Communications of the Association for Information Systems

Volume 39 Article 14

9-2016

An Empirical Examination of an Agile Contingent Project/Method Fit Model

Diana K. Young
Trinity University, dyoung1@trinity.edu

Nicole L. Beebe
University of Texas at San Antonio

Glenn Dietrich
University of Texas at San Antonio

Charles Zhechao Liu
University of Texas at San Antonio

Communications of the Association for Information Systems

Research Paper ISSN: 1529-3181

An Empirical Examination of an Agile Contingent Project/Method Fit Model

Diana K. Young
Finance and Decision Science
Trinity University
dyoung1@trinity.edu

Nicole L. Beebe
Information Systems and Cyber Security
University of Texas at San Antonio

Glenn Dietrich
Information Systems and Cyber Security
University of Texas at San Antonio

Charles Zhechao Liu
Information Systems and Cyber Security
University of Texas at San Antonio

Abstract:

While research has demonstrated positive productivity and quality gains from using agile software development methods (SDMs), some experts argue that no single SDM suits every project context. We lack empirical evidence about the project contextual factors that influence when one should use these methods. Research suggests several factors to explain agile method appropriateness; however, generalizable empirical evidence supporting these suggestions is weak. To address this need, we used contingency theory and the information processing model to develop the agile contingent project/method fit model. Subsequently, we used the model to analyze the influence of project contextual factors and agile practices on software development professionals' perceptions regarding agile SDM appropriateness. We tested the model using survey data collected from 122 systems development professionals who provided information regarding: 1) contextual factors surrounding a recent agile development project, 2) agile practices applied during the course of that project, and 3) perceptions regarding the relative fit (appropriateness) of the agile method used. Linear regression identified several significant relationships between project contextual factors, agile practices, and respondents' relative fit perceptions.

Keywords: Software Development, Systems Development Methods, Agile Methods, Agility, Contingency Theory, Information Processing Model, Agile Practices.

This manuscript underwent peer review. It was received 10/27/2015 and was with the authors for 3 months for 2 revisions. Jeremy Rose served as Associate Editor.

1 Introduction

Maruping, Venkatesh, and Agarwal (2009) describe information systems development as “an inherently complex activity that is embedded with interdependencies [and] requires the collective input of multiple individuals with non-overlapping knowledge sets” (p. 377). This characterization underscores the multifaceted, complicated nature of the development process. Accordingly, management of development efforts necessitate the use of coordination, collaboration, and control structures to ensure that projects align with organizational objectives. Development teams often apply systems development methods (SDMs) to provide these structures.

Presently, developers use two types of SDMs widely in practice: plan driven and agile. Plan-driven SDMs are based on the waterfall model (Royce, 1987) of systems development, which deconstructs the project process into a series of linear phases. In contrast, agile SDMs are based on an incremental, iterative, lightweight approach that focuses on teamwork and face-to-face collaboration over formal processes (Baskerville & Pries-Heje, 2004; Kautz, Madsen, & Nørbjerg, 2007). Researchers have made attempts to identify contextual factors that indicate when one should use agile SDMs instead of plan-driven SDMs, but the generalizable evidence supporting the efficacy of specific factors is weak (Batra, Weidong Xia, VanderMeer, & Dutta, 2010; Cao, Mohan, Xu, & Ramesh, 2004; Harris, Collins, & Hevner, 2009).

Even with this weak generalizable evidence, the adoption of agile SDMs continues to grow. A recent survey of 3,925 software development professionals found that 94 percent work for organizations that practice some form of agile development (VersionOne.com, 2015). Further, 45 percent report that, in their organizations, development teams developed the majority of projects using agile SDMs—a significant increase over the results of a similar survey conducted six years earlier in which 31 percent reported that only one or two development teams in their organization used an agile SDM (VersionOne.com, 2009).

Researchers have made several recommendation to help practitioners choose which agile methods to use for their projects (Barlow et al., 2011; Batra et al., 2010; Boehm & Turner, 2005, 2003; Cao et al., 2004; Cockburn, 2002). However, evidence supporting these recommendations is largely anecdotal, qualitative, and/or case based (noted exceptions include Chow & Cao, 2008; Lee & Xia, 2010; Maruping et al., 2009). Additionally, research concerning agile team co-location has produced contradictory results (Batra et al., 2010; Sarker & Sarker, 2009). Accordingly, we argue that we need further research to quantitatively validate recommendations previously proposed in the literature and resolve contradictions.

Researchers have suggested but not widely investigated the notion that organizations should use different SDMs for different project contexts (Barlow et al., 2011; Boehm & Turner, 2003; Ratbe, King, & Young-Gul, 1999). Austin and Devin (2009) support a contingent perspective of agile SDM use and call on researchers to develop a “robust contingency framework for deciding when (in what conditions) plan-based and agile methods should be used” (p. 462). As such, we address that call to action and empirically assess whether or not specific contextual factors influence software development professionals’ perceptions regarding the fit (contextual appropriateness) of agile SDMs over plan-driven SDMs.

A second issue concerns the manner in which development teams apply agile SDMs in practice. Research has shown that use of agile methods varies widely. Some teams adopt only a handful of a SDM’s prescribed practices, while other teams adopt most or all practices (Wang, Conboy, & Pikkarainen, 2012). Additionally, some teams combine practices of different SDMs to form hybrid implementations (Baird & Riggins, 2012). Some experts argue against partial and hybrid implementations (Paredes, 2015), while a small body of qualitative evidence indicates that such approaches can be quite effective in certain contexts (Batra et al., 2010; Berger & Beynon-Davies, 2009). Accordingly, we also address this conflict in the literature and empirically assess whether or not the use of more agile practices is associated with improved perceptions regarding the fit of agile SDMs.

To summarize, we address two research questions (RQs):

RQ1: Do specific project contextual factors influence software development professionals’ agile fit perceptions?

RQ2: Does the number of agile practices used on a project influence software development professionals’ agile fit perceptions?

To answer these questions, we developed and administered an electronic survey instrument to a sample of agile software development professionals. We asked respondents questions regarding the contextual factors of an agile project they had recently worked on, the agile practices implemented on that project, their perceptions regarding the fit of the agile SDM used, and the fit they believe could have occurred if the project had used a plan-driven SDM instead. The contextual factors assessed included requirements uncertainty, user representative involvement, team size, development team expertise, and team location distribution. We then used linear regression to analyze the influence that the contextual factors and agile practices had on agile fit perceptions.

Our results fill an existing gap in the SDM literature by providing generalizable, quantitative evidence concerning project contextual factors that significantly influence software development professionals' perceptions regarding agile SDM fit. Further, the findings help move the IS discipline toward a contingent perspective of agile SDM use. We collected the study data from a moderately large, highly heterogeneous sample of software development professionals who represent a broad range of industries, organizational sizes, and software project contexts. We believe our contribution is important since prior investigations regarding agile SDM selection have largely relied on case-based and/or qualitative research methods.

Additionally, we based our model on a strong theoretical foundation, which we believe is an important contribution because much research has noted that many agile studies lack a theoretical basis (Abrahamsson, Salo, Ronkainen, & Warsta, 2002; Wang et al., 2012). Both contingency theory and the information processing model are well-established, highly regarded organizational principles that researchers have applied in the systems development context. By focusing on the propositions that these theories provide, we bring together several contextual factors discussed in the prior literature and test their influence on SDM fit perceptions in a holistic and rigorous manner.

Methodologically, with this study, we contributes to the literature by developing and testing measures for several new constructs. Specifically, we introduced a new ordinal measure to quantitatively assess a project's team location distribution. Additionally, we developed and used a measure to assess agile bias to control against respondents' preferences for agile methods. Finally, we introduced, developed, and validated a new relative fit construct to assess respondents' agile fit perceptions in relation to the fit they believe could have occurred had they used a plan-based method. These measures contribute to the literature and will help future research aimed at quantitative assessment in the agile and systems-development domains.

Finally, we help software development practitioners when making method selection decisions because we provide quantitative, empirical evidence derived from a moderately large, highly heterogeneous sample of software development professionals regarding the project contextual factors that they can use to contingently determine when using an agile SDM will likely be most effective.

2 Literature Review

In Section 2.1, we review the literature regarding SDM use. In Sections 2.2 and 2.3, we discuss the foundational theories used to develop our research model. In Section 2.4, we present the project contextual factors discussed in the literature in relation to agile SDM usefulness and outline our process for selecting the contextual factors included in the study.

2.1 Systems Development Method Use

Both plan-driven and agile SDMs are widely used in practice. Plan-driven methods rely heavily on up-front planning and documentation to develop a formal, detailed, complete specification of the target system before constructing that system. Proponents argue that this approach results in enhanced communication, reduced project risks, and better control of the development process (Fitzgerald, 1996). Opponents note that the approach's sequential nature severely impedes a development team's ability to respond to changing requirements (Baskerville & Pries-Heje, 2004).

In contrast, agile SDMs focus heavily on teamwork and collaboration over formal processes, production of working software over documentation, and adaption to change over adherence to a formal plan (Beck et al., 2001b). Unlike plan-driven methods, agile methods do not prescribe a specific series of tasks for completing a project. Collectively, the assigned team has discretion in designing and executing the development process. The team can perform requirements gathering, design, development, and testing sequentially, concurrently, and/or iteratively. As a result, agile methods strongly encourage collaboration and face-to-face communication over pre-defined processes so that the interactive, iterative process of continuously constructing, demonstrating, and revising system components replaces the need for copious documentation.

Use of SDMs varies widely in practice (Fitzgerald, 1997), and teams sometimes combine the components of different SDMs to form hybrid method implementations (Baird et al., 2012). Research has further shown that systems development professionals judge agile SDM usage dichotomously (either used or not used), when, in reality, individual implementations often only partially (and sometimes minimally) apply a prescribed set of agile practices (Wang et al., 2012). Further, individuals often describe development projects as using an agile SDM when, in reality, the method applied is a hybrid implementation. As a result, one can find it difficult to identify agile SDM use because significant differences exist between the practices prescribed by a normative SDM and the way that method is enacted in practice.

These challenges indicate that the phrase “agile SDM use” is subjective. As such, we believe that a scalar approach is best for assessing the construct, which is in line with research that indicates that different teams use different numbers of and combinations of agile practices (VersionOne.com, 2015). Accordingly, in our study, we deemed a project to have used an agile SDM if 1) the respondent believed the development team developed the project using an agile SDM and 2) the development team used at least one standard agile activity to complete the project. We see projects that used many agile practices as being more agile than projects that used fewer practices.

Some experts argue that no single SDM suits all development contexts (Boehm & Turner, 2003; Norbjerg, 2002). Researchers have suggested but not widely investigated the notion that organizations should use different SDMs for different project contexts (Barlow et al., 2011; Boehm & Turner, 2003; Ratbe et al., 1999). The Gartner Group recommends including an agile SDM in an organization’s *portfolio* of SDMs (Paredes, 2015). The notion of an organization’s having a *portfolio* of methods supports a contingent perspective of agile SDM usage, but we still lack empirical evidence regarding the specific contextual factors that should drive agile SDM usage.

2.2 Contingency Theory

Contingency theory originated in the organizational theory literature and proposes that organizations need to adapt their structure, processes, and coordination strategies to fit the degree of uncertainty in their environment (Burns & Stalker, 1961; Lawrend & Lorsch, 1986). Uncertainty stems from contextual factors such as culture, environment, technology, size, and knowledge. Specifically, contingency theory suggests that, when uncertainty is high, one needs more organic coordination strategies and processes to choose appropriate courses of action. Conversely, when uncertainty is low, one needs more formalized, rigid processes and coordination strategies. While early contingency theory discussions focused on structure at the firm level, numerous studies have applied the theory at the subunit or work group level (Andres & Zmud, 2001; Burns & Stalker, 1961; Drazin & Van de Ven, 1985).

