5-22-2020

When Agile Means Staying: A Moderated Mediated Model

Tenace Kwaku Setor

When Agile Means Staying: A Moderated Mediated Model

Journal:	*Journal of Computer Information Systems*
Manuscript ID	UCIS-2020-0030.R2
Manuscript Type:	Research
Keywords:	Agile development, job satisfaction, stay intention, job characteristics theory
Abstract:	The design of software development methods focuses on improving task processes, including accommodating changing user requirements and accelerating product delivery. However, there is limited research on how the use of different software development methods impacts IT professionals' perceptions of organizational mobility. Drawing on concepts from the agile development literature and job characteristics theory, we formulate a moderated mediation model explicating the mechanism and the condition under which agile development use exerts an influence on IT professionals' intention to stay with their current employer. Specifically, we examine job satisfaction as mediating the effect of using agile development on the intention to stay as well as how the strength of the mediated relationship differs across firms. We test our hypotheses using a sample of 32,389 software developers. We find that job satisfaction fully mediates the effect of using agile development on the intention to stay. The strength of the mediation effect is significantly different for large and small firms.

When Agile Means Staying: A Moderated Mediated Model

Abstract

The design of software development methods focuses on improving task processes, including accommodating changing user requirements and accelerating product delivery. However, there is limited research on how the use of different software development methods impacts IT professionals' perceptions of organizational mobility. Drawing on concepts from the agile development literature and job characteristics theory, we formulate a moderated mediation model explicating the mechanism and the condition under which agile development use exerts an influence on IT professionals' intention to stay with their current employer. Specifically, we examine job satisfaction as mediating the effect of using agile development on the intention to stay as well as how the strength of the mediated relationship differs across firms. We test our hypotheses using a sample of 32,389 software developers. We find that job satisfaction fully mediates the effect of using agile development on the intention to stay. The strength of the mediation effect is significantly different for large and small firms.

Keywords: Agile development, job satisfaction, intention to stay, job characteristics theory

Introduction

Much of the effort in redesigning software development focuses on revamping processes, including accommodating fluctuating user requirements and accelerating product delivery.[1, 2] The disproportionate attention paid to process improvements, while important, upstages the critical roles developers play in the sociotechnical system of software production. Indeed, the analysis of a production system cannot be complete without understanding the link between the work system and how participants respond to it, both emotionally and cognitively.[3]

In practice, the successful redesign of software development methods is predicated on the affective experiences that IT professionals derive from applying the new methods.[4] Work redesign can alter the psychological states of employees, prompt internally motivated work behaviors, and affect employee satisfaction.[5, 6] New work methods that are responsible for negative affective experiences, even when the methods are proven to be productive, could result in adverse effects for organizations, including lower employee retention rates.

Many organizations adopt contemporary software development methods without a clear understanding of their impact on individual IT professional outcomes.[7] While a substantial body of work has found that the use of agile development increases IT professionals' *satisfaction with their jobs*[2, 4, 8, 9], the literature does not address how the use of agile development relates to IT professionals' *perceptions of their organizations or employers*. It could conceivably be argued that agile development practices impact not only job attitudes but also IT professionals' perceptions regarding their organizations and employers. Typically, the adoption of a particular software development methodology is organization-specific. Potential adverse consequences of work redesign highlight the need to understand IT professionals' intention to stay as a result of using contemporary development methods to produce software.[7]

In this paper, we advance a theoretical model that examines the relationship between the use of agile development and IT professionals' intention to stay with their current employer. We rely upon and extend two theoretical perspectives. First, we draw on job characteristics theory, which delineates the relationships between core dimensions of work and psychological outcomes.[10] Second, we draw on the agile development literature and the concepts of agile development to emphasize the importance of the skill variety, job autonomy, and feedback it provides. Integrating insights from job characteristics theory and agile development, we relate agile development practices to the intention to stay. We argue that core dimensions of these practices infuse software development with a positive affect that suppresses thoughts of leaving in software developers.

The present research provides a point of departure from prior IT research that has examined the positive affective experiences connected to the use of agile development.[2, 4, 8, 9] Although we concur with the benefits of examining the affective experiences of those practicing agile development, we see even greater benefit in shifting the focus to organizational mobility outcomes. Specifically, we build on the extant literature by considering how the use of agile development can have effects beyond positive affective experiences, and specifically, effects on software developers' intention to stay with the organization.

Whereas the primary aim of our study is to offer a new perspective that is missing in the extant literature on the implications of using agile development, we also believe that our work is linked to conversations in the literature concerning the use of agile practices across different firms[11]. We investigate a condition under which the strength of the mediated relationship between agile development use, job satisfaction, and the intention to stay varies. Prior research hints that the risks associated with agile development use vary by firm size.[11] This finding implies that the levels of affective experiences derived from the use of agile development may not be homogenous across firm sizes.

