Needs Fulfillment of Software Professionals

While there is much research on user acceptance of technology products, studies to determine characteristics of software development methodology acceptance are not as plentiful. However, based on prior user acceptance research and the methodology studies that have been done, it appears that three characteristics continue to be significant in terms of acceptance. We included these four in our study. They are complexity, compatibility and relative advantage. Complexity entails how hard the developer thinks the methodology is to understand and use. Compatibility is how much the developer feels that the methodology fits in with their way of doing things. This could be based on their prior experience or preferred work style. Relative advantage refers to how much the developer thinks the methodology helps them in improving job performance.

In addition to the methodology characteristics discussed above, we also investigated the process agility of the methodology to see how it impacts a developer's motivation. Agile processes assist the developer in responding quickly and effectively to a rapidly changing development environment. Evolutionary Development and Process Flexibility are two dimensions comprising Software Development Process Agility. Evolutionary development entails delivery of working software in short iterations in response to user feedback to address their evolving needs. Stakeholders are actively involved due to the collaborative nature of the process. It is characterized by adaptability and responsiveness to changing conditions. Process Flexibility captures the reaction capabilities that characterize agile development. It is concerned with flexibility of team members' roles and their readiness to deviate from established processes in order to deliver working software. Oftentimes, this involves improvising and crafting solutions on the fly.

Therefore, our study investigated the impact of the four factors discussed above: complexity, compatibility, relative advantage and process agility, and their role in fulfilling the needs of competence, autonomy and relatedness. As discussed earlier, fulfillment of these needs should result in increased developer motivation leading to increased commitment to usage of the methodology. All links in the model were found to be significant. These results indicate that the methodology characteristics do significantly influence the fulfillment of the three needs. Further, fulfillment of these needs did lead to higher levels of self-determined motivation, which lead to higher levels of commitment.

4.4 Results

As mentioned earlier, the study was conducted in a software developer population with 479 respondents. All of the respondents were working in organizations on software development projects. They were using various development methodologies and in some cases, a combination. The specific methodology being used was not so important to the study because we were measuring the degree of agility inherent in the process and its downstream impact on the developer's motivation as opposed to the degree of agility in any given methodology.

Characteristics of the methodology and Agility of the methodology were both significantly related to fulfillment of the psychological needs. If a software development methodology is less complex, more compatible with existing ways of doing work, provides more advantage, better enables evolutionary development, and is more flexible, then the developers can be expected to experience higher levels of fulfillment in terms of autonomy, competence and relatedness, leading to increased motivation and eventually a heightened sense of commitment.

The following methodology descriptions were reported by two respondents with high scores on compatibility, relative advantage and process agility.

> "We must react quickly to remain competitive in the markets we compete. Our goals and priorities are continually changing to meet the needs of our customer base balanced with those of our sales department."

> "Rapid design and visualization is a process that utilizes user centered design + application simulation for visualizing to be state and blueprinting of gray requirements prior to coding."

Tables 4.1 and 4.2 illustrate the average scores for needs fulfillment based on various levels of the factors. The factors in Table 4.1 represent an average of the high and low one-third of the individual factor scores. In Table 4.2, the factors represent an average of the high and low one-third taken from a composite of the factors. A composite of the factors was calculated by computing an average score for complexity, compatibility, relative advantage and agility.

Table 4.1 Average Needs Scores for Individual Factors

Factors	Competence	Relatedness	Autonomy
High Complexity	4.96	5.01	4.82
Low Complexity	5.57	5.53	5.45
High Compatibility	5.66	5.61	5.51
Low Compatibility	4.81	4.92	4.75
High Relative Advantage	5.78	5.67	5.57
Low Relative Advantage	4.76	4.85	4.68
High Agility	5.76	5.74	5.66
Low Agility	4.92	4.97	4.74

Table 4.2 Average Needs Scores for a Composite of Factors

Factors	Competence	Relatedness	Autonomy
Average High	5.80	5.77	5.68
Average Low	4.74	4.83	4.66

Table 4.3 contains a sampling of the questions used to measure the Factors and Needs.

Table 4.3 Sample of Measurement Items

Compatibility	Using the methodology is compatible with my past development experience.
	The methodology enables me to work in the way I prefer.
Complexity	Learning the methodology was easy for me.
	I think the methodology is clear and understandable.
Relative Advantage	Using the methodology increases my productivity.
	Using the methodology enhances the quality of my work.
Evolutionary Development	Our requirements specification process dynamically evolves through continuous feedback from users
	Developers communicate and collaborate with business people continuously to incorporate their evolving requirements.
Process Flexibility	The roles and relationships of our team members are flexible and not strictly defined.
	We don't mind deviating from established processes and procedures as long as we continuously deliver working software.
Competence	I have been able to learn interesting new skills on my job
	Most days I feel a sense of accomplishment from working
Autonomy	I feel like I can make a lot of inputs to deciding how my job gets done
	I am free to express my ideas and opinions on the job
Relatedness	I consider the people I work with to be my friends.
	People at work care about me.

4.5 Managerial Implications

First, we have developed a scale to measure the agility of any software development methodology. This is a valuable tool that can be used by managers to assess the level of agility inherent in their particular development methodology. Secondly, our study found that agile methods are, indeed, leading to desirable outcomes in terms of increased developer motivation and commitment. This is welcome news for practitioners of agile development. Even incorporating some of the agile tenets may have a positive impact. The positive relationships between process agility and increased feelings of competence, autonomy and relatedness mean that more process agility leads to increased developer motivation and less process agility (in terms of traditional more rigid development methods) results in decreased developer motivation.

Thirdly, we have provided insights into not only which factors impact a developer's commitment to the methodology, but the reasons behind it. This will inform better strategies for choosing and implementing development methodologies. And, while effectiveness of a given methodology is certainly a critical component of methodology choice, ensuring that developers fully utilize it is just as important. Complexity, compatibility, relative advantage and process agility were all found to be significant in terms of developer motivation and commitment. The significance of complexity and compatibility emphasize the importance of training. Having developers try the methodology and ensuring that they feel confident in their ability to use it successfully will increase their feelings of competence in terms of its use. They will also become familiar with it which should have a positive impact on compatibility.

Introduction strategies should provide salient examples of how it will improve their job performance, emphasizing its relative advantage. Looking at Table 4.2, autonomy and competence receive the greatest impact from these factors, emphasizing the important role these needs play in developer motivation and commitment. Therefore, management strategies and interventions should empower developers and give them freedom to participate in the decision-making process as well as provide tools and opportunities to gain a sense of accomplishment.