Central to contingency theory is the concept of fit, which Lyytinen and Mathiassen (1998) define as a heuristic that describes the matching of structure and processes to contextual factors to reduce risk. Drazin and Van de Ven (1985) note that the extant literature has offered three different conceptualizations of fit: the selection view, the interaction view, and the systems view. The selection view describes fit as congruence between context and structure. The interaction view goes a step further and describes fit as an interaction between pairs of contextual and structural factors. The systems view takes the most complex approach and describes fit as an optimal state of consistency achieved when multiple contextual factors are matched with multiple structural factors to reduce uncertainty. All views believe that better fit enables better performance, and several studies have substantiated this relationship in the systems development context (Andres & Zmud, 2001; Barki, Rivard, & Talbot, 2001; Kyu Kim & Umanath, 1992; Lyytinen & Mathiassen, 1998).

Systems development most closely aligns with the systems view of fit because system development projects feature multiple contextual factors that influence the level of uncertainty associated with them. Similarly, SDMs often incorporate multiple coordination mechanisms, which one can view as multiple structural factors. Accordingly, in the systems development context, we argue that one achieves fit when one uses an SDM containing a specific set of coordination structures on a project characterized by contextual factors that require that level of structural support. One challenge in adopting the systems view of fit in this study is that identifying an optimal coordination strategy (SDM) can be difficult when multiple competing contingency factors and structures must be simultaneously addressed (Gresov, 1989).

Many prior studies have used contingency theory to study the notion of “fitting” software development practices to project context (Andres & Zmud, 2001; Austin & Devin, 2009; Barki, Rivard, & Talbot, 2001; Franz, 1985; Kyu Kim & Umanath, 1992; Mathiassen, Saarinen, Tuunanen, & Rossi, 2007). Barki et al.

(2001) used the theory to study the influence of fit between project risk levels and project management practices on software development performance. Results suggest that the project management practices one uses need to vary based on the level of risks emanating from a project's complexity, rate of change, availability of information, and clarity of information. Additionally, Barki et al. found that fit significantly influenced performance. McKeen, Guimaraes, and Wetherbe (1994) used contingency theory to study the influence of fit resulting from systems development complexity, user influence, user-developer communications styles, and user representative involvement on user satisfaction. Results indicate that the fit between project factors and user representative involvement does influence user satisfaction. Andres and Zmud (2001) used contingency theory to study the use of software development coordination strategies (not methods) based on the degree of uncertainty, task interdependence, and goal conflicts involved. Results show that fit between context and coordination strategies significantly influence developers' productivity and satisfaction. However, in reviewing the literature, we did not find any studies that used contingency theory to address the question of when agile SDMs best fit project contexts.

We use contingency theory as a theoretical basis for our research model because the theory's idea of matching structure and processes to contextual factors is in line with our goal of matching SDMs to specific project contextual factors. In the software development context, SDMs represent coordination strategies that system development teams use to manage the uncertainty associated with systems development projects. Accordingly, system development teams should adopt the coordination strategy (i.e., SDM) that best fits the uncertainty resulting from project contextual factors.

2.3 Information Processing Model

The information processing model relates closely to contingency theory and proposes that optimal work group performance results when the coordination strategies that a group uses fit the information processing needs of the task the group seek to complete (Galbraith, 1977; McCann & Galbraith, 1981). Different coordination strategies provide different mechanisms for processing information (Andres & Zmud, 2001). Well-understood tasks have low information processing needs and can easily be pre-planned. Less-understood tasks have high information processing needs, which makes pre-planning difficult. Accordingly, well-understood tasks better suit formal, mechanistic coordination strategies, while less-understood tasks better suit flexible, organic coordination strategies that emphasize communication, collaboration, and liaison devices in order for the team to effectively process information (McCann & Galbraith, 1981).

Barki et al. (2001) combined the information processing model and contingency theory to develop a contingent model of software project risk management. The model proposes that the fit between project risk exposure and the strategies used to manage that risk influence project performance. Results indicate that varying risk-management strategies to fit the degree of risk exposure (and the corresponding information-processing needs) significantly influences both project quality and project costs. Andres and Zmud (2001) used the information processing model to study the influence of software development task interdependencies and goal conflict on information processing needs, the fit between those needs and the coordination strategies used, and the resulting impact on productivity and user satisfaction. Results indicate that a significant interaction between task interdependencies and coordination strategies influences productivity. Additionally, the researchers found that organic coordination strategies resulted in higher levels of productivity overall.

When building the research model, we looked to both contingency theory and the information processing model for theoretical support. Whereas contingency theory supports our argument for matching coordination strategies to project context based uncertainty, the information processing model explains why organic strategies work best in uncertain contexts and mechanistic strategies work best in more certain contexts. High levels of uncertainty require development teams to process more information before choosing a path forward. More organic coordination strategies (agile SDMs) allow development teams to adapt the project path based on the knowledge they gain from continually processing information. Conversely, with low levels of uncertainty, development teams do not need to process a great deal of information to determine their next steps. In those instances, more mechanistic SDMs (plan-driven) approaches are appropriate because the path forward is well defined.

2.4 Project Context Factors

In this study, we use the phrase "project context" to refer to the factors that characterize a development project's environment. Past research has discussed a variety of project contextual factors thought to influence agile method appropriateness, such as requirement uncertainty, user representative involvement, team size, team expertise, team location, complexity, criticality, organizational culture, and management style (Batra et al., 2010; Berger & Beynon-Davies, 2009; Boehm & Turner, 2005, 2003; Lindvall et al., 2004; Sarker & Sarker, 2009). However, generalizable evidence supporting the individual factors is rather weak. A significant body of anecdotal and qualitative evidence indicates that agile methods can be quite effective when the project scope is small, requirements are uncertain, and the development team is co-located (Cao, Mohan, Xu, & Balasubramanim, 2009; Maruping et al., 2009). Some qualitative evidence suggests that organizations may face difficulties when adopting agile methods for large, complex, distributed, and/or business-critical projects (Barlow et al., 2011; Berger & Beynon-Davies, 2009; Boehm & Turner, 2005), while other qualitative evidence suggests agile methods can be effective when used on large, distributed development efforts (Batra et al., 2010). Researchers have proposed theoretical frameworks to contingently choose between agile and plan-based SDMS (Barlow et al., 2011; Boehm & Turner, 2003; Ratbe et al., 1999), but the contextual factors included in the frameworks are inconsistent and evidence supporting their efficacy is weak and/or lacking.

We believe one reason for the lack of generalizable, quantitative evidence in the agile domain is data-collection difficulties that stem from companies' reticence to participate in large software development research efforts. Software development projects often involve tight deadlines and stressful situations (Chilton, Hardgrave, & Armstrong, 2005). Accordingly, managers are reluctant to participate in efforts that could distract developers from critical project tasks. For the current study, we approached numerous companies seeking permission to collect information from software development teams. One small company offered to allow informal interviews with a handful of agile developers, which would be insufficient to draw robust conclusions. Two companies stated that prior difficulties getting a non-disclosure agreement approved by legal counsel precluded their teams from participating. Four other companies, citing tight deadlines as the reason, declined our request. Several other companies did not respond at all to numerous participation requests.

Possibly as a result, past research has largely relied on case-based and qualitative research methods to study how contextual factors influence agile SDM fit. These methods are particularly well suited for exploring emerging or complex phenomena in a given context (Benbasat, Goldstein, & Mead, 1987; Eisenhardt & Graebner, 2007; Geambasu, Jianu, Jianu, & Gavrila, 2011; Yin, 2003), but one must take care when generalizing results to other contexts (Creswell, 2009). We believe that prior agile research offers important insights regarding specific contextual factors that influence agile SDM fit; however, we need empirical validation to test the efficacy of the suggested factors in broader contexts. Additionally, we believe we need to investigate the relative influence of individual factors to determine if certain factors are more important than others are when contingently deciding whether or not to use an agile SDM.

Due to the data-collection challenges noted above, we strove to develop a parsimonious model with broad explanatory value. Accordingly, we employed three criteria to select the contextual factors included in our research model. First, we looked for factors that had been widely discussed in the literature in relation to agile SDM use. Second, we focused on factors that we could reliably measure at the individual unit of analysis. Third, we looked for factors whose relationship with agile SDMs one could explain using contingency theory and the information processing model. Using these criteria, we chose to include requirements uncertainty, team size, user representative involvement, team development expertise, and team location distribution in our model.

3 Hypotheses Development

We incorporated the chosen contextual factors, along with the number of agile practices the development team used, into a holistic contingency framework to determine if the independent variables influence systems development professionals' agile SDM relative fit perceptions. Figure 1 presents the research model.

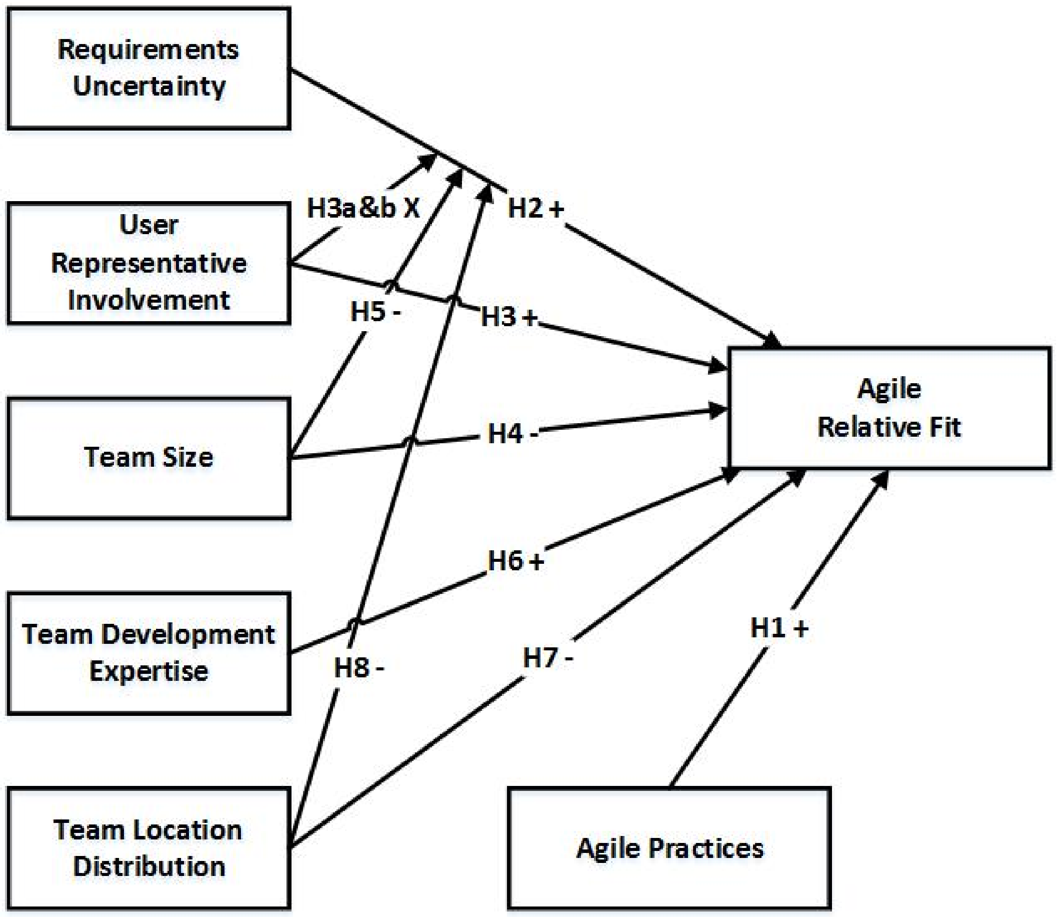

Figure 1. Agile Project/Method Fit Model

3.1 Agile SDMs

Agile SDMs are a set of software development methods that share a common collection of underlying values and guiding principles but differ in implementation practices. Extreme programming (XP) (Beck, 2000), scrum (Schwaber & Beedle, 2003), test-driven development (Beck, 2003), and kanban (Brechner, 2015) are four normative agile methods that have been widely adopted in practice. Each agile SDM comprises a bundle of prescribed practices and recommended processes for enacting those practices. Prescribed scrum practices include sprint planning, daily meetings, sprint reviews, burndown charts, and project retrospectives (Schwaber & Beedle, 2003). Extreme programming practices include release planning, daily meetings, acceptance testing, pair programming, and unit testing (Beck, 2000). Test-driven development focuses on test criteria development, unit testing, and acceptance testing. Kanban focuses on introducing incremental improvements in the development process through using signaling techniques that promote communication and collaboration in work units (Brechner, 2015). While terminology differences exists between agile methods, the practices that the different agile SDMs prescribe strongly overlap. For example, the scrum practices of sprint planning and daily meeting are similar to extreme programming's release planning and daily stand-up meetings.