Theoretical Model and Hypothesis Development

In this section, we provide an overview of Hackman and Oldham's[10] job characteristics theory and review the literature on agile development, and we synthesize the core tenets of job characteristics theory with the concepts of agile development methods. Figure 1 presents our research model, which is inspired by previous work in the IT job design literature that has found that the characteristics of IT work influence IT professionals' motivational outcomes. [9, 12, 13]

---Insert Figure 1 About Here---

Job Characteristics Theory

Job characteristics theory delineates the relationships between the core dimensions of work and psychological responses to it.[10] According to this theory, the core dimensions of work are the skill variety that is involved, the significance of the task, the identity of the task, and the autonomy and feedback that the work offers. The skill variety is the degree to which the work requires the use of a broad range of skills and competencies. The task significance is the extent to which the work has a meaningful impact on the lives of people both within and outside of the immediate organization. The task identity is the extent to which the work has a beginning and an end with a visible outcome. Autonomy is the degree to which the work provides substantial freedom for the individual in scheduling the work and determining the procedures for carrying out work tasks. Finally, feedback is the extent to which individuals receive direct and clear information regarding the performance of their work tasks.

Job characteristics theory postulates that work that is designed to include these core dimensions evokes positive affect that is reinforcing to individuals and incentivizes them to continue to perform well. The individual experiences positive affect to the extent that “she learns (knowledge of results) that she personally (experienced responsibility) has performed well on a task that is valuable and internally rewarding (experienced meaningfulness)”.[10(p.256)] That is, the core dimensions of work foster the emergence of three psychological states—knowledge of the actual results of the work, experienced responsibility for the work outcomes, and experienced meaningfulness of the work—which in turn influence individual and work outcomes.

Software Development Approaches: Traditional and Agile Methods

Traditional software development methods follow a linear and sequential approach. The development process is stringently broken down into four stages: planning, analysis, design, and implementation. Each stage is generally finished before the next stage begins. Germane to the traditional approach is its focus on milestones and prespecified deliverables at each stage of the development cycle.[14] This focus adds predictability to the development process. The traditional approach mandates extensive documentation of requirements and training materials.[15] In addition, the approach requires that the roles and responsibilities of the development team are well established and defined.

A value proposition of the traditional plan-driven approach is that it allows the development team to maintain a more detailed and robust project scope.[16] As a result, developers adhere to stringent development procedures. Another benefit of using the traditional approach is that, due to its extensive documentation requirement, the approach is well-suited to large projects that are joined by new individuals at different points in the development lifecycle.[15] New team members can readily refer to the documentation to get up to speed with the development project.

Despite the many advantages of the traditional plan-driven approach, it does not lend itself well to changing user requirements. The prespecified deliverables lock in the software design specifications even when the user requirements may have changed.[16] Agile development practices were developed to address this drawback of the traditional approach.

Relative to the traditional plan-driven approach, agile development is a fluid and self-adaptive approach to developing software.[17] At the core of agile development are the mechanisms of iterative and incremental development.[14] These approaches to development enable software teams to adequately respond to changing user requirements by frequently upgrading the software. The working versions usually have discrete units of software functionality or a subset of the requested software features. Lean principles are sometimes incorporated into agile practices to facilitate code and functional freezes.[18] There exist many different agile development methods, including Scrum, XP (eXtreme Programming), Lean Programming, Kanban, and DSDM (Dynamic Systems Development Method). The methods differ in their implementations and specific practices. However, they share the same core concepts that distinguish agile development from the traditional plan-driven approach.[19] The concepts reiterate the importance of skill variety, job autonomy, and customer collaboration and feedback.

In contrast to the plan-driven approach, agile development emphasizes individuals rather than processes and tools.[20] Agile development involves assembling individual software developers who bring unique but complementary skillsets to the team. They work together to develop the software solution while exchanging knowledge and sharing their varied skillsets.[20] Agile development challenges individual developers to learn and acquire a variety of skills that are different from their preexisting stock of skills.

The iterative nature of agile development involves shorter release cycles, which places time pressure on agile teams to create software prototypes. The time pressure necessitates that managers cede a high degree of control to the developers, granting the software teams the autonomy to develop their approaches to solving problems.[1, 21] Decision-making power is decentralized and transferred to the developers who carry out the actual tasks. The IT project management literature supports the notion that agile development offers considerable latitude to software teams in making decisions.[1, 20] These decisions include scheduling the work, determining work procedures, assigning tasks, determining communication protocols, and managing changes.