In summary, insights into factors affecting developer commitment to methodology usage and the reasons for their effect suggest better strategies for successful implementation of systems development methodologies as well as providing information to guide monitoring and prescribe necessary interventions. The measurement instrument used in this study could be employed as a tool in evaluating an organization's current climate in terms of methodology commitment and general levels of motivation, thus providing valuable information that can be used in managing the development process.

Table 4.4 provides a summary of key findings and their implications.

Table 4.4 Findings and Implications

Key Finding	Implication
Low Complexity and High Compatibility lead to increased motivation	Emphasizes importance of training and letting developers try the methodology to become familiar with it prior to actual implementation
If Developers perceive the methodology as providing an advantage in performing their job, they will have increased motivation	Training and introduction strategies should provide salient examples of how the methodology will improve job performance
Increased levels of process agility are associated with higher levels of motivation	Even incorporating some of the agile principles could have a positive impact on developer motivation

APPENDIX A

CONSTRUCT TABLE WITH ITEMS

Construct	Type	Items
Complexity	Reflective	Learning the methodology was easy for me.
		I think the methodology is clear and understandable.
		Using the methodology does not require a lot of mental effort.
		I find the methodology easy to use.
		The methodology is not cumbersome to use.
		Using the methodology does not take too much time from my normal duties
Compatibility	Reflective	Using the methodology is compatible with my past development experience.
		Using the methodology is not similar to anything that I've done before.
		The methodology enables me to work in the way I prefer.
		Using the methodology fits well with the way I like to work.
		Using the methodology does not fit the way I view the world.
		Using the methodology runs counter to my values about how to conduct my job.
Relative Advantage	Reflective	Using the methodology improves my job performance.
		Using the methodology increases my productivity.
		Using the methodology enhances the quality of my work.
		Using the methodology makes it easier to do my job.
		The advantages of using the methodology outweigh the disadvantages.
		The methodology is useful in my job.
Social Influence	Formative	People who influence my behavior think I should use the methodology.
		People who are important to me think I should use the methodology.
		Coworkers think I should use the methodology.

Process Agility	Formative	We frequently develop working software that is tested, integrated and executable as a partial system.
		Adjustments and refinements to requirements are always welcome at any stage of the development process.
		We constantly seek users' feedback to shape new requirements and re-prioritize features of the system.
		Our requirements specification process dynamically evolves through continuous feedback from users.
		We meticulously document every aspect of the system throughout the development cycle.
		Our initial system plan consists of minimal, yet essential requirements without complete and detailed specifications.
		We believe changing requirements are normal and help to enhance the system quality.
		Developers communicate and collaborate with business people continuously to incorporate their evolving requirements.
		Our project schedules and estimates are determined up front and are not subject to change.
		We improvise and experiment with new ways of doing things which may differ from the old routines.
		The roles and relationships of our team members are flexible and not strictly defined.
		We don't mind deviating from established processes and procedures as long as we continuously deliver working software.
		We use short iterations of fixed intervals to quickly design, implement and test a small subset of the requirements.
		Working software is the primary measure of progress.
Process Agility Indicator	Reflective	Overall, our development process is adaptive and responsive to changing user needs.
		In general, our development process is flexible with minimal planning.

Competence	Reflective	I do not feel very competent when I am at work.
		People at work tell me I am good at what I do.
		I have been able to learn interesting new skills on my job
		Most days I feel a sense of accomplishment from working
		On my job I do not get much of a chance to show how capable I am
		When I am working I often do not feel very capable
Autonomy	Reflective	I feel like I can make a lot of inputs to deciding how my job gets done
		I feel pressured at work
		I am free to express my ideas and opinions on the job
		When I am at work, I have to do what I am told.
		My feelings are taken into consideration at work
		I feel like I can pretty much be myself at work
		There is not much opportunity for me to decide for myself how to go about my work.
Relatedness	Reflective	I really like the people I work with
		I get along with people at work.
		I pretty much keep to myself when I am at work
		I consider the people I work with to be my friends.
		People at work care about me.
		There are not many people at work that I am close to.
		The people I work with do not seem to like me much.
		People at work are pretty friendly towards me.

Motivation	Formative	Because I think that the methodology is interesting..
		Because I am using the methodology for my own good.
		Because I am supposed to use the methodology.
		There may be good reasons to use the methodology, but personally I don't see any.
		Because I think that using the methodology is pleasant.
		Because I think that using the methodology is good for me.
		Because using the methodology is something that I have to do.
		I use the methodology but I am not sure if it is worth it.
		Because using the methodology is fun.
		It is my personal decision to use the methodology.
		Because I am required to use the methodology
		I don't know; I don't see what using the methodology brings me.
		Because I feel good when using the methodology.
		Because I believe that using the methodology is important for me.
		Because I feel that I have to use the methodology.
		I use the methodology, but I am not sure it is a good thing to pursue it.
Commitment	Reflective	I am doing whatever I can to help this methodology be successful.
		I am fully supportive of this methodology.
		I have tried (or intend to try) to convince others to support this methodology.
		I intend to fully support my supervisor in the implementation and/or continued use of this methodology.

Absorptive Capacity	Reflective	We are successful in learning new things within this group.
		We are effective in developing new knowledge or insights that have the potential to influence product development.
		We are able to identify and acquire internal (e.g.,within the group) and external (e.g.,market) knowledge.
		We have effective routines to identify, value, and import new information and knowledge.
		We have adequate routines to analyze the information and knowledge obtained.
		We have adequate routines to assimilate new information and knowledge.
		We can successfully integrate our existing knowledge with the new information and knowledge acquired.
		We are effective in transforming existing information into new knowledge.
		We can successfully exploit internal and external information and knowledge into concrete applications.
		We are effective in utilizing knowledge into new products.
Voluntariness	Formative	Although it may be helpful, using the methodology is certainly not compulsory in my job.
		My supervisor does not require me to use the methodology.
		My use of the methodology is voluntary.

APPENDIX B

DEVELOPMENT OF PROCESS AGILITY MEASUREMENT SCALE

The instrument development process consisted of 3 stages: 1) item creation; 2) scale development; and 3) instrument testing. During item creation an initial pool of 14 items was created based on the 12 Principles behind the Agile Manifesto (http://agilemanifesto.org/principles.html). The items went through several iterations of review and refinement by academics knowledgeable in survey design as well as software methodologies. During the scale development stage, a panel of 4 experts with extensive software development methodology experience rated the items and sorted them into categories. After making changes based on input from the panel, the instrument was pilot tested with 20 respondents. Most of the respondents were developers from industry who closely resemble the target population. Based on comments from the pilot, minor changes were made. Next, the instrument was field tested using an online survey accessible via the internet. This resulted in a total of 479 usable surveys. The respondents were software developers working on software projects and using a software development methodology.