MacCormack, Kemer, Cusuman, and Crandall (2003) studied how the use of the extreme programming practices of incremental development, prototypes, daily builds, and regression testing influenced team productivity and defect rates. Results indicate that the use of prototypes and regression testing influenced defect rates and that prototyping and daily builds influenced productivity when measured as new lines of code.

Parades (2015) argues that organizations should not pick and mix agile practices but fully adopt a normative SDM. However, a great deal of research shows that broad differences exist in the way teams enact agile SDMs in practice. Some teams adopt only a subset of prescribed practices (Baird & Riggins, 2012; Fitzgerald, 1997; VersionOne.com, 2015; Wang et al., 2012) while others combine practices from different SDMs to form a custom or hybrid method (Batra et al., 2010; Persson, Mathiassen, & Aaen, 2012; Theocharis, Kuhrmann, Munch, & Diebold, 2015). Accordingly, we take a scalar approach and use the surrogate measure of number of agile practices to represent the breadth of the agile SDM that a team used. Accordingly, in our model, a team that used more agile practices represents a *more agile* implementation, and a team that used fewer agile practices represents a *less agile* implementation. As such, we hypothesize:

Hypothesis 1: The number of agile practices used on a project positively influences software development professionals' agile relative SDM fit perceptions.

3.2 Requirements Uncertainty

Requirements uncertainty refers to the lack of clarity and stability of a project's functional requirements from project inception to completion. Several papers have discussed the relationship between requirements uncertainty and the use of agile SDMs. Boehm and Turner (2003) include requirements uncertainty in their theoretical framework for selecting the appropriate SDM for a project: they argue that projects with high levels of requirements uncertainty should adopt an agile SDM, projects with moderate levels of uncertainty should adopt a hybrid SDM, and projects with low levels of uncertainty should adopt a plan-driven SDM. Maruping et al. (2009) offer empirical evidence of a relationship between requirement uncertainty and agile SDM efficacy. Their results indicate that the three-way interactive effect of requirements change, agile method use, and use of management outcome based control modes relates to project quality.

High levels of requirements uncertainty make successfully completing a project more difficult because the team is unsure of the functionality that it needs to deliver. Accordingly, the development team must work with project stakeholders to gather and process information regarding those requirements. Agile SDMs specifically deal with unclear and/or changing requirements. The agile practices of user-created functional stories, iterative development, release planning, daily meetings, and early prototyping mitigate the difficulty generated by uncertain requirements by providing supportive coordination structures to progressively process information to work through requirements ambiguity. As such, we hypothesize:

Hypothesis 2: Requirements uncertainty positively influences software development professionals' agile SDM relative fit perceptions.

3.3 User Representative Involvement

User representative involvement refers to the degree that business representatives and/or actual end users actively participate in the development process. Chow and Cao (2008) report that, when a development team uses an agile SDM, customer representative involvement is highly correlated with the team's ability to fulfill users' needs and requirements. Harris et al. (2009) argue that one should use the combined effect of requirements uncertainty and user representative involvement to determine the most appropriate development method for a project. They theorize that projects with minimal up-front specifications need more frequent user feedback and that projects with more exact up-front specifications need less frequent user feedback. Boehm and Turner (2005) found anecdotal support for a relationship between customer representative involvement and the efficacy of agile methods, with systems development professionals citing lack of user involvement as a key inhibitor to agile methods' success.

We believe that user representative involvement influences agile fit in a multi-faceted manner. First, most systems development contexts require user representative involvement so the development team can gather information regarding the functional requirements of the system it needs to build. Accordingly, user representative involvement is integrally tied to a development teams' ability to gather and process information concerning what the system must do. The agile practices of user created stories, daily meetings, release planning, prototyping, and iteration demonstrations provide supportive coordination mechanisms for development team members to work effectively with user representatives to complete the project. As such, we hypothesize:

Hypothesis 3: User representative involvement positively influences software development professionals' agile SDM relative fit perceptions.

However, we also believe that user representative involvement and requirements uncertainty have an interactive effect on perceived agile fit. Harris et al. (2009) argue that one should consider the combined effect of project uncertainty and user representative involvement when determining the most appropriate

SDM for a project. They argue that projects with minimal or unclear up-front specifications need more frequent user feedback. Conversely, when requirements are more certain, high levels of user representative involvement are not necessary and could disrupt a development team's ability to develop the target system.

The information processing model suggests that, when uncertainty is high, one cannot pre-plan the path forward for a project. Project team members must collect and process information to understand the underlying cause(s) of the uncertainty and choose the next steps forward. In the systems development context, developers are often unfamiliar with the business rules and processes that motivate the functional requirements of the system they need to build and must rely on user representatives to share that information with them. Further, when those requirements are uncertain, developers must work closely with users to unravel the causes of the uncertainty and suggest alternative solutions to resolve the conflict. Accordingly, the information processing model suggests that high requirements uncertainty necessitates high user representative involvement. Conversely, when requirements are more certain, developers depend less on user representatives because they can pre-plan the steps forward. In those instances, high user representative involvement may prove distracting for the development team because the team may spend time interacting with users rather than developing components to fulfill the known requirements.

When reviewing the agile principles and practices in terms of requirements uncertainty and user representative involvement together, one can easily see that an agile team's ability to deal with uncertainty depends on user representatives' being readily available and involved in the development process. When that occurs, the agile process of repeatedly modifying and demonstrating a system module until user satisfaction is achieved provides a robust mechanism for dealing with uncertain requirements. However, we also believe that high user representative involvement when a team knows the requirements could disrupt its development efforts. As such, we hypothesize:

Hypothesis 3a: User representative involvement moderates the influence of requirements uncertainty on software development professionals' agile SDM relative fit perceptions such that fit is better when high requirements uncertainty coincides with high user representative involvement.

Hypothesis 3b: User representative involvement moderates the influence of requirements uncertainty on software development professionals' agile SDM relative fit perceptions such that fit is better when low requirements uncertainty coincides with low user representative involvement.

3.4 Team Size

Team size refers to the number of individuals involved in a development project. Several researchers have found empirical support for a negative relationship between team size and project success (Barki, Rivard, & Talbot, 1993; Cao et al. 2009; Kraut & Streeter, 1995). Barki et al. (1993) found that team size significantly influenced the level of risk that systems development professionals associated with projects. The professionals perceived projects characterized by large team sizes as being much riskier efforts than projects characterized by small team sizes. More recently, research has noted team size as an important factor to consider when determining the most appropriate SDM for a development effort (Barlow et al., 2011; Boehm & Turner, 2003; Cao et al., 2009; Kraut & Streeter, 1995).

Kraut and Streeter (1995) argue that team size influences the extent to which project teams require use of formal communication methods and the level of effectiveness that project members and managers associate with those communication methods. Boehm and Turner (2003) include the number of people involved in a project as one of the five factors they argue that one should consider when deciding whether to use an agile, plan-driven, or hybrid SDM. They suggest that small teams should use an agile SDM method and that large teams should use a plan-driven approach. Barlow et al. (2011) also argue that agile methods do not suit large projects. They note that flexible approaches rely heavily on ad-hoc, informal communication to coordinate project work and the effectiveness of those communication methods rapidly degrades as a project's team size increases. Accordingly, we believe that a plan-driven or hybrid SDM best suits large teams and that agile SDMs best suit small teams.

In terms of the information processing model (McCann & Galbraith, 1981), we believe that, as a project's team size increases, the effectiveness of agile SDMs' informal communications mechanisms to disseminate information across the team decreases (Kraut & Streeter, 1995). With projects involving a large number of developers and many business units, face-to-face collaboration among all of the project participants would be difficult to coordinate and necessitating the use of more formal information-processing mechanisms. Additionally, achieving consensus among the many business participants concerning the requirements the developers need to address in each development iteration could be both time-consuming and complicated. Finally, when building information systems, all of the developers' individual efforts should fit together to form the target deliverable. Large development teams can best achieve this goal with design documents and diagrams that depict how the individual parts of the system fit together. However, developing these documents and diagrams runs counter to the agile principle of allowing the architecture and designs to emerge from the development team and not be imposed up-front. Accordingly, we believe that large teams will view agile SDMs as lacking effective coordination and collaboration mechanisms. As such, we hypothesize:

Hypothesis 4: Project team size negatively influences software development professionals' agile SDM relative fit perceptions.

Additionally, we believe that an interactive effect involving requirements uncertainty and team size influences individuals' relative fit perceptions. Teams dealing with high requirements uncertainty must process large amounts of information to clarify a system's underlying functional and non-functional requirements (McCann & Galbraith, 1981). Agile SDMs largely focus on using face-to-face communications to complete the requirements elicitation process. However, large teams frequently need to use formal communications methods to disseminate information across the many members (Kraut & Streeter, 1995). Accordingly, we believe that, as a team's size increases, the effectiveness of face-to-face communication methods to work through uncertainty declines, which, in turn, reduces individuals' perceptions regarding the fit of the agile SDM to the given project context. As such, we hypothesize:

Hypothesis 5: Team size negatively moderates the influence of requirements uncertainty on software development professionals' agile SDM relative fit perceptions such that the positive impact of requirements uncertainty decreases as team size increases.

3.5 Team Development Expertise

Development expertise refers to a development team's skill level. Yang, Kang, and Mason (2008) found empirical support for a link between development team expertise and project performance. Several authors propose that development expertise influences a team's ability to effectively use an agile SDM, but supporting empirical evidence of this association is weak. In the theoretical vein, Cockburn (2002) categorizes systems developers according to their level of skill development and argues that these categories are an important factor to consider when staffing agile project teams. Boehm and Turner (2003) include team expertise as a factor in their method selection framework. Chow and Cao (2008) found that development professionals strongly associated the caliber and capability of a development team with the on-time and on-budget project delivery of agile project.

In terms of the information processing model, we believe that more expert development teams will be better at using agile SDMs' organic coordination strategies to elicit system requirements and develop system solutions because the team members' skills and experience provide a knowledge base for dealing with uncertainty (Yang, Kang, & Mason, 2008). Conversely, less skilled teams need the additional rigor of plan-driven SDMs to complete the development process because their structured approach provides them with a work plan to progressively identify requirements, design the system modules, and develop the target solution (McCann & Galbraith, 1981).

Agile practices are loosely framed activities to enable coordination and collaboration in agile teams. Because an individual's ability to enact these activities depends on the individual's systems development skill level, we believe that team development expertise is positively associated with agile SDM relative fit perceptions. Increased development skill levels result in improved task performance, which, in turn, results in improved overall developer satisfaction with the agile SDM used to complete the project. Additionally, we believe that highly skilled development teams view the reduced processing requirements of agile methods more positively because they allow those individuals to focus on what they are good at—building systems and not on writing documentation and fulfilling processes. As such, we hypothesize:

Hypothesis 6: Team development expertise positively influences software development professionals' agile SDM relative fit perceptions.

3.6 Team Location Distribution

Team location distribution refers to the physical distance that exists between project team members' workspaces. Most discussions of agile development suggest that agile SDMs best suit projects where team members share a common workspace based on the belief that placing an entire team in a common location facilitates face-to-face communication (Beck, 2000 Boehm et al., 2003; Ramesh, Cao, Mohan, & Peng, 2006; Sarker & Sarker, 2009). In contrast, distributed development projects involve teams that are geographically and/or temporally distributed. Researchers believe the resulting distance between team members makes reliance on informal communication methods impractical when they use agile SDMs (Ramesh et al., 2006; Sarker & Sarker, 2009).

Sarker and Sarker (2009) investigated the dimensions of agility in a large multinational high-tech organization whose systems development personnel were distributed across five cities on three continents. The authors conclude that strict adherence to agile SDMs did not prove effective for distributed development. However, Batra et al. (2010) present case evidence of a team that combined scrum with a plan-driven development approach to successfully complete a large, distributed development effort. Similarly, Persson et al. (2012) present case evidence of a successful distributed agile development effort that used technology-mediated communication tools to enhance team collaboration and project control.