Regarding customer collaboration, agile development values interactions via rapid user reviews and feedback.[22] Customers are considered an integral part of the development team throughout the entire development process.[14] Compared to the traditional plan-driven approach, agile methods enforce customer involvement.[1]

In sum, agile development incorporates dynamic and iterative approaches to developing software. Iterative development empowers software teams to overcome the changing user requirement problems that hamstring the traditional plan-driven approach. At the core of the various agile development methods are principles that promote skill variety, job autonomy, client participation, and feedback.[9, 14, 16]

Although their characteristics make agile methods effective for adapting more quickly to changing business requirements, it should also be noted that agile practices come with their own challenges that differ from the traditional methodologies. Communication breakdown is a common occurrence among agile teams.[23] The risk of communication breakdown is even more prevalent among distributed agile teams because they are limited in their modes of communication and do not have the benefit of the extensive documentation required by traditional methodologies.[24] There is also a potential risk of communication breakdown between the developers and customers because customers may not be readily available during the development process.[23]

To address the challenges associated with adopting a specific development approach, teams and organizations adopt hybrid approaches by combining well-structured development processes and flexible agile practices.[25] The variety of hybrid models follow a pattern in which the traditional plan-driven approach serves as the framework that incorporates agile practices.[26] Hybrid approaches are often neither planned nor designed but are a result of a natural evolution of different development practices that are known to work.[26]

Use of Agile Development Practices

A significant body of literature has examined the use of agile development practices in the production of software, resulting in two dominant streams of research. The first stream examines the drivers that lead to the adoption and use of agile development practices.[27-29] The second stream examines the outcomes associated with the adoption and use of agile methods. This latter stream has consistently supported the view that agile development practices elevate feelings of job satisfaction among software development teams.[2, 4, 9] Tripp et al.[9] explain the relationship between the use of agile development and job satisfaction by invoking job characteristics theory. Agile development practices organize software development in a fashion that establishes a client relationship, combines a variety of skills, and encourages continuous feedback.[9] These characteristics of agile development lead to a greater feeling of job satisfaction because they provide meaningfulness to the software development process and an opportunity to improve via feedback.

Departing from examination of the individual-centric outcomes of agile development, some studies explore software-centric outcomes of agile development. For example, Wellington et al.[30] assigned two teams of computer science students the same software project, with one team using the plan-driven approach and the other an agile method. By the project's end, the agile team scored consistently higher on quality metrics, including source code/design quality and product quality—the level of functionality and usability, than the team using a plan-driven approach. Other studies[21, 31] that have sampled IT professionals corroborate Wellington et al.'s[30] findings. Specifically, clients and quality assurance teams report 13% fewer defects in software modules developed by agile teams than those developed by non-agile teams[31], and Maruping et al.[21] found evidence to support their prediction that agile methodology enhances software project quality as measured by the objective indicators of bug severity, component complexity, coordinative complexity, and dynamic complexity. Capiluppi et al.[32] attribute the project quality benefits of agile development to the high levels of complexity control and the iterations, in which customers can provide broader and richer feedback. The growth of software complexity is inevitable in the evolution of software development, and this growth is effectively managed in agile development using complexity control mechanisms such as refactoring.[9]

Use of Agile Development and IT Professionals' Intention to Stay: The Mediating Role of Job Satisfaction

Retaining IT professionals is a perennial challenge faced by managers. While external market conditions contribute to the high turnover culture in IT, scholars find that the work environment and the nature of the jobs drive turnover.[13, 33] IT professionals leave their organizations when their managers fail to design meaningful jobs.[12] This is particularly the case for many software jobs. Drawing inspiration from job characteristics theory[10], we argue that the skill variety, job autonomy, and feedback that characterize agile development cause IT professionals to experience positive feelings about their work[9], and that this is the mechanism through which agile development methods exert a positive effect on IT professionals' intention to stay.

Agile developers perform a variety of tasks other than coding. In collaboration with quality assurance individuals and single points of contact (SPOCs), agile developers must estimate user stories and develop a precise understanding of use cases in different business domains.[8] This challenges agile teams to develop competencies in nontechnical domains and requires agile developers to combine skills and share responsibilities. By acquiring varied skills, agile developers experience software development as significant for their professional development. Work holds positive meaning for them because it expands their range of skills, expertise, and knowledge, and individuals are more satisfied with jobs that are designed to challenge them to use an assortment of their skills to complete work tasks.[21]

Agile development practices prescribe that developers choose their own methods to perform their work, make independent work-related decisions, and schedule their work tasks.[20] These independent work-related and scheduling decisions increase perceptions of job autonomy.[6, 34] Individuals derive pleasure and satisfaction in choosing their methods for performing work, making independent work-related decisions, and scheduling work tasks.