An Exploratory Factor Analysis performed on the data resulted in two components for the process agility construct. After interpretation, the components were labeled Evolutionary Development and Process Agility. The finalized agility scale consists of 11 items which measure two underlying factors: Evolutionary Development and Process Flexibility. The first factor, Evolutionary Development, captures the dynamic aspect of an agile process that strives to deliver working software in short iterations by anticipating and embracing change, actively involving stakeholders and continually seeking users' feedback to address their evolving needs. The items capture the collaborative dimension of agile processes coupled with the adaptability and responsiveness to changing conditions. The second factor, Process Flexibility, captures the reaction capabilities inherent in agile processes. Process Flexibility is characterized by flexibility of team members' roles and the willingness to deviate from established processes in order to deliver working software. It is the ability and willingness to improvise and come up with new solutions on the fly.

Next we modeled Process Agility as a 2^{nd} Order Formative Construct with Evolutionary Development and Process Flexibility as First Order Formative Constructs representing two dimensions of process agility. We used the method of repeated manifest variables whereby the second-order construct of Process Agility was created by using all indicators from each of the two first-order constructs (Chin, 2000). Further assessment of construct validity was accomplished by modeling agility as a second-order formative construct and testing whether it is highly correlated with its indicator (Diamantopoulos and Winklhofer, 2001; Karimi *et al.*, 2007; Pavlou and El Sawy, 2006). For the Process Agility Indicator, two items were included in the questionnaire to measure the overall agility of the development process. The analysis was performed using a Partial Least Squares and Structural Equation Modeling tool PLS-GRAPH Version 3.0, Build 1130 (Courtesy of Dr. Chin, University of Houston). The Process Agility aggregate construct was highly correlated with its indicator illustrating that it is describing what it is intended to measure and exhibiting good construct validity.

Finally, we assessed the predictive validity of the instrument by using it to test hypotheses related to a positive relationship between the agility of a methodology and a developer's motivation to commit to using the methodology. The instrument demonstrated good predictive validity. Items that comprise the Agility Measurement Scale are illustrated in the following table.

Process Agility	Formative	Our requirements specification process dynamically evolves through continuous feedback from users.
		We constantly seek users' feedback to shape new requirements and re-prioritize features of the system.
		Developers communicate and collaborate with business people continuously to incorporate their evolving requirements.
		We frequently develop working software that is tested, integrated and executable as a partial system.
		Our initial system plan consists of minimal, yet essential requirements without complete and detailed specifications.
		Adjustments and refinements to requirements are always welcome at any stage of the development process.
		We believe changing requirements are normal and help to enhance the system quality.
		We don't mind deviating from established processes and procedures as long as we continuously deliver working software.
		The roles and relationships of our team members are flexible and not strictly defined.
		Working software is the primary measure of progress.
		We use short iterations of fixed intervals to quickly design, implement and test a small subset of the requirements.
Process Agility Indicator	Reflective	Overall, our development process is adaptive and responsive to changing user needs.
		In general, our development process is flexible with minimal planning.

REFERENCES

Abrahamsson, P., Warsta, J., Siponen, M.T., and Ronkainen, J. 2003. "New Directions on Agile Methods: A Comparative Analysis". Paper presented at the 25th International Conference on Software Engineering 2003 Proceedings.

Agarwal, Ritu and Prasad, Jayesh, "The Role of Innovation Characteristics and Perceived Voluntariness in the Acceptance of Information Technology", *Decision Sciences*, Vol. 28, No. 3, Summer, 1997, pp. 517-582.

Agarwal, Ritu and Prasad, Jayesh, "A Field Study of the Adoption of Software Process Innovations by Information Systems Professionals", *IEEE Transactions On Engineering Management*, Vol. 47, No. 3, August, 2000.

AgileAlliance "Manifesto for Agile Software Development," 2001 (available online at http://www.agilemanifesto.org).

Agility Forum (1998), website: www.AgilityForum.Org

Ambler, Scott. "Survey Says…Agile Has Crossed the Chasm," *Dr. Dobb's Journal,* July 2, 2007.

Ambler, Scott W. "The Extreme Programming Software Process Explained," *Computing Canada*, Vol. 26, Issue 5 (March 2000): 24-26.

Ambler, Scott W. "Adopting Extreme Programming," *Computing Canada*, Vol. 26, Issue 8 (April 2000): 26-27.

Ambler, Scott W. "Values, Principals and Practices Equal Success," *Computing Canada*, Vol. 27, Issue 10 (May 2001): 11-15.

Anderson, D.J. (2004) *Agile Management for Software Engineering*, Prentice Hall, Upper Saddle River, New Jersey.

Arteta, B.M. and Giachetti, R.E. (2004) 'A measure of agility as the complexity of the enterprise system', *Robotics and Computer-Integrated Manufacturing*, Vol. 20, pp.495–503.

Asproni, Giovanni (2004) 'Motivation, Teamwork, and Agile Development", *Agile Times, Vol. 4, February, 2004.*

Baard, Paul P., Deci, Edward L. and Ryan, Richard M. (2004), "Intrinsic Need Satisfaction: A Motivational Basis of Performance and Well-Being in Two Work Settings", *Journal of Applied Social Psychology,* 34(10) pp. 2045-2068.

Bagozzi, R. P. (1994). Structural Equation Models in Marketing Research: Basic Principles. In R. Bagozzi (Ed.), *Principles of Marketing Research* (pp. 317-385). Backwell: Oxford.

Balijepally, Venugopal, (2006), "Task Complexity and Effectiveness of Pair Programming: An Experimental Study", Phd. Dissertation, The University of Texas at Arlington, 2006.

Bandura, A. (1978), "Reflections on self-efficacy", *Advances in Behavioural Research and Therapy, 1*, 237-269.

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

Barki, Henri, Titah, Ryad and Boffo, Celine. (2007). "Information System Use-Related Activity: An Expanded Behavioral Conceptualization of Individual-Level Information System Use", *Information Systems Research,* Vol. 18, No. 2, pp. 173-192.

Beck, K. (2000), *Extreme Programming Explained*, Addison-Wesley, New York.

Beck, K. and Fowler, M. (2001), *Planning Extreme Programming*, Addison-Wesley, New York.

Benko, C. and McFarlan, W. *Connecting the Dots: Aligning Projects with Objectives in Unpredictable Times*. Harvard Business School Press, Boston, 2003.

Biddle, S. J. H., Soos, I, & Chatzisarantis, N. L. D. (1999). Predicting physical activity intentions using goal perspectives and self-determination theory approaches. *European Psychologist*, *4*, 83-89.