When analyzing the agile principles in terms of team distribution and the information processing model, it appears that agile SDMs best suit co-located teams. Many of the agile practices, such as daily meetings and sprint retrospective, are based on team members' ability to interact face-to-face and respond to changes in a timely manner. While advanced information technologies can facilitate information processing in distributed teams, research has found both spatial and temporal distance to exacerbate team coordination problems (Batra et al. 2010; Sarker & Sarker, 2009). We believe that physical distance between agile team members creates an additional challenge to a development team's ability to process information using informal communication techniques. As such, we hypothesize:

Hypothesis 7: Team location distribution negatively influences software development professionals' agile SDM relative fit perceptions.

Additionally, we believe that team location distribution has an interactive influence with requirements uncertainty on agile fit perceptions. In projects with high levels of requirements uncertainty, team members must process a great deal of information to clarify the underlying functional and non-function requirements (McCann & Galbraith, 1981). Agile SDMs largely focus on using face-to-face communication to process that information. However, as spatial and temporal distance increases, teams face additional challenges when relying on agile SDMs to work through uncertain requirements (Ramesh et al., 2006; Sarker & Sarker, 2009). When team members are co-located, individual members can quickly and easily share important information throughout the workday. However, in distributed teams, individuals are faced with the additional challenge of either coordinating physical meetings or relying on other modes of communications to work through the underlying uncertainty. In extreme cases where team members are spread across countries and time zones, members may even need to shift work schedules so that they share work times and, thus, can discuss important information. As such, we hypothesize:

Hypothesis 8: Team location distribution negatively moderates the influence of requirements uncertainty on software development professionals' agile SDM relative fit perceptions such that the positive impact of requirements uncertainty decreases as team location distribution increases.

4 Research Method

To test the research model and its associated hypotheses, we employed a survey-based research method. In Sections 4.1 to 4.5, we outline the measures we used, the sample frame we included, the data-collection processes we employed, and the analyses we conducted to complete the study.

4.1 Survey Process

We developed a new electronic survey instrument for the study, which began with following priming statement and question:

Agile methods have been described as valuing:

- *teamwork and collaboration over formal processes*
- *production of working software over documentation, and*
- *adaption to change rather than following a plan.*

Have you ever worked on a software development project that used an agile method?

We instructed respondents who replied in the affirmative to think of an agile project that they had recently worked on and refer to that project when answering all subsequent questions. We then displayed a list of 16 agile practices and asked respondents to indicate which of those practices the referenced project used. Next, we asked respondents a series of questions concerning the contextual factors of the focal project and their perceptions regarding the fit of the agile SDM that their team used relative to the fit they believed would have occurred had their team used a plan-driven SDM instead. We thanked respondents who indicated that they had not worked on an agile project and removed them from the survey.

A panel of ten software development experts assessed the survey instrument for content validity, and the survey went through three iterations of pilot testing before we administered it. Where possible, we drew measures from the existing literature. For measures based on a Likert response format, we included a minimum of four items per construct to ensure that we could calculate a quasi-interval composite score using the individual items. Prior research has shown that such composite scores are appropriate for parametric analyst and robust to minor violations of the assumptions of linear regression (Boone & Boone, 2012; Carifio & Perla, 2007; Glass, Peckman, & Sanders, 1972). Additionally, many prior information systems studies have used such composite scores (Chin, 1998; Goel, Johnson, Junglas, & Ives, 2011; Venkatesh, Thong, & Xu, 2012). We randomized all measurement items and their corresponding response options in the instrument to protect against order effects (Shadish, Cook, & Campbell, 2002). Additionally, we reverse scored at least one item of each measure to protect against response pattern biases. We discuss the adapted and newly developed measures in Sections 4.2 and 4.3.

4.2 Independent Measures

The survey instrument assessed six independent variables: agile practices, requirements uncertainty, user representative involvement, team location distribution, team size, and team development expertise. In Sections 4.2.1 and 4.2.2, we outline the processes used to develop or adapt the measures for these variables.

4.2.1 Agile Practices

We could not find a list of standard agile practices in the literature. Accordingly, we developed a new measure. We began by synthesizing the lists of practices that scrum and extreme programming prescribe. To do so, we rephrased several practice names because terminology differences exist between the SDMs. We then showed the synthesized list to a focus group of four experienced agile developers who represented two companies and asked them to review it for accuracy and completeness. Together, the focus group members had experience using scrum, extreme programming, and kanban. The group suggested wording changes for four of the listed practices and recommended two additions. We adopted all of the focus group's suggestions and incorporated the updated list in the survey instrument. When completing the survey, we asked respondents to indicate all practices that their team enacted during the course of the focal project. Appendix A shows the agile practices included in the list and all other survey measures.

4.2.2 Project Contextual Factors

Maruping et al. (2009) employed three subjective items to assess requirements uncertainty using a seven-point Likert agreement response format. We made small adaptions to those items for our study and added one item, which we took from Ratbe et al. (1999), to enhance construct reliability. Chow and Cao (2008) assessed user representative involvement using a four-item, seven-point Likert agreement response format; however, they do not report an exact reliability statistic for the measure. Based on feedback from a panel of software development experts, we heavily modified the wording of these items for the current study. To assess team size, we asked respondents to specify how many total people the project team had for all iterations of development. Wording of the item included instructions to use decimals to indicate part-time assignments.

We could not locate an existing measure of location distribution in the literature. Accordingly, we developed a new ordinal measure, which assessed this construct using one dichotomous item and one multiple-choice item. The dichotomous item asked: “Were all members of the development team located in a common workspace?”. If the response to this question was negative, we then displayed a multiple-choice item and asked respondents to select the option that most closely described the location distribution of the team. Ordinal measure values ranged from zero, representing instances where the entire team was located in a common workspace, and five, representing instances where team members were located in different countries and time zones. Finally, we adopted Barki et al.’s (1993) four item, seven-point Likert response format measure of team development expertise. Based on feedback from a panel of software development experts, we slightly modified the wording of these items.

4.3 Dependent Measures

We included two measures, perceived agile fit and perceived structured fit, in the survey and used them to calculate the dependent variable tested during the analysis phase. We believe that project/method fit is most meaningful when viewed as a subjective, relative value concerning the appropriateness of one method versus an alternative. Software professionals may believe that a method worked well for a project but that an alternative would have worked better.

For this study, we were most interested in how project context influences developers’ perceptions regarding the fit of an agile method relative to a plan-driven method because they are the two main types of methods in use today. Accordingly, we first asked respondents to respond to a set of items that assessed how well they believed the agile method used fit the project context (perceived agile fit). We then displayed the following priming statement:

> *In contrast to agile methods, structured/waterfall methods rely on up-front planning and documentation to progressively develop a detailed specification of the target system before beginning construction. Typically, structured methods begin with a requirements gathering phase which is followed sequentially by design, construction, and implementation phases. Assume for a moment that a structured/waterfall rather than an agile method had been used for the project. Based on that assumption, indicate your level of agreement with the following statements.*

We then asked respondents to indicate how well they believed a plan-driven method would have fit the focal project context (perceived plan drive fit). To assess both constructs, we drew two Likert response format items from Kacmar, McManus, Duggan, Hale, and Hale (2009) and developed and added two new Likert response format items to enhance reliability.

Prior to analyzing the collected data, we averaged the Likert format responses for both perceived agile fit and perceived plan-driven fit. Research has shown that calculating a composite score for multiple Likert response format items yields a quasi-interval value that is appropriate for parametric analysis and is robust to minor violations of the assumptions of regression (Boone & Boone, 2012; Carifio & Perla, 2007; Glass et al., 1972). Next, we conducted a paired sample t-test to confirm that significant differences existed between the perceived agile fit and perceived plan-driven fit composite scores. We found significant differences in respondents’ perceptions regarding the fit of agile SDMs ($M = 5.77$, $SD = 1.13$) and plan-driven SDMs ($M = 2.93$, $SD = 1.47$) conditions ($t(120) = 14.86$, $\rho < 0.001$). Because we detected statistically significant differences, we subtracted perceived plan-driven fit from perceived agile fit to yield the dependent relative fit variable that we used for our regression analyses. Accordingly, relative fit is a continuous variable and assesses whether a respondent believed that the applied agile method was superior or inferior to a plan-driven alternative for the given project context. A large positive relative fit value represents high satisfaction with the agile method used relative to a plan-driven alternative, whereas a large negative relative fit value represents high dissatisfaction with the agile method used relative to a plan-driven alternative. Values close to zero represent cases where the respondent believes either method would have worked equally well.

4.4 Sample Frame Selection

The study population of interest was software development professionals who had experience using agile SDMs. We chose these individuals because they have actively used the focal SDMs and their work experience allows them to personally assess how well the SDM they used fit the contexts of the project in which they applied it.

To reach such a broad range of individuals, we included members of several software development professional organizations in the sample frame. One such organization is the Agile Alliance (AA), which provides online resources, coordinates conferences, and supports numerous local agile user groups worldwide. In addition to AA, we scanned the descriptions of all registered Linkedin.com and Yahoo.com groups for the terms: agile, scrum, XP, extreme programming, software, programming, software development, software engineering, Java, C, and .Net. For all matches, we sent a message to the group's administrator seeking permission to join the group and post discussion threads inviting members to participate in our research study. In total, 49 Linkedin.com and 18 Yahoo.com groups agreed to let us join and post participation solicitations.

One concern that arose when designing the study was that the sample frame may be biased toward agile methods. The simple act of joining a group shows an interest and possibly a preference for a group's focal topic. As such, we felt that members of the Agile Alliance and agile-specific groups would hold biases for agile SDMs over plan-driven SDMS. Accordingly, we employed two methods to control against such biases. First, 24 of the 67 (35%) Linkedin.com and Yahoo.com groups included in the sample were general software development groups and not agile-specific groups. Examples of general software development groups included computer and software professions groups (40,635 members) and the Java developers group (181,759 members). While the sample frame included more agile-specific groups in total than general software development groups, the total membership of the general software development groups (458,802 members) far exceeded the total membership of the agile-specific groups (111,642). Accordingly, we felt our sampling approach included strong potential for individuals with general software development interests, not just agile specific interests, to participate in the study.

To further control against agile biases, we also included an agile bias measure in the survey instrument. When completing the survey, we asked respondents to rate their level of agreement with the statement: "Agile methods are suitable for use in all project contexts" using a seven-point Likert scale (1 = strongly disagree; 7 = strongly agree). We incorporated this measure, along with years of systems development experience and years of agile experience, as control variables in the statistical model we used to analyze the collected data.

4.5 Data-collection Processes

We sent 170 email messages to all agile group leaders listed on the agilealliance.org website. The email messages contained: 1) a link to the electronic survey instrument, 2) a request that the message recipient complete the survey, and 3) a request that the message be forwarded to all members of the local user group that recipient leads. Eight emails failed to send due to inactive email addresses on the website. We sent follow-up messages to all active addresses two weeks after the original message, which thanked those who had already participated and reiterated the request that others take the time to participate and forward the message on to their local group members.

Additionally, we posted discussion threads announcing the study and inviting members to participate in the investigation on 49 Linkedin.com and 18 Yahoo.com group sites. While no method to assess the number of individuals who actually saw the discussion thread invitations exists, the combined membership of the 67 social media groups at the time of the survey was approximately 570,000 accounts. We posted follow-up discussion threads two weeks later reiterating our invitation to participate in the study. We finished collecting data two weeks after posting the follow-up discussion threads. We offered no incentives for study participation.

5 Results

In Sections 5.1 to 5.3, we discuss our sample demographic, instrument validity and reliability results, and multiple regression results.

5.1 Sample Characteristics

In total, we collected 195 responses. However, we had to drop 73 responses because the respondents failed to complete the entire survey instrument. While this completion rate is low, it is in line with the findings of other researchers who have employed electronic survey instruments to collect data from business professionals (Beebe, Young, & Chang, 2014; Roster, Rogers, Hozier, Baker, & Albaum, 2007; Roy & Berger, 2005).