Further, customer collaboration and interactions, which are at the core of agile development, also prompt employee job satisfaction.[8] Through sustained interactions, IT professionals have direct and continuous access to performance-related information from the customers who are the key stakeholders. Continuous feedback facilitated by sustained interactions is instrumental for achieving performance goals, which results in the employees' positive feelings about their job.[9]

Work methods crafted to elicit affective responses have implications for employee outcomes, including intentions to stay and related behaviors. We argue that agile development practices reduce the intention to withdraw or strengthen the intention to stay because given these practices, IT professionals experience satisfaction with their jobs. Insofar as the adoption of a particular software development methodology remains at the discretion of the organization[16], we expect IT professionals to attribute their job satisfaction that results from the use of agile development to the organizational context and, as a consequence, to stay with their employers.

Hypothesis 1: Job satisfaction mediates the positive relationship between the use of agile development and the intention to stay.

The Moderating Role of Firm Size in the Mediated Relationship Between Use of Agile Development, Job Satisfaction, and the Intention to Stay

While larger organizations have adapted agile methods to include hybrid models[25], the general consensus in the software development literature is that the implementation of pure agile practices is often challenging for larger organizations.[35] Larger organizations tend to work on complex IT projects by deploying multiple teams that must engage in cross-function activities. The larger number of teams adds complexity in managing team–team and team–client interactions.[36] This complexity becomes even more challenging when deploying globally distributed software teams, as is commonly done by larger organizations.

We argue that in larger organizations, cross-team communication and interaction—a critical requirement of agile practices—is difficult to achieve, although not impossible.[37, 38] Agile teams rely heavily on in-person or face-to-face conversations during the development process.[36] While use of videoconferencing technologies has been advocated to improve communication within collocated agile teams[39], these technologies may not be as effective as in-person meetings.

There is an increased risk of communication breakdowns in these software teams (and with their clients) when agile methods are used in larger organizations.[40] We contend that the risk of communication breakdowns exacerbated by the use of agile methods in larger organizations can adversely affect individual IT professionals. The reason is that a breakdown in communication among teams and clients results in significant work disruption and induces stress. Exposure to stress in the IT work context has been linked to negative affective experiences, which should weaken the role of job satisfaction in explaining the link between the use of agile development and the intention to stay.

Hypothesis 2: Firm size moderates the mediated relationship between use of agile development and the intention to stay via job satisfaction, such that the strength of the mediated relationship is weaker in large firms than in small firms.

METHODS

Data Collection

This study's sample and data are drawn from the Stack Overflow survey dataset. Stack Overflow is an online developer platform founded in 2008 and owned by the Stack Exchange Network. The platform enables programmers to learn, collaborate, and exchange knowledge about a wide range of computer programming topics. As of 2019, the platform had over 10 million registered users.

The Stack Overflow Developers' Survey has been run annually since 2011. We test our model using the 2018 Stack Overflow Developers' Survey, which was conducted between January 8 and January 28, 2018. The 2018 Stack Overflow Developers' Survey is ideal for this study because it asks questions about a broad range of topics, including work attitudes and perceptions, labor market experiences, and software development practices. Stack Overflow contacted the entire population of its registered members via email to ask them to respond to the survey. The email invitations contained unique links to minimize the possibility of respondents submitting multiple responses.[41] Respondents received digital badges to encourage participation. The 2018 Stack Overflow Developers' Survey includes 101,592 respondents from 183 countries.

Sample

To construct our sample, we consider all IT professionals from the universe of 101,592 individuals who responded to the 2018 Stack Overflow Developers' Survey. Our sample inclusion criterion is that respondents must have provided data for all of our study's variables. Overall, our sample includes a total of 32,389 individuals from 165 countries: 92.7% males, 1.3% females, and 6% others. Regarding ethnicity, 74.4% of the sample is Caucasian, 13.8% is Asian, 6.2% is Hispanic, 3.3% is Middle Eastern, 1.9% is Black, and 0.4% is Native American.

IT jobs in the 2018 Stack Overflow Developers' Survey dataset include back-end, front-end, and full-stack developers, enterprise and desktop application developers, mobile developers, game developers, engineering and product managers, data and business analysts, and database and systems administrators. We recoded the job titles into the broader categories of developers, managers, systems administrators, and analysts. Table 1 reports the number of respondents in the respective IT job roles.

---Insert Table 1 About Here---

Measures

In this section, we provide details about how we operationalize each variable. Table 2 provides a list of all the variables used in the current study and their respective definitions.

Dependent Variable. Our dependent variable is the intention to stay. The 2018 Stack Overflow Survey asked respondents about their intention to stay with their current employer. We code *Intention to Stay* as 1 if a respondent's response to the survey item "Which of the following best describes your current job-seeking status?" was "I am not interested in new job opportunities," and as 0 if a respondent selected either "I am actively looking for a job" or "I am not actively looking, but I am open to new opportunities."

Independent Variable. Our independent variable is use of agile development. *Agile Development Use* is operationalized as the use of agile methods for the production of software in the respondent's firm. The agile methods identified in the dataset include Extreme programming, Scrum, Kanban, Lean, Pair, and Mob programming. *Agile Development Use* is a categorical variable coded as 1 if a respondent reported using agile development methods for the production of software in their firm, and as 0 if the respondent reported using non-agile methods, including the ISO 9001 or IEEE 12207 (Waterfall) model, in their firm.