Boehm, B. and Turner, R. (2004), *Balancing Agility and Discipline*, Addison-Wesley, New York.

Bollen, K. A. (1989). *Structural Equations with Latent Variables*. New York: John Wiley and Sons.

Bollen, K., and Lennox, R. (1991). Conventional Wisdom on Measurement: A Structural Equation Perspective. *Psychological Bulletin*, *110*(2), 305-314. Biocca,

Breu, Karin, Hemingway, Christopher J. and Strathern, Mark. (2001) "Workforce Agility: The New Employee Strategy for the Knowledge Economy", *Journal*

Of Information Technology, 17, 21-31.

Brown, R., Nerur, Sridhar and Slinkman, C. (2004) "Philosophical Shifts in Software Development", Proceedings of the Tenth Americas Conference on Information Systems, New York, New York, August 2004.

Burton-Jones, Andrew and Hubona, Geoffrey S. (2006) "The Mediation of External Variables in the Technology Acceptance Model", *Information and Management,* 43, 706-717.

Burton-Jones, Andrew (2007) "Veni, Vidi, Vici: Breaking the TAM Logjam," *Journal of the AIS*: Vol. 8, Article 14.

Campion, M. A., Medsker, G. J., & Higgs, C. A. (1993). Relations Between Work Group Characteristics and Effectiveness: Implications for Designing Effective Work Groups. *Personnel Psychology, 46*(4), 823-850.

Charette, R. (2003). "Challenging the fundamental notions of software development", *Agile Project Management*, Arlington, MA: Cutter Consortium.

Chin, G. (2004) *Agile Project Management*, AMACOM, New York.

Chin, Wynne W., (2000), "Partial Least Squares for Researchers: An Overview And Presentation of Recent Advances using the PLS Approach", Presented at the 2000 ICIS conference, see website: http://disc-nt.cba.uh.edu/chin/plsfaq/2ndorder.htm.

Chin, Wynne W. and Gopal, Abhijit. (1995) "Adoption Intention in GSS: Relative Importance of Beliefs", *DATA BASE ADVANCES*, Vol. 26, Nos. 2 and 3, 42-64.

Chin, W. W., B. L. Marcolin, P. R. Newsted. 2003. "A partial least squares latent variable modeling approach for measuring interaction effects: Results from a Monte Carlo simulation and anelectronic-mail emotion/adoption study", *Information Systems Research* 14(2) 189–217.

Chirkov, V. I., & Ryan, R. M. (2001). Parent and teacher autonomy-support in Russian and U.S. Adolescents: Common effects on well-being and academic motivation. *Journal of Cross Cultural Psychology, 32*, 618-635.

Chirkov, V. I., Ryan, R. M., Kim, Y. & Kaplan, U. (2003). Differentiating autonomy from individualism and independence: A self-determination theory perspective on internalization of cultural orientations and well-being. *Journal of Personality and Social Psychology, 84*, 97-110.

Clarke, Peter. "Euro Design Verification Services Startup Adding Engineers," *Electronic Engineering Times*, Issue 1173 (July 2001): 58-60.

Cockburn, A. and J. Highsmith. (2001) "Agile Software Development 2: The People Factor", http://Alistair.cockburn.us/crystal/articles/asdpf/ asd2peoplefactor.htm.

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

Conboy, Kieran and Fitzgerald, Brian. "Toward a Conceptual Framework of Agile Methods: A Study of Agility in Different Disciplines," *Foundations of Software Engineering Proceedings of the 2004 ACM workshop on Interdisciplinary Software Engineering Research*, 37-44/

Conger, S. (1994) *The New Software Engineering*, Wadsworth, Belmont, California.

Conrad, Bruce. "Taking Programming to the Extreme Edge," *Infoworld*, Vol. 22, Issue 30 (July 2000) 61.

Copeland, Lee. "Developers Approach Extreme Programming with Caution," *Computerworld*, Vol. 35, Issue 43 (September 2001): 7-10.

Costello, Anna B. and Osborne, Jason W. (2005). "Best Practices in Exploratory Factor Analysis: Four Recommendations for Getting the Most FromYour Analysis". *Practical Assessment, Research and Evaluation*, 10(7). Available online: http://pareonline.net/getvn.asp?v=10&n=7.

Couger, J. D., Adelsberger, H., Borovits, I., Zviran, M. and Motiwalla, J. "Commonalities in Motivating Environments for Programmers/Analysts in Austria, Israel, Singapore, and the U.S.A.", *Information and Management* 18: 41-46 (1990).

Couger, J. Daniel, and Motiwalla, Juzar, "Occidental Versus Oriental I.S. Professionals' Perceptions on Key Factors for Motivation", *Proceedings of the Sixth International Conference on Information Systems*, Indianapolis, IN, December, 1985, pp. 105-112.

Couger, J. Daniel and Zawacki, Robert A., (1980), *Motivating and Managing Computer Personnel*, Wiley Interscience, New York.

Craig, Annemieke, Paradis, Rose and Turner, Eva, "A Gendered View of Computer Professionals: Preliminary Results of a Survey", *ACM SIGCSE Bulletin*, Vol. 34, Issue 2 (June, 2002).

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

Davis, Fred (1989) "Perceived Usefulness, Perceived Ease of Use and User Acceptance of Information Technology", *MIS Quarterly*, (13) 2, pp. 319-340.=

Deci, E. L., and Ryan, R. M. (1985). *Intrinsic motivation and self-determination in human behavior*. New York: Plenum.

Deci, E.L.. and Ryan, R.M. (2000). "The 'What' and 'why' of Goal Pursuits: Human Needs and the Self-Determination of Behavior", *Psychological Inquiry,* 11(4), 227- 268.

Deci, E. L., Ryan, R. M., Gagné, M., Leone, D. R., Usunov, J., & Kornazheva, B. P. (2001). Need satisfaction, motivation, and well-being in the work organizations of a former Eastern Bloc country. *Personality and Social Psychology Bulletin, 27, 930-942*.

Diamantopoulos, A., and Winklhofer, H. M. (2001). Index Construction with Formative Indicators: An Alternative to Scale Development. *Journal of Marketing Research, 38*(2), 269-277.

D'Souza, D. E. and Williams, F. P. (2000), "Toward a taxonomy of manufacturing Flexibility dimensions," *Journal of Operations Management*, 18, 588-593.

Edmunds, J., Ntoumanis, N. & Duda, J. L. D. (2008). Testing a self-determination theory based teaching style in the exercise domain. *European Journal of Social Psychology, 38,* 375-388.