The 122 usable responses represented individuals who possessed respectable levels of systems development experience and who worked in a wide variety of industries and organizational sizes. The average years of software development (SD) experience was 17.02 and the standard deviation was 9.61. Twenty-four percent of the respondents had between one and ten years of SD experience, 37 percent had between 11 and 20 years of SD experience, and 37 percent had over 20 years of SD experience. The average years of agile experience was 6.41 and the standard deviation was 6.01. Fifty-five percent of the respondents possessed between one and five years of agile experience, 34 percent possessed between six and ten years of agile experience, and seven percent possessed more than ten years of agile experience. While all respondents had worked on an agile development effort, 88 percent also had experience using plan-driven SDMs. Nearly 60 percent of the respondents worked in the information technology industry, while the remaining 40 percent were well distributed across the utilities, health care, financial services, education, and manufacturing industries. Eighteen percent of respondents worked for companies that employed 10,000 or more individuals, 20 percent worked for companies with between 1,000 and 10,000 employees, 30 percent worked for companies with between 100 and 1,000 employees, and 32 percent worked for companies with fewer than 100 employees.

The sample respondents also reported a wide range of team location distributions. Sixty-two percent reported that all development team members were co-located in the same work space, eight percent reported that team members were located in different work spaces in the same building, two percent reported that team members were located in different buildings in the same metropolitan area, four percent reported that team members were located in different metropolitan areas of the same country but in the same time zone, seven percent reported that team members were located in different time zones in the same country, and 17 percent reported that team members were located in different countries and different time zones.

In terms of agile practices, the mean number used was 10.25 and the standard deviation was 3.34. Only five respondents reported using fewer than five agile practices. Thirty-five respondents used between five and eight agile practices. Forty-five used between nine and twelve practices. Thirty-seven respondents used more than twelve practices.

5.2 Construct Validity, Reliability, and Descriptive Statistics

We used principal component analysis (PCA) with varimax rotation and Kaiser normalization to assess the convergent and discriminate validity of the multi-item constructs included in the study. Initial results revealed cross-loading problems with three items. Accordingly, we dropped those items and subsequent results showed that all items loaded significantly (> .705) on the theorized factors and all cross-loadings were less than .277. Table 3 shows the descriptive statics and correlations for each construct and the reliability coefficients (α) and square root of the AVEs for all multi-item measures (Nunnally, 1994; Gefen, Rigdon, & Straub, 2011). The reliability coefficients and square roots of the AVE for the agile fit and plan-driven fit composite values that we used to calculate the relative fit dependent variable were α = 0.89, SR of AVE = 0.88 and α = 0.88, SR of AVE = 0.86, respectively.

Table 1. Construct Reliability Coefficients, Descriptive Statistics, and Correlations

					Correlations						
Num	**Variable**	**α**	**Mean**	**St. dev.**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	ReqUnc	0.790	4.64	1.28	**0.83**						
2	UserInv	0.757	5.22	1.21	0.05	**0.81**					
3	TeamSize		13.11	12.14	0.04	-0.08	**NA**				
4	TeamLocDist		1.52	2.07	0.03	-0.07	0.25**	**NA**			
5	TeamDevExp	0.725	5.40	1.01	-0.07	0.39**	-0.04	-0.08	**0.75**		
6	AgilePractices		10.25	3.33	0.11	0.25**	0.06	-0.40	0.36**	**NA**	
7	RelativeFit		2.80	2.18	0.12	0.31**	0.03	0.04	0.47**	0.47**	**NA**

We used the unmeasured latent factor technique that Podsakoff, MAcKenzie, Lee, and Podsakoff (2003) outline to test for the presence of common method bias in the data. This technique calls for comparing the factor analysis results of the hypothesized model with the results of a revised model that includes a common latent construct where all items are modeled to load on both their hypothesized construct and a common latent construct. To conduct the test, we used AMOS to estimate a confirmatory factor analysis model for the hypothesized measurement model. We then added the common method construct and re-estimated the model. Finally, we compared the standardized regression weights and chi-squared values of the two models. We found no significant differences, which suggests that common method bias was not a significant concern.

5.3 Regression Analysis

We used hierarchical linear regression (Baron & Kenny, 1986) to test the model. We mean-centered all variables to reduce the possibility of multicollinearity (Aiken & West, 1991). We entered the control variables into the model first, followed by the main effect variables, and then the interaction terms. Table 2 presents regression results for the full sample. The block 1 column in the table presents the results with only the control and main effects variables included in the model. The block 2 column presents the results with the interaction terms added. Parentheses show variance inflation factors (VIFs), which were all well below the suggested threshold of 10 (Ryan, 1997), which indicates that multicollinearity was not a concern in the block 2 model.

Table 2. Relative Fit Regression Results

	Block 1	**Block 2**
Controls		
Years SD experience	-0.007	-0.004 (1.404)
Years agile experience	0.086**	0.087** (1.366)
Agile bias	0.190*	0.161+ (1.210)
Main effect variables		
Number of agile practices	0.170***	0.156** (1.138)
Requirements uncertainty (ReqUnc)	0.243*	0.243* (1.080)
User representative involvement (UserInv)	0.054	0.132 (1.276)
Team size	-0.001	-0.001 (1.096)
Team development expertise	0.681***	0.645*** (1.340)
Team location distribution (LocDist)	0.132+	0.112 (1.123)
Interaction Variable		
ReqUnc X UserInv		-0.302** (1.156)
ReqUnc X Team Size		-0.001 (1.246)
ReqUnc X LocDist		0.051 (1.262)
R^2	**0.427**	**0.469**
F	**16.133**	**18.892**
Adjusted R^2	**0.381**	**0.411**
ΔR^2		**0.042**

Notes: n = 122. Variance inflation factors are shown in parentheses.
+p<0.10, *p<0.05, **p<0.01, ***p<0.001

The block 2 model, which includes both the main effects and interaction terms, accounted for 46.9 percent of the variance detected in the data and offered a 0.042 improvement in explanatory power over the block 1 model. Accordingly, the full model provided the best explanation of the relationships present in the collected data ($F = 18.892$, $p < 0.001$). In the block 2 model, the number of agile practices was significantly and positively associated with developers' relative fit perceptions ($\beta = 0.152$, $p < 0.01$). This result means that respondents that used more agile practices rated the relative fit of agile SDMs more favorably than did individuals who used fewer agile practices. Specifically, those individuals who used more agile practices perceived a larger, more positive difference between the fit of agile and plan-driven SDMs. This finding provides strong support for H1 in which we hypothesized that the number of agile practices applied positively influences relative fit perceptions.

The block 2 results also show that requirements uncertainty was positively and significantly associated with relative fit perceptions (β = 0.244, $p < 0.05$). This means that respondents who reported on projects characterized by more uncertain requirements believed that agile SDMs provided significantly better fit for the focal project when compared to plan-driven SDMs. Specifically, high requirements uncertainty was associated with a greater, more positive difference between agile and plan-driven fit perceptions. Conversely, individuals who reported on projects with more certain requirements, perceived less difference in agile and plan-driven SDM fit. This finding provides support for hypothesis H2 in which we hypothesized that requirements uncertainty is positively associated with relative fit perceptions. Additionally, this finding is in line with both prior theoretical recommendations and research findings (Boehm & Turner, 2003; Harris et al., 2009; Maruping et al., 2009).

Team development expertise (β = 0.653, $p < 0.001$) was also significantly and positively associated with developers' relative fit perceptions in the block 2 model. Respondents who worked on project teams that possessed higher levels of expertise rated the fit of the agile SDMs more favorably than did individuals who worked on projects with lower levels of development expertise. Specifically, when expertise was high, respondents believed that agile SDMs fit the project contexts much better than plan-driven SDMs. Similarly, when team expertise was lower, the gap between respondents' agile and plan-driven fit perceptions was smaller. This finding provides support for hypothesis H6 in which we hypothesized that team development expertise positively influences relative fit perceptions. Additionally, this finding provides empirical support for prior theoretical recommendations made in the literature regarding the importance of development skill and experience when structuring agile teams (Boehm & Turner, 2003; Cockburn, 2002).

We hypothesized that the main effect of user representative involvement positively influences relative fit perceptions (H3). However, the block 2 results show that user representative involvement by itself was not significantly associated with relative fit perceptions. Additionally, we proposed that the main effect of team size negatively influences relative fit perceptions (H4). The block 2 results show that the team size beta coefficient was negative but that the relative weight of the variable's influence was small (-0.001) and not statistically significant. Finally, we argued that the main effect of team location distribution negatively influences relative fit perceptions (H7). The block 2 results show that the influence of team location distribution was non-significant. Accordingly, our results do not support H3, H4, or H7.

We also hypothesized that the interactive effect of requirements uncertainty and user representative involvement influences relative fit perceptions in a differential manner. The block 2 results show that the two-way interaction between requirements uncertainty and user representative involvement was significantly related to relative fit perceptions (β = -0.289, $p < 0.01$). To better understand the combined influence of these variables' influence on relative fit perceptions, we plotted the corresponding regression line using the guidelines that Aiken and West (1991) state. We calculated the interactive influence of the variables on relative fit at one standard deviation below the mean (low), at the mean (middle), and one standard deviation above the mean (high) for both requirements uncertainty and user representative involvement. Figure 2 show the plotted regression lines for low, middle, and high levels of requirements uncertainty.

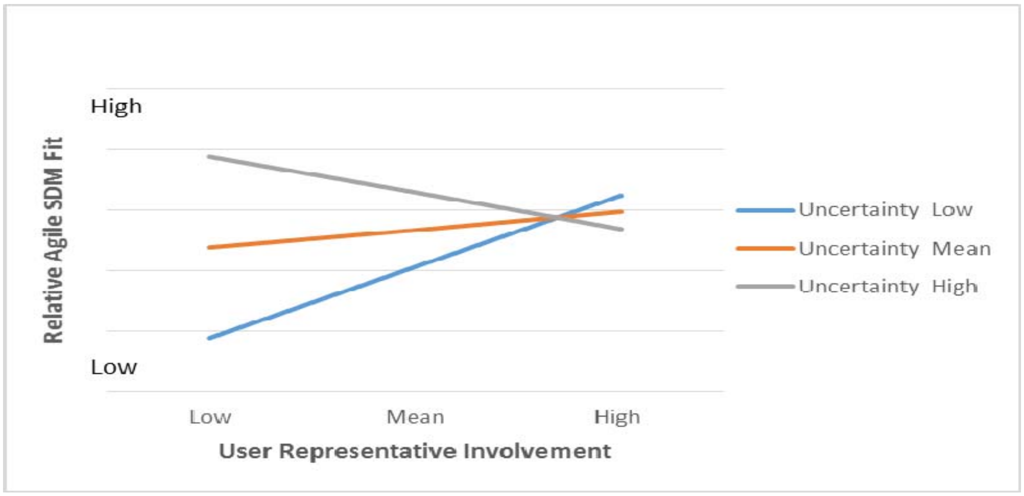

Figure 2. Requirements Uncertainty/User Representative Involvement Interaction

As the chart shows, the interaction between requirements uncertainty and user representative involvement did have a differential effect on relative fit. However, the direction of the interactive influence was opposite to our hypotheses. We argued that relative fit would degrade in low requirements uncertainty contexts with increased user representative involvement. Conversely, we believed that, in highly uncertain projects, increased user representative involvement would result in improvements in relative fit perceptions. As the results show, though, relative fit perceptions in low requirements uncertainty projects were actually better when user representative involvement was at its highest level and relative fit perceptions in high requirements uncertainty projects were actually better when user representative involvement was at its lowest level. Accordingly, the results do not support H3a or H3b. Additionally, the results show that neither team size nor team location distribution significantly interact with requirements uncertainty to influence relative fit perceptions. Therefore, the results do not support H5 or H8.

When reviewing the regression results related to the location distribution, we noted that approximately 60 percent of the sampled respondents reported on projects whose entire agile team was co-located. As such, we wondered whether the weight of the co-located observations was dampening the influence of the location distribution variable for projects that were not co-located. To test if it was, we split the sample into projects that were co-located and those that were not and re-ran the model for the non-co-located group. Table 3 provides the split sample results.