Mediator Variable. Our mediator variable is *Job Satisfaction*. The job satisfaction measure in the 2018 Stack Overflow Developer's Survey is a single item providing a general indication of respondents' levels of affective attachment to their current job. *Job Satisfaction* is measured on a 7-point Likert scale ranging from 1 (*Extremely dissatisfied*) to 7 (*Extremely satisfied*).

Moderator Variable. Our moderator variable is *Firm Size*. Consistent with many previous studies (e.g.,[42]), we use the number of employees in the firm to operationalize firm size. *Firm Size* is a derived variable coded as 1 = Fewer than 10 employees; 2 = 10 to 19; 3 = 20 to 99; 4 = 100 to 499; 5 = 500 to 999; 6 = 1,000 to 4,999; 7 = 5,000 to 9,999; and 8 = 10,000 or more employees.

Controls. To rule out alternative explanations for the relationship between the use of agile development and IT professionals' intention to stay, we include several control variables.

We control for demographic variables known to influence job mobility.[43] We code *Gender* as 1 if gender identity was reported as *"*male,*"* 2 if it was reported as "female," and 0 for "other." We code *Ethnicity* as 1 if the reported ethnicity was "Caucasian," 2 for "Asian," 3 for "Hispanic," 4 for "Middle Eastern," 5 for "Black," and 6 for "Native American." We decided against controlling for age since it was highly correlated with IT experience.

The human capital variables included in this study are *Education Level*, *IT Experience*, and *Job Role*. *Education Level* is an ordered categorical variable that is coded as 0 if the respondent reported having no formal education, while 1 = "high school diploma," 2 = "some college education," 3 = "associate degree," 4 = "bachelor's degree," and 5 = "postgraduate degree." *IT Experience* is a continuous variable that indicates the number of years a respondent has held any IT job. *Job Role* is a dummy variable that represents *Developers* (coded as 1), *Managers* (coded as 2) *Systems Administrators* (coded as 3) and *Systems Analysts* (coded as 0). We also control for salary level and computing workload. *Salary Level* is a natural logarithmic measure of the annual salaries of respondents in US dollars. *Computing Workload*, an important indicator of turnover behaviors among IT professionals, measures the number of hours respondents spent at their work computers. *Computing Workload* is coded as 0 for less than 1 hour, 1 for 1–4 hours, 2 for 5–8 hours, 3 for 9–12 hours, and 4 for more than 12 hours.

---Insert Table 2 About Here---

Data Analysis

A criterion for establishing mediation is demonstrating that the causal variable is correlated with the outcome.[44] We test the association between agile development use and the intention to stay using the logistic regression model; logistic regression is an analytical technique that models a dichotomous or binary outcome as a linear combination of the covariates. The results of the logistic regression are shown in Table 3. For easier interpretation of our results, we have transformed the regression coefficients into average marginal effects., which describe a change in the dependent variable as a function of the change in the independent variable, holding all covariates constant. As can be seen in Table 3, the results indicate a positive and significant relationship between the use of agile development and IT professionals' intention to stay (b = 0.011, $p < 0.01$, 95% CI [0.010, 0.021]). The results indicate that there is an effect that may be mediated.

--- Insert Table 3 About Here ---

We estimate the mediation effects following Hicks and Tingley's[45] approach. Standard procedures of mediation analysis (e.g.,[44]) are organized and implemented within the framework of linear regression models, and thus the procedures cannot be correctly extended to a nonlinear model such as the present study's model. The statistical limitations associated with applying standard or traditional approaches of mediation analysis to nonlinear models are well documented.[46] Following Imai et al.'s[46] recommendation, we employ Hicks and Tingley's[45] statistical approach to compute the point estimates, mediation, and direct effects. This approach is generalizable to nonlinear models because it simulates "predicated values of the mediator/outcome variable, which are unobserved".[45(p.4)]

We analyze our hypotheses by including country fixed effects in our empirical model. We include these to capture the possible variation in the employability of IT professionals across different geographic regions.

Results

Table 4 shows the correlations and descriptive statistics. The results of the mediation analysis are presented in Table 5 and Figure 2. The results of the moderated mediation analysis are presented in Table 6.

--- Insert Table 4 About Here ---

The results of the mediation analysis confirm a significant total effect ($b_{\text{TE}} = 0.014$, $p < 0.001$, 95% CI [0.006, 0.020]) as well as a significant indirect effect ($b_{\text{IE}} = 0.007$, $p < 0.001$, 95% CI [0.005, 0.010]).