Edwards, Jeffrey R. (2001). “Dimensional Constructs in Organizational Behavior Research: An Integrative Analytical Framework”, *Organizational Research Methods, (*4), 144-192.

Ettlie, J. E. (1998), “R & D and global Manufacturing Performance”, *Management Science* , 44, 1, 1-11.

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

Fawcett, S. E. and Clinton, S.R. (1996), “Enhancing Logistics Performance to Improve the Competitiveness of Manufacturing Organizations, *Production And Inventory Management Journal*, First Quarter, 49-46.

Fedor, Donald B., Caldwell, Steven and Herold, David M. (2006). “The Effects of Organizational Changes On Employee Commitment: A Multilevel Investigation”, *Personnel Psychology,(59) 1-29*.

Fichman. R. Information technology diffusion: A review of empirical research. In J.I. DeGross. J.D. Becker, and J.J. Elam (eds.). *Proceedings of the Thirteenth* International Conference on Information Systems. Baltimore: ACM.1992, pp.195-206.

Fichman, Robert G., “The Role of Aggregation in the Measurement of IT-Related Organizational Innovation”, *MIS Quarterly*, Vol. 25, No. 4, December, 2001, pp. 427-455.

Fichman, Robert G. and Kemerer, Chris F., “Adoption of Software Engineering Process Innovations: The Case of Object Orientation”, *Sloan Management Review*. Winter 1993, Vol. 34, No. 2, pp. 7-22.

Fichman, Robert G. and Kemerer, Chris F., “The Assimilation of Software Process Innovations: An Organizational Learning Perspective”, *Management Science*, 1997, Vol. 43, No. 10, pp. 1345-1363.

Field, Andy. (2006), *Discovering Statistics Using SPSS,* SAGE Publications London.

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

Fitzgerald, Brian, “An Empirical Investigation into the Adoption of Systems Development Methodologies”, *Information and Management*, 34 (1998), pp. 317-328.

Fornell, C., and Bookstein, F. L. (1982). Two Structural Equation Models: LISREL and PLS Applied to Consumer Exit-Voice Theory. *Journal of Marketing Research, 19*(4), 440-452.

Forsyth, D. R. (1999). *Group Dynamics* (Third ed.). Belmont, CA: Wadsworth Publishing.

Gagne, Marylene and Deci, Edward L., “Self-Determination Theory and Work Motivation”, *Journal of Organizational Behavior,* 26 (2005) 331-362.

Gagne, M., Ryan, R. M., & Bargmann, K. (2003). Autonomy support and need satisfaction in the motivation and well-being of gymnasts. *Journal of Applied Sport Psychology, 15*, 372-390.

Gallivan, Michael J., “The Influence Of Software Developers’ Creative Style On Their Attitudes To and Assimilation of a Software Process Innovation”, *Information and Management*, 40 (2003) 443-465.

Gefen, David and Straub, Delmar., “A Practical Guide to Factorial Validity Using PLS-GRAPH: Tutorial and Annotated Example”, *Communications of the Association for Information Systems*, 16 (2005) 91-109.

Gerwin, D., (1993), “Manufacturing Flexibility: A Strategic Perspective”, *Management Science,* 39,4,395-410.

Glass, R.L., “A snapshot of systems development practice”, *IEEE Software. 16* 3 (1999) 110-111.

Goldman, S. L., Nagel, R. N., and Preiss, K. (1994), *Agile Competitors and Virtual Organizations: Strategies for Enriching the Customer,* Van Nostrand Reinhold, New York, New York.

Goldsborough, W. (1992), “Global Logistics Management: Gaining a Competitive Edge through integrated systems, “*Business Intelligence Program,* SRI International.

Gopalakrishnan, S. and Damanpour, F., “A Review of Innovation Research in Economics, Sociology and Technology Management”, *International Journal Of Management Science,* Vol. 25, No. 1, 1997, pp. 15-28.

Griffin, A. (1993), “Metrics for Measuring Product Development Cycle Time,” *Journal*

Of Product Innovation Management, 10, 112-125.

Grolnick, W. S., & Ryan, R. M. (1989). Parent styles associated with children's self-regulation and competence in school. *Journal of Educational Psychology, 81,* 143-154.

Grouzet, Frederick, M.E., Vallerand, Robert J., Thill, Edgar E. and Provencher, Pierre J. (2004). "From Environmental Factors to Outcomes: A Test of an Integrated Motivational Sequence", *Motivation and Emotion*, 24(3), 175-213.

Guay, F., Vallerand,R.J., and Blanchard, C. (2000). "On the Assessment of Situational Intrinsic and Extrinsic Motivation: The Situational Motivation Scale (SIMS)", *Motivation and Emotion,* (24)3, 175-213.

Gunasekaran, A. (1998), "Agile manufacturing: enablers and an implementation Framework", *International Journal of Production research,* 36, 5, 1223-1247.

Gunasekaran, A. (1999), "Agile manufacturing: A framework for research and Development", *International Journal of Production Economics,* 62, 87-105.

Gupta, A. K. and Somers, T. M. (1992), "The measurement of manufacturing flexibility," *European Journal of Operational Research,* 60, 166-182.

Hair, Joseph F. Jr., Anderson, Rolph E., Tatham, Ronald L., Black, William C. (1998). *Multivariate Data Analysis*, New Jersey: Prentice Hall.

Hardgrave. B.C. "When to prototype: Decision variables used in industry", *Information and Software Technology. 37. 2* (1995). 113-118.

Hardgrave, B. C. and Johnson R. A., "Toward an information systems development acceptance model: the case of object-oriented systems development", *IEEE*

Transactions on Engineering Management, 50(3), (2003) 322-336.

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

Herzberg, F., Maunser, B. and Snyderman, B. (1959), *The Motivation to Work,* John Wiley and Sons, Inc., New Yor, NY.

Highsmith, J. (2004), *Agile Project Management*, Addison-Wesley, New York.

Highsmith, J. (2002), *Agile Software Development Ecosystems*, Addison-Wesley, New York.

Highsmith, J. "Order for Free: An Organic Model for Adaptation", in L. L. Constantine, Ed., *Beyond Chaos: The Expert Edge in Managing Software Development*, Addison-Wesley, 2001, pp. 251-257.

Hugos, Michael H. "Opinion: The Agility Factor", *Computerworld,* August, 13, 2007.

IBM Global CEO Study 2006. "Expanding the Innovation Horizon", http://www-935.ibm.com/services/us/gbs/bus/html/bcs_ceostudy2006.html.

Jeffries, R., Anderson, A. and Hendrickson, C. (2001), *Extreme Programming Installed*, Addison-Wesley, New York.