Table 3. Distributed Location Regression Results

	Block 1	**Block 2**
Controls		
Years SD experience	0.021	0.041 (1.961)
Years agile experience	0.035	0.008 (1.774)
Agile bias	0.172	0.136 (1.383)
Main effect variables		
Number of agile practices	0.151+	0.134+ (1.326)
Requirements uncertainty (ReqUnc)	0.291	0.202 (3.284)
User representative involvement (UserInv)	0.236	0.240 (1.198)
Team size	0.004	0.004 (1.093)
Team development expertise	0.759**	0.711** (1.277)
Team location distribution (LocDist)	0.242+	0.276* (1.115)
Interaction Variable		
ReqUnc X UserInv		-0.389* (1.185)
ReqUnc X Team Size		0.008 (1.663)
ReqUnc X LocDist		0.050 (3.674)
R^2	**0.484**	**0.573**
F	**7.064**	**9.719**
Adjusted R2	**0.370**	**0.438**
ΔR2		**0.089**

Notes: n = 49. Variance inflation factors shown in parentheses.
+*p*<0.10, **p*<0.05, ***p*<0.01, ****p*<0.001

For the split sample, the block 2 model provides the best explanation of the variance in the data (R2 = 0.573, F = 9.719, p < 0.05). Significance levels for most of the model variables dropped, but the level for the team location distribution variable increased and became statistically significant (β = 0.276, p < 0.05). However, the direction of the team location distribution beta coefficient was positive—not negative as we had predicted, indicating that the respondents referring to highly distributed projects (spread across countries and time zones) actually rated relative agile fit higher than respondents referring to less distributed projects (spread across buildings or metropolitan). Accordingly, even this more nuanced examination of the collected data did not support H7.

6 Discussion

Table 6 summarizes the results of our hypotheses tests based on the collected data.

Table 4. Results of Hypotheses Tests

Hypothesis	Description	Result
H1	The number of agile practices used on a project positively influences software development professionals’ agile relative SDM fit perceptions.	Supported
H2	Requirements uncertainty positively influences software development professionals’ agile SDM relative fit perceptions.	Supported
H3	User representative involvement positively influences software development professionals’ agile SDM relative fit perceptions.	Not supported
H3a	User representative involvement moderates the influence of requirements uncertainty on software development professionals’ agile SDM relative fit perceptions such that fit is better when high requirements uncertainty coincides with high user representative involvement.	Supported (in opposite direction)
H3b	User representative involvement moderates the influence of requirements uncertainty on software development professionals’ agile SDM relative fit perceptions such that fit is better when low requirements uncertainty coincides with low user representative involvement.	Supported (in opposite direction)
H4	Project team size negatively influences software development professionals’ agile SDM relative fit perceptions.	Not supported
H5	Team size negatively moderates the influence of requirements uncertainty on software development professionals’ agile SDM relative fit perceptions such that the positive impact of requirements uncertainty decreases as team size increases.	Not supported
H6	Team development expertise positively influences software development professionals’ agile SDM relative fit perceptions.	Supported
H7	Team location distribution negatively influences software development professionals’ agile SDM relative fit perceptions.	Not supported
H8	Team location distribution negatively moderates the influence of requirements uncertainty on software development professionals’ agile SDM relative fit perceptions such that the positive impact of requirements uncertainty decreases as team location distribution increases.	Not Supported

In this study, we focused on identifying the project contextual factors that influence software development professionals’ agile SDM relative fit perceptions. The regression results show that the survey respondents rated agile fit higher than plan-driven fit when the project requirements were more uncertain and the development team possessed higher levels of expertise. The requirements uncertainty finding adds merit to our argument that agile SDMs’ organic information-processing strategies are preferable when requirements are more uncertain because they allow a team to continually adapt the direction of the project as it reduces ambiguity (McCann & Galbraith, 1981). Accordingly, our finding substantiates previous recommendations regarding agile SDM appropriateness for dealing with uncertain requirements (Boehm & Turner, 2003; Maruping et al., 2009).

The findings also add merit to our assertion that teams that possess high levels of development expertise are more adept at using agile SDMs and that teams with low levels of development expertise may need the additional rigor provided by plan-driven SDMs. We based this argument on the belief that more expert development teams are better at using agile SDMs’ organic coordination strategies because the team members’ skills and experience provide a knowledge base for dealing with project challenges and uncertainty (Yang et al., 2008). This contribution is important because the literature has suggested that an association between development team expertise and agile SDM appropriateness exists but not empirically investigated it (Boehm & Turner, 2003; Cockburn, 2002).

When developing the research model, we believed that user representative involvement would affect agile SDM relative fit perceptions both directly and via an interaction with requirements uncertainty. We based these beliefs on the fact that user involvement is a key agile principle; some SDMs even recommend that development teams embed user representatives for the duration of the project (Beck et al., 2001b; Schwaber & Beedle, 2003). We argued that, when project requirements are more uncertain, user representatives provide the development team with input so that the development team can identify project outcomes and reduce ambiguity (Harris et al., 2009; McCann & Galbraith, 1981).

In our analysis, user representative involvement did not have a significant direct influence on agile fit perceptions. However, the variable did significantly moderate the influence of requirements uncertainty on agile SDM relative fit perceptions. This finding supports our argument that user involvement is essential when dealing with uncertain requirements and substantiates prior recommendations regarding the importance of balancing user representative involvement against the degree of uncertainty in the project context (Harris et al., 2009). However, we believed that the direction of the interactive influence would be different and hypothesized that relative fit would be better when high requirements uncertainty coincided with high user involvement and when low requirements uncertainty coincided with low user involvement. We based this argument on the assumption that high requirements uncertainty results in high information processing needs that, in turn, necessitate high user involvement (McCann & Galbraith, 1981). Conversely, low requirements uncertainty results in low information processing needs that require less user involvement. Accordingly, we argued that combining high user involvement with low requirements uncertainty could result in distractions and disruptions because the team must spend time interacting with the user representatives when they could be working on previously defined development tasks.

However, plots of the interaction show that the direction of the influence was the opposite of what we expected. Relative fit perceptions decreased on highly uncertain projects when user representative involvement increased. Conversely, relative fit perceptions increased on low uncertainty projects when user representative involvement increased. These results are interesting because they contradict prior theoretical suggestions in the literature (Harris et al., 2009). One possible explanation for the improvement detected in relative fit perceptions when requirements uncertainty was low and user representative involvement was high is that the continued feedback and support supplied by the user representatives throughout the project heightens development team members' work satisfaction, which, in turn, heightens their positive perceptions regarding the fit of the agile SDM. We need further research to fully understand our finding that relative fit perceptions decreased on highly uncertain projects when user representative involvement increased. However, one possible reason for the finding is that, in our study, we focused on measuring the level of user involvement, but we did not assess the quality or consistency of that involvement. It is entirely possible that high levels of user involvement do not equate to productive levels of user involvement. If, for instance, two user representatives were highly involved in a project but they each provided contradictory guidance, their involvement would heighten uncertainty not reduce it. Similarly, if ten user representatives were highly involved but did not possess the authority to make critical decisions, their level of involvement would do little to reduce uncertainty. Accordingly, future research efforts should focus on assessing the quantity, quality, and consistency of user representative involvement and the combined influence of those variables with requirements uncertainty on relative fit perceptions.

When building the research model, we believed that team size would directly influence relative fit perceptions and that it would moderate requirements uncertainty's influences on relative fit perceptions. The information processing model suggest that agile SDMs do not offer adequate coordination mechanisms to support large development teams (Kraut & Streeter, 1995; McCann & Galbraith, 1981) because agile SDMs rely heavily on ad-hoc, informal methods of communication to coordinate work efforts and the effectiveness of those methods degrades as the number of team members increases (Barlow et al., 2011). This argument is in line with prior theoretical suggestions in the literature (Boehm & Turner, 2003) but contradicts qualitative evidence that suggests that large teams can effectively use hybrid SDMs (Batra et al., 2010). However, our analysis did not detect a significant link between team size and relative fit perceptions or an interactive effect of team size and requirements uncertainty on relative fit perceptions. As such, insufficient evidence exists to support using team size as an agile SDM contingency factor.

Still, we do suggest that future researchers more closely examine the way that large teams use agile methods. Theory suggests that agile SDMs should not fit projects that large teams staff (Kraut & Streeter, 1995; McCann & Galbraith, 1981), and yet several respondents reported very high relative fit perceptions for projects that large teams developed. Possibly, their large agile teams were organized into smaller sub-teams with each sub-team following agile practices to simultaneously work on pieces of the overall project during a given iteration. This approach could allow a large team to use an agile SDM to produce the final systems solutions. Accordingly, future research efforts should examine the structure and coordination mechanisms that enable large teams to successfully use agile SDMs.

We also argued that team distribution would negatively influence agile SDM relative fit perceptions and moderate the influence of requirements uncertainty on relative fit perceptions. We based the argument for the direct influence on the assumption that geographical and temporal distance between team members degrades a team's overall ability to efficiently and effectively communicate (McCann & Galbraith, 1981). Prior arguments for colocation are largely based on the premise that geographic and temporal distance impede agile team cohesion, informal communication effectiveness, and informal control mechanism usefulness (Maruping et al., 2009; Ramesh et al., 2006). Because agile SDMs focus on using collaboration over planning and documentation (Becket al., 2001b), we felt that distance between team members would create more communications challenges and result in reduced agile SDM relative fit perceptions (Kraut & Streeter, 1995). Additionally, we believed that distance would inhibit a team's ability to use informal control mechanisms, which would negatively influence individuals' relative fit perceptions.

With the full sample, we did not find a significant relationship between team location distribution and relative fit perceptions. Additionally, team location distribution did not significantly moderate requirements uncertainty's influence on relative fit perceptions. However, our focused analysis of only the non-co-located projects revealed a significant, positive association between team distribution and relative fit perceptions. This finding means that relative fit perceptions were actually better for the non-co-located teams in our sample that were characterized by greater degrees of distribution than for the non-co-located teams characterized by less distribution. This finding contradicts the prevailing assumption that agile development works best for co-located teams and that more distribution creates greater challenges for the development teams (Ågerfalk & Fitzgerald, 2006; Batra et al., 2010; Sarker & Sarker, 2009). One possible explanation of this finding is that advanced tele-presence and collaboration technologies allow distributed teams to more effectively communicate over distance. Persson et al. (2012) provide case evidence that demonstrates how communication technologies can effectively support agile development in distributed environments. Further, Persson et al. (2012) note a case team's use of mediated communication technologies to enact both formal and informal project control throughout the project. Accordingly, as communication technologies advance, distance may be becoming less of a problem for agile development team coordination and management. Accordingly, future research efforts should focus on how electronic collaborative tools and control mechanisms influence agile SDM fit perceptions in distributed development projects and how different control mechanism combinations influence agile fit perceptions in distributed settings.

Finally, we also analyzed whether the number of agile practices used on a project influences relative fit perceptions. Paredes (2015) cautions organizations against adopting agile SDMs in a piecemeal fashion by arguing that each practices contributes to a SDMs' overall value. However, other research has shown that hybrid implementations of agile SDMs can be quite effective in specific project contexts (Baird & Riggins, 2012; Barlow et al., 2011; Batra et al., 2010; Theocharis et al., 2015). In our study, we did not address the questions of whether or not respondents used a complete, single SDM because we believe that the subjectivity regarding the term "agile" prohibits one from reliably assessing an agile SDM's "completeness" at the individual unit of analysis (Fitzgerald, 1997; VersionOne.com, 2015; Wang et al., 2012). Accordingly, we used the number of agile practices used to assess the extensiveness of the SDM that the respondents used. We hypothesized that the number agile practices used positively influences agile SDM relative fit perceptions. The results indicate that the study respondents did rate relative fit higher when they used more agile practices. Specifically, respondents who used more agile practices believed that the agile SDM used suited the project context much better than a plan-driven SDM would have. Conversely, respondents who used fewer agile practices saw less difference between the fit of an agile SDM and a plan-driven SDM for the given project context. This contribution is important because it indicates that software developers perceive agile SDMs to be most beneficial when they employ more agile practices. To summarize, the results of analysis indicate that the best levels of relative fit occur when one uses agile SDMs containing more practices in project contexts characterized by high levels of requirements uncertainty, supportive levels of user representative involvement, and high levels of development expertise.