--- Insert Table 5 About Here ---

The direct effect, however, is not significantly different from zero ($b_{\text{DE}} = 0.007$, *n.s.*, 95% CI [-0.002, 0.010]). Overall, the results indicate that job satisfaction fully mediates the effect of the use of agile development on the intention to stay. Hypothesis 1, which predicted that job satisfaction mediates the relationship between the use of agile development and the intention to stay, is therefore supported.

--- Insert Figure 2 About Here ---

Hypothesis 2 predicted that the strength of the mediated relationship between the use of agile development, job satisfaction, and the intention to stay is weaker in large firms than in small firms. We tested this moderated mediation hypothesis by operationalizing large and small firms as one standard deviation above and one below the mean score, respectively. The results in Table 6 indicate that the conditional indirect effects are significantly different, such that the mediated effect of the use of agile methodology on IT professionals' intention to stay through job satisfaction is weaker for IT professionals in large firms than for those in small firms ($b_{\text{IE (Large – Small)}} = -0.004$, $p < 0.05$, 95% CI [-0.011, -0.001]). Hypothesis 2 is therefore supported.

--- Insert Table 6 About Here ---

Discussion

In this study, we demonstrate that the mechanism through which the use of agile development influences the intention to stay with the organization is the satisfaction that IT professionals derive from the characteristics of their work. Our findings show that the influence of agile development methods on IT professionals' intention to stay via job satisfaction differs for different firm sizes. Specifically, we uncovered that the effect of the use of agile methodology on the intention to stay via job satisfaction is weaker in large firms than in small firms. This is consistent with our argument that the risks that threaten to derail software projects are more pronounced in large firms. These risks can generate stress for IT professionals, which lowers their job satisfaction levels.

Cumulatively, our findings resonate with prior IT research on job design. For example, in studies that explored the job characteristic determinants of employee turnover, Thatcher et al.[13] and Igbaria et al.[47] found that dimensions of the job design, including skill variety and autonomy, are related to increased job satisfaction levels for IT workers. Non-IT studies also corroborate our findings. Spector and Jex[48] concluded that jobs with autonomy and continuous feedback are negatively correlated with intentions to leave.

Implications for Research

The findings of the current study have implications for research on software job design and contribute to the literature in the following ways. First, our study is novel in its approach of examining individual IT professional outcomes of software development methods. Specifically, to the best of our knowledge, our study is the first to advance a moderated mediated model of the use of agile development, job satisfaction, the intention to stay, and firm size. In doing so, we extend the agile development literature, which has primarily examined the process efficiency and client-oriented consequences of adopting agile development methods, to include IT professional outcomes. We provide insights into the relative importance of the use of agile development in determining individual perceptions regarding organizational mobility. Structural changes in the design of software development impact not only task processes and the quality of software but also job attitudes and cognitions.

In agreement with Grant's[49] theoretical propositions about job design, we also call for future research to investigate how the use of agile development shapes the identity of software developers. Grant proposes that relational job characteristics—that is, the characteristics of jobs that are designed to provide opportunities for workers to have sustained interactions with people affected by their work—shape the identities of workers. We suspect that the relational job characteristics of agile development resulting from sustained participation and interactions with customers give meaningfulness to work and conceivably shape the professional identities of software developers as social change agents.

Second, previous IT studies have investigated general characteristics of job design and their effect on employment mobility.[13, 47] However, our study narrows these down to specific job designs or methods, namely, agile development and the traditional plan-driven approach. Doing so provides a granular understanding of how the nature of IT work impacts workers' perceptions. By placing software development methods within the context of job design, we begin to map out the conceptual landscape of IT-specific job design.

Our study underscores the need to examine how specific work methods in other IT domains impact job attitudes and perceptions. For example, the network development lifecycle specifies the process of building computer networks. There are several different network lifecycle models (e.g., Plan-Build-Manage, Prepare-Plan-Design-Implement-Operate-Optimise) that follow the waterfall approach. How do the differences in the lifecycle models lend themselves to determining individual work attitudes and perceptions? This question deserves scholarly attention in order to further our understanding of designing IT-specific jobs.

Finally, the choice of a software development model is organization-specific. It is a top-down decision that ensures that software development practices are streamlined company-wide, in the bid to achieve a consistent level of output. The successful adoption and use of a particular software development model are contingent on how a firm's management introduces and implements the model.[50] We suspect that variations in firm-level characteristics may moderate the relationship between the use of agile development and IT professional outcomes.

For example, the successful adoption and use of agile development may in part be determined by an organizational culture that is flexible and responsive to change.[51] Software developers in such organizations may see minimal disruptions to their job design and thus are likely to show positive job attitudes. In contrast, if the organizational culture conflicts with the core values of agile development, we expect disruptions in work processes, which in turn may influence individual job attitudes. Accordingly, we call for research that examines the cross-level effects of firm characteristics on the relationship between the software development methods that are used and individual IT professional outcomes.