Joshi, K.D. and Schmidt, Nancy L. "Is the Information Systems Profession Gendered?: Characterization of IS Professionals and IS careers, ACM SIGMIS Database, v. 37 n.4, Fall, 2005.

Karahanna, E., Agarwal, R., and Angst C. "Reconceptualizing Compatibility Beliefs in Technology Acceptance," *MIS Quarterly*, 30:4, December 2006, pp. 781 804.

Karimi, Jahangir, Somers, Toni M., and Bhattacherjee, Anol. (2007). "The Impact of ERP Implementation on Business Process Outcomes: A Factor-Based Study," *Journal of Management Information Systems,* 24(1) pp. 101-134.

Katayama, Hiroshi and Bennett, David. (1999), "Agility, Adaptability and Leanness: A Comparison of Concepts and A Study of Practice", *International Journal Of Production Economics",* 60-61, pp. 43-51.

Kemerer, C.F. How the learning curve affects case tool adoption. *IEEE Software. 9*. 3 (1992). 23-28.

Kessler, Robert R. and Laurie A. Williams. "If this is What It's Really Like, Maybe I Better Major in English: Integrating Realism into a Sophomore Software Engineering Course," Foundations in Education Conference, 1999. Downloadable from http://www.cs.utah.edu/~lwilliam/Papers/fie99.PDF.

Khalifa, Mohamed and Verner, June M. (2000), "Drivers for Software Development Method Usage," *IEEE Transactions on Engineering Management*, (47)3, August, 2000.

Kidd, Paul (2000), "Two definitions of agility, www. **Cheshire**-Henbury.com

Kim, Jae-On and Mueller, Charles W. (1978), *FACTOR ANALYSIS: Statistical Methods and Practical Issues*, London: Sage Publications.

Koste, L. L. and Malhotra, M. K., (2000), "A theoretical framework for analyzing the Dimensions of manufacturing flexibility", *Journal of Operations Management*, 192, 100-125.

Krievsky, J. (2002). "Continuous Learning", *Extreme Programming Perspectives*.

M. Marchesi, G. Succi, D. Wells and L. Williams. Boston, MA, Addison Wesley Professional, 299-310).

Layman, Lucas, Williams, Laurie and Cunningham, Lynn. (2006), "Motivations And Measurement in an Agile Case Study," *Journal of Systems Architecture,* 52, pp. 654-667.

Lee, Y., Kozar, K.A. and Larsen, K.R.T. "The Technology Acceptance Model: Past, Present, and Future," *Communications of the AIS* (12:50), 2003, pp. 752-780.

Legris, Paul, Ingham, John and Collerette, Pierre. Why Do People Use Information Technology? A Critical Review of the Technology Acceptance Model", *Information and Management*, Vol. 40, No. 3, January, 2003, pp. 191-204.

Larman, Craig (2004), *Agile and Iterative Development,* Addison-Wesley, New York.

Leffingwell, Dean (2007), *Scaling Software Agility,* Addison-Wesley, New York.

Loch, C., Stein, L., and Terwiesch, C. (1996), "Measuring Development Performance In Electronics Industry, *Journal of Product Innovation Management,* 13, 3-20.

Locke. E. A. (1976), "The Nature and Causes of Job Satisfaction", in Dunnette, M.D. (Ed.), *Handbook of Industrial and Organizational Psychology,* Rand McNally, Chicago, IL/

Ma, M., and Agarwal, R. (2007). Through a Glass Darkly: Information Technology Design, Identity Verification, and Knowledge Contribution in Online

Communities. *Information Systems Research*, *18*(1), 42-67.

Mageau, G. A. and Vallerand, R. J. (2003) “The coach-athlete relationship: a motivational model”, *Journal of Sports Sciences,* Vol. 21, Issue 11.

K. Mannaro, M. Melis, and M. Marchesi, “Empirical analysis on the satisfaction of IT Employees comparing XP practices with other software development methodologies,” in *Extreme Programming and Agile Processes in Software Engineering, Proceedings, Lecture Notesin Computer Science*, vol. 3092: Springer Verlag, 2004, pp. 166-174.

Martin, Robert C. (2002)*, Agile Software Development: Principles, Patterns, and Practices,*Prentice-Hall, New Jersey.

McCoy, D. W., and Plummer, D. C. (2006). “Defining, Cultivating and Measuring Enterprise Agility”, *Gartner Research* (April 28, 2006).

McDonald, J. (2001), "Why Is Software Project Management Difficult? And What That Implies for Teaching Software Project Management", *Computer Science Education*, 11, 1, 55-71.

McMurtrey, Mark E., Grover, Varun, Teng, James T. C. and Lightner, Nancy J., “Job Satisfaction of Information Technology Workers: The Impact of Career Orientation and Task Automation in a CASE Environment,” *Journal of MIS*, Vol. 19, No. 2, 2002, pp. 273-302.

Moore, Gary C. and Benbasat, Izak (1991), “Development of an Instrument to Measure the Perceptions of Adopting an Information Technology Innovation”, *Information Systems Research*, 2, 3, pp. 192-222.

Mowday R.T., Steers R.M., Porter. (1979).”The measurement of organizational commitment”. *Journal of Vocational Behavior*, *14*, 224–247.

Mulaik, S. A. (1990). “Blurring the Distinctions between Component Analysis and Common Factor-Analysis. *Multivariate Behavioral Research*, 25(1), 53-59.

Mulhotra, Yogesh (2004), “Desperately Seeking Self-Determination: Key to the New Enterprise Logic of Customer Relationships”, *Proceedings of the Americas Conference on Information Systems, New York, New York, August 2004.*

Mulhotra, Yogesh and Galletta, Dennis (2005), “A Multidimensional Commitment Model of Volitional Systems Adoption and Usage Behavior”, *Journal of Management Information Systems*, 22(1), pp. 117-151.

Nagel, R. and Dove, R. (1991) *21st Century Manufacturing, Enterprise Strategy* (Iacocca Institute, Lehigh University, Bethlehem, PA).

Narasimhan, R. and Das, A. (1999), “An Empirical Investigation of the Contribution of Strategic Sourcing to Manufacturing Flexibilities and Performance, *Decision Sciences*, 30, 3, 683-718.

Narasimhan, R., Jayaram, J., and Carater, J. R. (2001), “An Empirical Examination Of the Underlying Dimensions of Purchasing Competence”, *Production And Operations Management*, 10, 1, 1-15.

Niemiec, Christopher P., Lynch, Martin F., Vansteenkiste, Maarten, Bernstein, Jessey, Deci, Edward L. and Ryan, Richard M. (2006). “The antecedents and consequences of autonomous self-regulation for college: A self-determination theory perspective on socialization”, *Journal of Adolescence,* 29, 761–775.