7 Limitations

When interpreting the results of this study, one should note that the dependent variable of interest was perceived fit and not project performance. We felt strongly that this study needed to focus on agile software development professionals' perceptions regarding the methods they use to deliver systems. Accordingly, we used an anonymous survey instrument at the individual unit of analysis to collect our data. However, we chose not to collect performance-related data because we did not believe that all respondents could accurately report actual performance metrics in an unbiased manner. Prior contingency theory studies in the software development domain have provided evidence that better perceived fit is associated with better task performance (Andres & Zmud, 2001; Barki et al., 2001; Franz, 1985; Mathiassen et al., 2007; McKeen et al., 1994). Future efforts should study project/method fit at the project unit of analysis and, at that time, collect performance data to substantiate the assumed relationship between fit and performance.

We must also note that, when developing our research model, we focused on including project contextual factors that we could reliably assess at the individual unit of analysis and that we could explain using the theoretical principles of contingency theory and the information processing model. We additionally constrained the number of factors in the model based on our concerns about collecting an adequate sample size. Research has discussed numerous other project factors in relation to agile SDM appropriateness, such as complexity, criticality, compliance requirements, organizational culture, management style, and user's systems experience (Barlow et al., 2011; Boehm & Turner, 2005; Cao et al., 2009; Lindvall et al., 2004; Maruping et al., 2009; Ratbe et al., 1999). Accordingly, future research efforts should work to expand the agile contingent project/method fit model to include other important factors and additional interactions between factors.

A second limitation of the current study regards the biases held by some software development professionals that agile methods are superior to plan-driven methods. One could argue that an individual's choice to join an agile group signals favoritism for that method over a plan-driven method. Accordingly, those individuals are likely to rate the fit of an agile method as superior to the fit of a plan-driven method. We fully recognize this limitation and tried to protect against it in several ways. First, we invited individuals belonging to both agile-specific and more general software development groups to participate in the survey. Over one-third of the online groups included in the sample frame were general software development organizations rather than agile-specific ones. Second, we controlled for agile biases by measuring respondents' agile universality beliefs and incorporating the measure in our regression model. Lastly, note that we focused on collecting information from individuals who had recently worked on an agile project. It is possible that those individuals' perceptions regarding relative fit could be biased based on recency. Accordingly, future research efforts should focus on collecting data from both agile and plan-driven projects to ensure that method use does not bias results.

A third limitation of the current study is that the collected responses are all self-reported, retrospective perceptions of project characteristics. Accordingly, we had no way to assess the validity of those responses other than to note that we collected the data from a large heterogeneous sample and the measures used did exhibit acceptable levels of validity and reliability. Finally, the sample size was only moderately large. Accordingly, in the future, one should test the model using a larger sample of respondents.

A fourth limitation of the study concerns measurement of the location distribution construct. To asses that construct, we first asked respondents a yes/no question concerning whether or not all team members were physically located in a common workspace. We then asked individuals who responded negatively to choose from a list of ordinal workspaces distribution options ranging from "all workspaces located in the same building" to "workspaces were located in different countries". Results indicated that 60 percent of respondents replied yes to the original question concerning team co-location. It is possible that this result is an overstatement due to the vagueness of the term "common workspace". Individuals who worked with some team members that were co-located may have chosen the affirmative response since part of the team was co-located. Additionally, the collected location distribution responses were not normally distributed, which could have suppressed the true influence of location distribution on fit perceptions. Accordingly, we need further work to develop a more clearly worded, precise measure of the location distribution construct. Additionally, future efforts should focus on collecting a sample of responses that more closely approximates a normal distribution.

A fifth limitation of the study concerns measuring the agile practices construct. We did not find any predefined list or definition of what does or does not constitute an agile practice in the literature. Accordingly, we developed the list of practices used in the current study by first synthesizing the practices that scrum and extreme programming prescribe and presenting the combined list to a focus group of four agile experts. We then asked the focus group to review the list for accuracy and completeness before incorporating the list into our survey instrument. Accordingly, our list is possibly incomplete. Future research efforts should focus on clarifying the definition of “agile practice” and developing a more robust measure.

An additional concern is the possibility that non-respondent biases have influenced the results. The views of individuals who we included in the sample frame and who chose participate may not reflect the views of the greater software development community. Prior researchers have noted the potential for non-respondent biases to influence agile studies (Espinosa, Slaughter, Kraut, & Herbsleb, 2007). Additionally, because we focused on projects that had used an agile SDM, including non-agile projects in the sample frame could result in different outcomes. Prior researchers have noted the potential for non-respondent biases to influence agile studies (Espinosa et al., 2007). As such, future research efforts should strive to use archival data and include both agile and plan-driven projects in their research frame.

Finally, while techniques do exist for detecting the presence of non-response biases, we did not feel that we needed to apply those techniques to our collected sample. Standard techniques for testing for non-response biases are based on measuring the duration between when an individual was invited to participate in a study and when they actually did participate (Armstrong & Overton, 1977). One then compares the responses of early participants to the responses of late participants to determine if differences exist between the two groups. Our invitations were distributed using social media discussion threads. Most discussion thread tools allow users to configure message notifications settings so that the individual must periodically visit the discussion board to see new thread posts; the discussion board sends no direct notifications. Accordingly, while we know the date on which we posted the invitations, we do not know the dates on which the individual respondents actually saw the invitations. Some individuals might have seen the invitation on the day we posted it but put off responding until a later date. Others might not have seen the invitation until a much later date but responded immediately after seeing it. Accordingly, we feel that applying standard non-response bias tests to our sample could result in misleading results.

8 Conclusion

We developed and empirically tested the contingent agile project/method fit model using survey data collected from systems development professionals who provided information regarding projects they had recently worked on and their perceptions regarding the fit of the methods used on those projects. Linear regression identified several significant relationships between project contextual factors, the number of agile practices used, and respondents’ perceptions regarding method appropriateness. We found that uncertainty, number of agile practices used, and team development expertise significantly influenced system development professionals’ perceptions regarding the fit and usability of agile development methods. Further, we found significant relationships between location distribution and a requirements uncertainty/user representative involvement interaction term. However, these relationships were in the opposite direction from what the prior literature (Barlow et al., 2011; Berger & Beynon-Davies, 2009; Harris et al., 2009) and our interpretation of the factors using of contingency theory and the information processing model suggested. These findings offer evidence that project contextual factors do influence software development professionals’ fit perceptions regarding agile SDMs.

References

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). *Agile software development methods: Review and analysis*. Espoo, Finland: VTT Electronics.

Ågerfalk, P. J., & Fitzgerald, B. (2006). Flexible and distributed software processes: Old petunias in new bowls? *Communications of the ACM*, *49*(10), 26-34.

Aiken, L. S., & West, S. G. (1991). *Multiple regression: Testing and interpreting interactions*. London: Sage.

Andres, H., & Zmud, R. W. (2001). A contingency approach to software project coordination. *Journal of Management Information Systems*, *18*(3), 41-70.

Armstrong, J. S., & Overton, T. S. (1977). Estimating nonresponse bias in mail surveys. *Journal of Marketing Research*, *14*(3), 396-402.

Austin, R. D., & Devin, L. (2009). Weighing the benefits and costs of flexibility in making software: Toward a contingency theory of the determinants of development process design. *Information Systems Research*, *20*(3), 462-477.

Baird, A., & Riggins, F. J. (2012). Planning and sprinting: Use of a hybrid project management methodology within a CIS capstone course. *Journal of Information Systems Education*, *23*(3), 243-257.

Barki, H., Rivard, S., & Talbot, J. (1993). Toward an assessment of software development risk. *Journal of Management Information Systems, 10*(2), 203-225.

Barki, H., Rivard, S., & Talbot, J. (2001). An integrative contingency model of software project risk management. *Journal of Management Information Systems*, *17*(4), 37-69.

Baron, R. M., & Kenny, D. A. (1986). The moderator–mediator variable distinction in social psychological research: Conceptual, strategic, and statistical considerations. *Journal of personality and Social Psychology*, *51*(6), 1173-1182.

Barlow, J. B., Keith, M. J., Wilson, D. W., Schuetzler, R. M., Lowry, P. B., Vance, A., & Giboney, J. S. (2011). Overview and guidance on agile development in large organizations. *Communications of the Association for Information Systems*, *29*, 25-44.

Baskerville, R., & Pries-Heje, J. (2004). Short cycle time systems development. *Information Systems Journal*, *14*(3), 237–264.

Batra, D., Weidong Xia, VanderMeer, D., & Dutta, K. (2010). Balancing agile and structured development approaches to successfully manage large distributed software projects: A case study from the cruise line industry. *Communications of the Association for Information Systems*, *27*, 379-393.

Beck, K. (2000). *Extreme programming explained: Embrace change*. Reading, MA: Addison-Wesley.

Beck, K. (2003). *Test driven development by example*. Reading, MA: Addison-Wesley.

Beck, K., Beedle, M, van Bennkum, A., & Cockburn, A., Cunningham, W., Fowler, M., Greening, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Sutherland, J., & Thomas, D. (2001a). *Principles behind the agile manifesto*.

Beck, K., Beedle, M., van Bennkum, A., Cockburn, A., Cunningham, W., Fowler, M., Greening, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Sutherland, J., & Thomas, D. (2001b). *Manifesto for agile software development*.

Beebe, N. L., Young, D. K., & Chang, F. R. (2014). Framing information security budget requests to influence investment decisions. *Communications of the Association for Information Systems*, *35*, 133-143.

Benbasat, I., Goldstein, D. K., & Mead, M. (1987). The case research strategy in studies of information systems. *MIS Quarterly*, *11*(3), 369-386.

Berger, H., & Beynon-Davies, P. (2009). The utility of rapid application development in large-scale, complex projects. *Information Systems Journal*, *19*(6), 549-570.

Boehm, B., & Turner, R. (2003). Using risk to balance agile and plan-driven methods. *IEEE Computer*, *36*(6), 57-66.

Boehm, B, & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *IEEE Software*, *22*(5), 30-39.

Boone, H. N., & Boone, D. A. (2012). Analyzing Likert data. *Journal of Extension*, *50*(2).

Brechner, E. (2015). *Agile project management with kanban*. Redmond, WA: Microsoft Press.

Burns, T., & Stalker, G. M. (1961). *The management of innovation*. London: Tavistock.

Cao, L., Mohan, K., Xu, P., & Balasubramaniam, R. (2004). How extreme does extreme programming have to be? Adapting XP practices to large-scale projects. In *Proceedings of the 37th Hawaii International Conference on Systems Science*. Hawaii: IEEE Computer Society.

Cao, L., Mohan, K., Xu, P., & Balasubramanim, R. (2009). A framework for adapting agile development methodologies. *European Journal of Information Systems*, *18*, 332-343.

Carifio, J., & Perla, R. (2007). Ten common misunderstandings, misconceptions, persistent myths and urban legends about Likert scales and Likert response formats and their antidotes. *Journal of Social Sciences*, *3*(3), 106-116.

Chilton, M. A., Hardgrave, B. C., & Armstrong, D. J. (2005). Person-job cognitive style fit for software developers: The effect on strain and performance. *Journal of Management Information Systems*, *22*(2), 193-226.

Chin, W. W. (1998). The partial least squares approach to structural equation modeling. *Modern Methods for Business Research*, *295*(2), 295-336.

Chow, T., & Cao, D. (2008). A survey study of critical success factors in agile software projects. *The Journal of Systems and Software*, *81*(6), 961-971.

Cockburn, A. (2002). *Agile software development*. Boston: Addison-Wesley.

Creswell, J. W. (2009). *Research design: Qualitative, quantitative, and mixed methods approaches* (3rd Ed.). Los Angeles, CA: Sage.

Drazin, R., & Van de Ven, A. H. (1985). Alternative forms of fit in contingency theory. *Administrative Science Quarterly*, *30*, 514-539.

Eisenhardt, K. M., & Graebner, M. E. (2007). Theory building from cases: Opportunities and challenges. *Academy of Management Journal*, *50*(1), 25-32.