Implications for Practice

Our findings have implications for managing IT professionals. The results of our study can inform human resource practices in IT firms that are geared toward employee retention. Our findings highlight the importance of designing software jobs that provide autonomy, skill variety, and opportunities for employees to interact with customers as a means to retain IT professionals.

This study provides managers with evidence that adopting agile development can raise job satisfaction levels. We caution, however, that careful consideration must be given to the organizational context. Large organizations that seek to adopt agile methods should consider other management practices that might elevate the satisfaction levels of IT professionals. The findings of our study provide an incentive for small organizations to consider adopting agile methods as a retention strategy.

The design science and software engineering communities may also find our results relevant. We draw attention to the need to consider the psychological consequences of redesigning software methodologies. Our study provides a glimpse into how future models of IT work could be designed to elicit positive job attitudes.

Limitations and Future Research

Our study is not without limitations. First, considering that we use self-report data regarding the intention to stay, which may not necessarily translate to actual behavior, our study lays a foundation for future research to build on our findings. Do intentions to stay prompted by the use of agile practices translate to actual behavior? Examining the distal consequences of the use of agile development will further enrich our understanding of the link between IT job design and turnover behavior.

Second, our measure of the use of agile development is not perfect. Rigorous adherence to a specific software development method is low in the developer community.[52] Rather, software developers employ a variety of development methods, including hybrid models of agile and traditional approaches. Our measure of the use of agile development, which includes Extreme programming, Scrum, Kanban, Lean, Pair, and Mob programming, effectively captures agility in the development process.[16] Together, these practices fulfill the core tenets of the agile manifesto. Future research may replicate the findings of the present research using a more refined conceptualization and measurement of the use of agile development as well as hybrid models.

Finally, due to limitations in our dataset, we were unable to analyze our data at the team and project levels. Developers are often embedded within teams that work on development projects with designated leaders. In agile development, the leadership role may be rotated in every iteration. In line with the social contagion effect, it is conceivable that the individual attitudes and cognitions resulting from the use of agile practices may be influenced by the characteristics of team leaders. Future research examining how the use of agile methods affects individual developers should take into account the potential influence of team leadership characteristics.

Conclusion

Organizations have adopted and implemented agile development practices to improve the software development process and the quality of software.[20, 50] However, empirical research has shown that the use of agile development practices influences individual IT professional outcomes as well.[9, 53] Building on this stream of research, the present study theorizes the mechanism and the condition under which the use of agile development influences cognitions regarding staying with the organization. This study finds that the effect of the use of agile development on the intention to stay is explained by job satisfaction. In addition, we find that the mediated effect of the use of agile development on the intention to stay is weaker in large firms than in small firms.

References

1. Beck K, Andres C. Extreme programming explained : Embrace change. Addison-Wesley.

2. Dyba T, Dingsoyr T. What do we know about agile software development? IEEE Software 2009:26(5).

3. Mathieu M, Eschleman KJ, Cheng D. Meta-analytic and multiwave comparison of emotional support and instrumental support in the workplace. Journal of occupational health psychology 2019:24(3): 387.

4. Mannaro K, Melis M, Marchesi M. Empirical analysis on the satisfaction of it employees comparing xp practices with other software development methodologies. Springer. Extreme programming and agile processes in software engineering.

5. Peiró JM, Bayona JA, Caballer A, Di Fabio A. Importance of work characteristics affects job performance: The mediating role of individual dispositions on the work design-performance relationships. Personality and Individual Differences 2020:157(109808.

6. Tims M, Derks D, Bakker AB. Job crafting and its relationships with person-job fit and meaningfulness: A three-wave study. Journal of Vocational Behavior 2016:92(44-53).

7. Hong W, Thong JYL, Chasalow L, Dhillon G. User acceptance of agile information systems: A model and empirical test. Journal of Management Information Systems 2011:28(1): 235-272.

8. Tessem B, Maurer F. Job satisfaction and motivation in a large agile team. Springer. Agile processes in software engineering and extreme programming.

9. Tripp JF, Riemenschneider CK, Thatcher JB. Job satisfaction in agile development teams: Agile development as work redesign. Journal of the Association for Information Systems 2016:17(4): 267-307.

10. Hackman JR, Oldham GR. Motivation through the design of work: Test of a theory. Organizational Behavior and Human Performance 1976:16(2): 250-279.

11. Barlow J, Giboney J, Keith MJ, Wilson D, Schuetzler R, Lowry PB, Vance A. Overview and guidance on agile development in large organizations. Communications of the Association for Information Systems 2011:29(2): 25-44.

12. Koh C, Joseph D. Experienced meaningfulness and calling: Effects on it professionals' retention intention. Proceedings of the 2016 ACM SIGMIS Conference on Computers and People Research 2016.