Nerur, S., Mahapatra, R.K., Mangalaraj, G. “Challenges of Migrating to Agile Methodologies”, *Communications of the ACM,* Vol. 48, No. 5, 2005, Pages 72-78.

Nosek, J.T. "The Case for Collaborative Programming," *Communications of the ACM,* Vol. 41, No. 3, 1998, pp. 105-108.

Novak, S. and Eppinger, S. D., (2001), “Sourcing by Design: Product Complexity and The Supply Chain, *Management Science*, 47, 1, 189-204.

Nunnaly, J. (1978). Psychometric theory. New York: McGraw-Hill.

O’Hara, Margaret T., Wilson, Richard T., Kavan, C. Bruce. (1999), “Managing The Three Levels of Change”, *Information Systems Management,* Vol. 16, Issue 3, p. 63 – 70.

Ong, Chorng-Shyong, Lai, Jung-Yu. (2004), “Measuring user satisfaction with Knowledge management systems: scale development, purification, and Initial test”, *Computers in Human Behavior,* (23), pp. 1329-1346.

Parsons, D., Ryu, H. and Lal, R. 2007. “The Impact of Methods and Techniques on outcomes from Agile Software Development Projects”, *IFIP 8.6 Conference: Organisational Dynamics of Technology-based Innovation: Diversifying the Research Agenda, McMaster, Wastell, Ferneley and DeGross (eds.) Manchester UK, June 14-16 2007, Springer, 235-249.*

Pavlou, Paul A. and El Sawy, Omar A. (2006), "From IT Leveraging Competence to Competitive Advantage in Turbulent Environments: The Case of New Product Development," *Information Systems Research,* 17(3), pp. 198-227.

Pelletier, L.G., Fortier, M.S., Vallerand, R.J., and Brière, N. M. (2001). "Associations among perceived autonomy support, forms of self-regulation, and persistence: A prospective study", *Motivation and Emotion, 25*, 279-306.

Pett, Marjorie, A., Lackey, Nancy R. and Sullivan, John J. (2003), *Making Sense of Factor Analysis,* London: Sage Publications.

Petter, Stacie, Straub, Detmar, Rai, Arun. (2007), "Specifying Formative Constructs In Information Systems Research", *MIS Quarterly,* Vol. 31, No. 4, pp. 623-656.

Podsakoff, P. M., MacKenzie, S. B., Lee, J. Y., and Podsakoff, N. P. (2003). Common Method Biases in Behavioral Research: A Critical Review of the Literature and Recommended Remedies. *Journal of Applied Psychology*, *88*(5), 879-903.

Prater, Edmund, Biehl, Markus and Smith, Michael Alan. (2001), "International Supply Chain Agility: Tradeoffs Between Flexibility and Uncertainty," *International Journal of Operations and Production Management,* Vol. 21, No. 5/6, pp. 823-839.

Reis, H. T., Sheldon, K. M., Gable, S. L., Roscoe, J., & Ryan, R. M. (2000). Daily well-being: The role of autonomy, competence, and relatedness. *Personality and Social Psychology Bulletin*, *26*, 419-435.

Riemenschneider, C., Hardgrave, B.C. and Davis, D.D. (2002) "Explaining Software Developer Acceptance of Methodologies: A Comparison of Five

Theoretical Models", *IEEE Transactions on Software Engineering*, 28, 12, pp. 1135-1145.

Rogers, Everett (2003, 1995, 1983, 1971, 1962), *Diffusion of Innovations, Fifth Edition*, Free Press, New York.

Ryan, Richard M. and Deci, Edward L. (2000), "Self-Determination Theory and the Facilitation of Intrinsic Motivation, Social Development and Well-Being", *American Psychologist*, 55(1), pp. 68-78.

Sambamurthy, V., Bharadwaj, Anandhi and Grover, Varun, (2003), "Shaping Agility Through Digital Options: Reconceptualizing the Role of Information Technology in Contemporary Firms," *MIS Quarterly, Vol. 27, No. 2, pp237-263.*

Sethi, A. K. and Sethi, S. P.(1990), "Flexibility in Manufacturing: A Survey", *The International Journal of Flexible Manufacturing Systems*, 2, 289-328.

Sharifi, H. and Zhang, Z. (1999), "A methodology for achieving agility in Manufacturing organizations: An introduction," *International Journal Of Production Economics*, 62, 7-22.

Shin, H., Collier, D. A., and Wilson, D. D., (2000), "Supply Management Orientation And Supplier/buyer Performance,"*Journal of Operations Management,* 18, 317-333.

Sircar, Sumit, Nerur, Sridhar P. and Mahapatra, Radhakanta (2001), "Revolution Or Evolution: A Comparison of Object-Oriented and Structured Systems Development Methods", *MIS Quarterly,* Vol. 25, Issue 4, pp. 457-472.

Slack, N. (1983), "Flexibility as a manufacturing objective, "*International*

Journal of Operations and Production Management, 3, 3, 4-13.

Slack, N. (1987), "The Flexibility of Manufacturing Systems", *International Journal of Operations and Production Management*,7, 7, 36-45.

Management: Supplier Performance and Firm Performance, *International Journal of Purchasing and Materials Management*, August 2-9.

Standage, Martyn, Duda, Joan L. and Ntoumanis, Nikos (2003), "A Model of Contextual Motivation in Physical Education: Using Constructs from Self-Determinaton and Achievement Goal Theories to Predict Physical Activity Intentions", *Journal of Educational Psychology*,Vol. 95, No. 1, pp. 97-110.

Standish Group. Chaos chronicles II. West Yarmouth, MA, 2001.

Somers, Toni, M., Nelson, Klara and Karimi, Jahangir, "Confirmatory Factor Analysis Of the End-User Computing Satisfaction Instrument: Replication within an ERP Domain", *Decision Sciences*, Vol. 34, No. 3, Summer, 2003.

Surry, D.W. and Farquhar, J.D. (1996). "Incorporating Social Factors Into Instructional Design Theory", In M. Bailey, and M. Jones, (Eds.) Work , Education, and Technology (pp. 6.1-6.8). DeKAlb, IL: LEPS Press.

Swafford, Patricia M. Ph.D. Thesis, Georgia Institute of Technology, April, 2003.

Tan, K, Kannan, V. R., and Handfield, R. B. (1996), "Supply Chain

Teng, James T.C., Grover, Varun and Guttler, Wolfgang (2002), "Information Technology Innovations: General Diffusion Patterns and Its Relationships to Innovation Characteristics", *IEEE Transactions on Engineering Management*, Vol. 49, No. 1, February 2002, pp 13-27.