Espinosa, J. A., Slaughter, S. A., Kraut, R. E., & Herbsleb, J. D. (2007). Familiarity, complexity, and team performance in geographically distributed software development. *Organization Science*, *18*(4), 613-630.

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

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

Franz, C. R. (1985). User leadership in the systems development life cycle: A contingency model. *Journal of Management Information Systems*, *2*(2), 5-25.

Galbraith, J. R. (1977). *Organizational design*. Reading, MA: Addison-Wesley.

Geambasu, C. V., Jianu, I., Jianu, I., & Gavrila, A. (2011). Influence factors for the choice of a software development methodology. *Accounting and Management Information Systems*, *10*(4), 479-494.

Gefen, D., Rigdon, E. E., & Straub, D. (2011). An update and extension to SEM guidelines for administrative and social science research. *MIS Quarterly*, *35*(2), iii-xiv.

Glass, G., Peckman, P., & Sanders, J. (1972). Consequences of failure to meet assumptions underlying the fixed effects analysis of variance and covariance. *Review of Educational Research*, *42*(3), 237-288.

Goel, L., Johnson, N. A., Junglas, I., & Ives, B. (2011). From space to place: Predicting users' intentions to return to virtual worlds. *MIS Quarterly*, *35*(3), 749-772.

Gresov, C. (1989). Exploring fit and misfit with multiple contingencies. *Administrative Science Quarterly*, *34*(3), 431-452.

Harris, M. L., Collins, R. W., & Hevner, A. R. (2009). Control of flexible software development under uncertainty. *Information Systems Research*, *20*(3), 400-419.

Kacmar, C. J., McManus, D. J., Duggan, E. W., Hale, J. E., & Hale, D. P. (2009). Software development methodologies in organizations: Field investigation of use, acceptance, and application. *Information Resources Management Journal, 22*(3), 16-39.

Kautz, K., Madsen, S., & Nørbjerg, J. 2007. Persistent problems and practices in information systems development. *Information Systems Journal, 17*(3), 217-239.

Kraut, R. E., & Streeter, L. (1995). Coordination in software development. *Communications of the ACM*, *38*(3), 69-81.

Kyu Kim, K., & Umanath, N. S. (1992). Structure and perceived effectiveness of software development subunits: A task contingency analysis. *Journal of Management Information Systems*, *9*(3), 157-181.

Lawrend, P., & Lorsch, P. (1986). *Organization and environment: Managing differentiation and integration.* Homewood, Il: Irwin.

Lee, G., & Xia, W. (2010). Toward agile: An integrated analysis of quantitative and qualitative field data on software development agility. *MIS Quarterly*, *34*(1), 87-114.

Lindvall, M., Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D., May, J., & Kahkonen, T. (2004). Agile software development in large organizations. *IEEE Computer*, *37*(12), 27-34.

Lyytinen, K., & Mathiassen, L. (1998). Attention shaping and software risk—a categorical analysis of four classical risk management approaches. *Information Systems Research*, *9*(3), 233-255.

MacCormack, A., Kemerer, C. F., Cusumano, M., & Crandall, B. (2003). Trade-offs between productivity and quality in selecting software development practices. *IEEE Software*, *20*(5), 78-85.

Maruping, L. M., Venkatesh, V., & Agarwal, R. (2009). A control theory perspective on agile methodology use and changing user requirements. *Information Systems Research*, *20*(3), 377-399.

Mathiassen, L., Saarinen, T., Tuunanen, T., & Rossi, M. (2007). A contingency model for requirements development. *Journal of the Association for Information Systems*, *8*(11), 569-597.

McCann, J. E., & Galbraith, J. R. (1981). Interdepartmental relations. In P. Nystron & W. Starbuck (Eds.), *Handbook of organizational designs* (Vol. 2, pp. 60-84). New York: Oxford University Press.

McKeen, J. D., Guimaraes, T., & Wetherbe, J. C. (1994). The relationship between user participation and user satisfaction: An investigation of four contingency factors. *MIS Quarterly*, *18*(4), 427-451.

Norbjerg, J. (2002). Managing incremental development: Combining flexibility and control. In *Proceedings of the 2002 European Conference on Information Systems* (pp. 329-339).

Nunnally, J. C., & Bernstein, I. H. (1994). *Psychometric theory* (3rd ed.). New York: McGraw-Hill.

Paredes, D. (2015). 10 things CIOs need to know about agile development. *CIO.* Retrieved from http://www.cio.co.nz/article/578741/10-things-cios-need-know-about-agile-development/

Persson, J. S., Mathiassen, L., & Aaen, I. (2012). Agile distributed software development: Enacting control through media and context. *Information Systems Journal*, *22*(6), 411-433.

Podsakoff, P. M., MacKenzie, S. B., Lee, J.-Y., & 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.

Ramesh, B., Cao, L., Mohan, K., & Peng, X. (2006). Can distributed software development be agile? *Communications of the ACM*, *49*(10), 41-46.

Ratbe, D., King, W. R., & Young-Gul, K. (1999). The fit between project characteristics and application development methodologies. *The Journal of Computer Information Systems*, *40*(2), 26-33.

Roster, C. A., Rogers, R. D., Hozier, G. C., Jr., Baker, K. G., & Albaum, G. (2007). Management of marketing research projects: Does delivery method matter anymore in survey research? *Journal of Marketing Theory & Practice*, *15*(2), 127-144.

Roy, A., & Berger, P. D. (2005). E-mail and mixed mode database surveys revisited: Exploratory analysis of factors affecting response rates. *Journal of Database Marketing & Customer Strategy Management*, *12*(2), 153-171.

Royce, W. (1987). Managing the development of large software systems: Concepts and techniques. In *Proceedings of the 9th International Conference on Software Engineering* (pp. 328-338).

Ryan, T. (1997). *Modern regression analysis*. New York: Wiley.

Sarker, S., & Sarker, S. (2009). Exploring agility in distributed information systems development teams: An interpretive study in an offshoring context. *Information Systems Research*, *20*(3), 440-461.

Schwaber, K., & Beedle, M. (2003). *Agile software development with scrum*. Upper Saddle River, NJ: Prentice Hall.

Shadish, W. R., Cook, T. D., & Campbell, D. T. (2002). *Experimental and quasi-experimental designs for generalized casual inference*. Boston: Houghton Mifflin Company.

Theocharis, G., Kuhrmann, M., Munch, J., & Diebold, P. (2015). Is water-scrum-fall reality? On the use of agile and traditional development practices. In *Proceedings of the 16th International Product-Focused Software Process Improvement Conference*.

Venkatesh, V., L. Thong, J. Y., & Xu, X. (2012). Consumer acceptance and use of information technology: Extending the unified theory of acceptance and use of technology. *MIS Quarterly*, *36*(1), 157-178.

VersionOne.com. (2009). *4th annual state of Agile^{TM} survey*. Retrieved from https://www.versionone.com/pdf/2009_State_of_Agile_Development_Survey_Results.pdf

VersionOne.com. (2015). *9th annual state of Agile^{TM} survey*. Retrieved from https://www.versionone.com/pdf/state-of-agile-development-survey-ninth.pdf

Wang, X., Conboy, K., & Pikkarainen, M. (2012). Assimilation of agile practices in use. *Information Systems Journal*, *22*(6), 435-455.

Yang, H. D., Kang, H. R., & Mason, R. M. (2008). A exploratory study on meta skills in software development teams: Antecedents of cooperation and personality for shared mental Models. *European Journal of Information Systems*, *17*, 47-61.

Yin, R. K. (2003). *Case study research: Design and methods*. Thousand Oaks, CA: Sage.

Appendix A: Instrument Scales

Unless otherwise noted, we assessed items using a seven-point Likert scale (1 = strongly disagree; 7 = strongly agree).

Table A1. Instrument Scales

Check all that were adhered to	**Agile practices**
	Creation of user stories with acceptance testing criteria
	Iteration/sprint planning
	Release planning
	Daily team meetings/huddles/stand-ups
	Daily software builds
	Development of early prototypes
	Pair programming
	Pair testing
	Regression testing on each build
	Release retrospectives
	Backlog grooming
	Refactoring
	Unit testing
	Collective code ownership
	Adherence to coding standards
	Iteration demonstrations
Requirements uncertainty	
Item	
1	The functional requirements were stable throughout the course of the project.
2	The functional requirements fluctuated in the early phases of the project.
3	The functional requirements were stable in the later phases of the project. [1]
4	The functional requirements did not change much over the course of the project.
User representative involvement	
Item	
1	The functional requirements were stable throughout the course of the project.
2	The functional requirements fluctuated in the early phases of the project. [2]
3	The functional requirements were stable in the later phases of the project.
4	The functional requirements did not change much over the course of the project.
Location distribution	
Were all members of the development team located in a common workspace? Y/N Select the option which most closely describes the location distribution of the team members' work spaces.	
	All workspaces were located in the same building.
	All workspaces were located in the same metropolitan area.
	All workspaces were located in the same country and time zone.
	All workspaces were located in the same country but in different time zones.

[1] Item dropped due to factor analysis loading issues.
[2] Item dropped due to factor analysis loading issues.

Table A1. Instrument Scales

	Workspaces were located in different countries.
Development team expertise	
Item	
1	The development team's expertise in terms of the implementation tools was sufficient for the project.
2	The development team had sufficient technical expertise (hardware, software, and business processes) for the project.
3	The development team's expertise in terms of the development method was sufficient for the project.
4	The development team had insufficient expertise on the software development tools used for the project.
Size	
How many full time people were on the project? _____	
Perceived agile fit	
Item	
1	The development method used fit the needs of the project.
2	The development method used facilitated the team's efforts to meet the project requirements.
3	Use of the development method helped the project team achieve its goals.
4	The development method used was unsuitable for the project.
Perceived structured fit	
Item	
1	A structured development method would have facilitated the team's efforts to meet the project requirements.
2	A structured development method would have fit the needs of the project.
3	A structured development method would have been unsuitable for the project.
4	Use of a structured development method would have helped the project team achieve its goals.

About the Authors

Diana K. Young is an Assistant Professor at Trinity University in San Antonio, Texas. Dr. Young received her PhD in Information Technology from UTSA. She has over 15 years of practical IT experience in systems design and development. Her research interest include software development methodologies, security investment decisions as well as IS student attraction, engagement, and retention. She has published in the *Communications of the AIS (CAIS)*.

Nicole Lang Beebe is the Melvin Lachman Distinguished Professor and an Associate Professor in the Department of Information Systems & Cyber Security, at the University of Texas at San Antonio (UTSA). Dr. Beebe received her PhD in Information Technology from UTSA, M.S. in Criminal Justice from Georgia State University, and a BS in Electrical Engineering from Michigan Technological University. She has over fifteen years of experience in information security and digital forensics, from both the commercial and government sectors, is a Certified Information Systems Security Professional (CISSP), and holds three professional certifications in digital forensics. She has published several journal articles related to information security and digital forensics in *Decision Support Systems* (DSS), *IEEE Transactions of Information Forensics and Security*, *Digital Investigation*, and many other journals. Her research interests include digital forensics, information security, and data analytics.

Glenn Dietrich is a Professor of information systems and cyber security at The University of Texas at San Antonio. He received his PhD in Information Systems from The University of Texas at Austin. Dr. Dietrich developed the curriculum that has been designated by the NSA/DHS as a Center of Academic Excellence in Information Assurance Education and Research. He is currently the Director of the Center of Education and Research for Information and Infrastructure Security. His research interests include security of cyber physical systems, risk assessment, data analysis for security and technology adoption. His research has been presented at international conferences such as AMCIS, and INFORMS as well as several regional conferences. His work has been published in leading information systems journals such as *IEEE Transactions on Engineering Management*, *Communications of the AIS (CAIS)*, and *Decision Support Systems.*

Charles Z. Liu is an associate professor of information systems at The University of Texas at San Antonio. He received his PhD in Management Information Systems from the Katz Graduate School of Business, University of Pittsburgh. His research interests include the economics of information systems, network externalities, mobile apps and standards competition in IT markets. He is an ICIS Doctoral Consortium Fellow (2006) and a recipient of the Net Institute Research Grant (2007).