Thornke, S. H. (1997), "The Role of Flexibility in the Development of New Products: An empirical study", *Research Policy,* 26, 105-119.

Tiwana, Amrit and Keil, Mark (2004), "The One-Minute Risk Assessment Tool", *Communications of the ACM,* (47)11.

Toleman, Mark, Ally, Mustafa and Daroch, Fiona (2004), "Aligning Adoption Theory with Agile System Development Methodologies", (working paper, Internet address: www.sci.usq.edu.au/staff/markt/papers/PACIS2004.pdf.

Tornatzky, L.G. and Klein, K.J., "Innovation Characteristics and Innovation Adoption-Implementation: A Meta-Analysis of Findings," *IEEE Transactions on Engineering Management*, Vol. EM-29, pp. 28-45, February, 1982.

Truex, D. P., Baskerville, R., and Klein, H., "Growing Systems in Emergent Organizations", *Communications of the ACM* (42:8), 1999, pp. 117-123.

Upchurch, Richard and Laurie Williams. "Extreme Programming for Software Engineering Education?," 31st ASEE/IEEE Frontiers in Education Conference 2001. ," Downloadable from http://collaboration.csc.ncsu.edu/laurie/.

Upton, D. M. (1004), "The management of manufacturing flexibility", *California Management Review,* Winter, 72-89.

Vallerand, R.J. (1997). "Toward a Hierarchical Model of Intrinsic and Extrinsic Movivation", *Advances in Experimental Social Psychology, 29, 271-360.*

Vallerand, Robert J. (2000). "Deci and Ryan's Self-Determination Theory: A View from the Hierarchical Model of Intrinsic and Extrinsic Motivation," *Psychological Inquiry,* 11(4), 2000, pp. 312-318.

Vallerand, R.J. and Reid, G., (1984) "On the causal effects of perceived competence on intrinsic motivation: A test of cognitive evaluation theory", *Journal of Sports Psychology,6, 94-102.*

Vallerand, R. J., Gauvin, L. I., & Halliwel, W. R. (1986). Effects of zero-sum competition on children's intrinsic motivation and perceived competence. *The Journal of Social Psychology, 126,* 465-472.

Van Oyen, Mark P., Gel, Esma G.S., and Hopp, Wallace J. (2001), "Performance Opportunity for Workforce Agility in Collaborative and Noncollaborative Work Systems", *IIE Transactions,* 33, 761-777.

Venkatesh, Viswanath and Davis, Fred D. (2000). "A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies", *Management Science*, Vol. 46, No. 2, pp. 186-204.

Venkatesh, Viswanath, Morris, Michael G., Davis, Gordon B. and Davis, Fred D. (2003). "User Acceptance of Information Technology: Toward a Unified View", MIS Quarterly, September, 2003, 27(3) pp. 425-478.

Vokurka, R. J. and O'Leary-Kelly, S. W. (2000), "A review of empirical Research on manufacturing flexibility," *Journal of Operations Management,* 18, 485-501.

Weidong, Xia and Lee,Gwanhoo (2005). "Complexity of Information Systems Development Projects: Conceptualization and Measurement Development," *Journal of Management Information Systems,* 22(1) pp. 45-83.

Weier, Mary Hayes, (2008). "SAP Software A "Complete Failure," Lawsuit

Claims”, *InformationWeek*, March 27, 2008.

Whitehead, J. R. and Corbin, C. B. (1991). “Youth Fitness Testing: The effect of percentile-based evaluative feedback on intrinsic motivation, *Research Quarterly for Exercise and Sport*, 62, 225-231.

Williams, G. C., McGregor, H. A. Sharp, D., Levesque, C., Kouides, R. W., Ryan, R. M., & Deci, E. L. (2006). Testing a self-determination theory intervention for motivating tobacco cessation: Supporting autonomy and competence in a clinical trial. *Health Psychology, 25,* 91-101.

Williams, Laurie Ann. "The Collaborative Software Process." Ph.D. diss., The University of Utah, 2000.

Williams, Laurie, Kessler, Robert R., Cunningham, Ward and Jeffries, Ron. (2000), “Strengthening the Case for Pair Programming”, *IEEE Software*, July/August, 2000, 19-25.

Wixom, Barbara H. and Todd, Peter A., (2005), “A Theoretical Integration of User Satisfaction and Technology Acceptance”, *Information Systems Research*, Vol. 16, No. 1, pp. 83-102.

Yusef, Y. Y., Sarhadi, M., and Gunasekaran, A. (1999), “Agile manufacturing: The Drivers, concepts and attributes”, *International Journal of Production Economics*, 62, 33-43.

Zawacki, Robert A., Norman, Carol A., Zawacki, Paul A. and Applegate, Paul D, (1995), *Transforming the Mature Information Technology Organization: Reenergizing and Motivating People*, EagleStar Publishing, Colorado Springs.

Zhang, Zhengwen and Sharifi, Hossein., “Towards Theory Building in Agile Manufacturing Strategy—A Taxonomical Approach”, *IEEE Transctions On Engineering Management*, Vol. 54, No. 2, May, 2007.

Zmud, Robert W. “Diffusion of Modern Software Practices: Influence of Centralization and Formalization.”, *Management Science*, Vol. 28, No. 12, December, 1982, pp. 1421-1431.

Zmud, Robert W., “The Effectiveness of External Information Channels in Facilitating Innovation Within Software Development Groups”, *MIS Quarterly,* June, 1983, pp. 43 – 58.

Zmud, Robert W. “An Examination of the “Push-Pull” Theory Applied to Process Innovation in Knowledge Work.”, *Management Science*, Vol. 30, No. 6, June, 1984, pp. 727-738.

Zmud, Robert W. “Individual Differences and MIS Success: A Review of the Empirical Literature”, *Management Science*, Vol. 25, No. 10, October, 1979, pp. 966-979.

Zmud, Robert W., “Management of Large Software Development Efforts”, *MIS Quarterly,* June, 1980, pp. 45 – 55.

BIOGRAPHICAL INFORMATION

Nancy Bonner has a Master of Science degree in Information Systems. She worked in the software development industry for over 10 years prior to pursuing a doctorate in Information Systems, with a minor in Psychology. Over the course of her career in the software development industry she held various positions including Programmer/Analyst, Programming Manager and Project Development Manager working on large-scale development projects. Her current research and teaching interests include software development and implementation, leadership/project management and knowledge management. She is a member of Beta Gamma Sigma, Upsilon Pi Epsilon, ACM, AIS and the IEEE Computer Society. Her goal is to obtain a faculty position where she can teach and pursue her research interests.