13. Thatcher JB, Stepina LP, Boyle RJ. Turnover of information technology workers: Examining empirically the influence of attitudes, job characteristics, and external markets. Journal of Management Information Systems 2002:19(3): 231-261.

14. Dattero R, Galup SD, Kan A, Quan J. It pays to be agile. Journal of Computer Information Systems 2017:57(3): 252-257.

15. Balaji S, Murugaiyan MS. Waterfall vs. V-model vs. Agile: A comparative study on sdlc. International Journal of Information Technology and Business Management 2012:2(26-30).

16. Valacich JS, George JF, Valacich JS. Modern systems analysis and design. 8 ed: Pearson Education.

17. Stoica M, Mircea M, Ghilic-Mico B. Software development: Agile vs. Traditional. Informatica Economica 2013:17(4): 64-76.

18. Dharmapal S, Sikamani KT. Applying lean on agile scrum development methodology. Compusoft 2014:3(3): 633.

19. Conboy K. Agility from first principles: Reconstructing the concept of agility in information systems development. Information Systems Research 2009:20(3): 329-354.

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

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

22. Manifesto for agile software development twelve principles of agile software

23. Hoda R, Murugesan LK. Multi-level agile project management challenges: A self-organizing team perspective. Journal of Systems and Software 2016:117(245-257).

24. Zhang Z, Arvela M, Berki E, Muhonen M, Nummenmaa J, Poranen T. Towards lightweight requirements documentation. Journal of Software Engineering and Applications 2010:3(9): 882-889.

25. Boehm B, Turner R. Balancing agility and discipline: Evaluating and integrating agile and plan-driven methods. International Conference on Software Engineering; Edinburgh, UK: IEEE; 2004.

26. Kuhrmann M, Diebold P, Münch J, Tell P, Garousi V, Felderer M, Trektere K, McCaffery F, Linssen O, Hanser E et al. Hybrid software and system development in practice: Waterfall, scrum, and beyond. Proceedings of the 2017 International Conference on Software and System Process; 2017.

27. Misra SC, Kumar V, Kumar U. Identifying some important success factors in adopting agile software development practices. Journal of Systems and Software 2009:82(11): 1869-1890.

28. Sidky A, Arthur J, Bohner S. A disciplined approach to adopting agile practices: The agile adoption framework. Innovations in Systems and Software Engineering 2007:3(3): 203-216.

29. Sureshchandra K, Shrinivasavadhani J. Adopting agile in distributed development. In. Adopting agile in distributed development. IEEE International Conference on Global Software Engineering: IEEE; 2008.

30. Wellington CA, Briggs T, Girard CD. Comparison of student experiences with plan-driven and agile methodologies. Proceedings, Frontiers in Education 35th Annual Conference 2005.

31. Ilieva S, Ivanov P, Stefanova E. Analyses of an agile methodology implementation. In. Analyses of an agile methodology implementation. 30th EUROMICRO Conference: IEEE Computer Society; 2004.

32. Capiluppi A, Fernandez-Ramil J, Higman J, Sharp HC, Smith N. An empirical study of the evolution of an agile-developed software system. 29th International Conference on Software Engineering (ICSE'07): IEEE; 2007.

33. Joseph D, Ng KY, Koh C, Ang S. Turnover of information technology professionals: A narrative review, meta-analytic structural equation modeling, and model development. MIS Quarterly 2007:31(3): 547-577.

34. Dysvik A, Kuvaas B. Intrinsic motivation as a moderator on the relationship between perceived job autonomy and work performance. European Journal of Work and Organizational Psychology 2011:20(3).

35. Kähkönen T. Agile methods for large organizations - building communities of practice. Proceedings of the Agile Development Conference; 2004.

36. Hummel M, Rosenkranz C, Holten R. The role of communication in agile systems development: An analysis of the state of the art. Business and Information Systems Engineering 2013:5(5): 343-355.

37. Ghobadi S, Mathiassen L. Risks to effective knowledge sharing in agile software teams: A model for assessing and mitigating risks. Information Systems Journal 2017:27(6): 699-731.

38. Turk D, France R, Rumpe B. Limitations of agile software processes. Proceedigns of the 3rd International Conference on eXtreme Programming Agile Processes in Software Engineering 2014.

39. Alzoubi YI, Gill AQ, Al-Ani A. Distributed agile development communication: An agile architecture driven framework. Journal of Software 2015:10(6): 681-694.

40. Paasivaara M, Lassenius C, Heikkilä VT. Inter-team coordination in large-scale globally distributed scrum: Do scrum-of-scrums really work? Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement: ACM; 2012.

41. Singh A, Taneja A, Mangalaraj G. Creating online surveys: Some wisdom from the trenches tutorial. IEEE Transactions on Professional Communication 2009:52(2): 197-212.

42. Ang S, Slaughter S, Yee Ng K. Human capital and institutional determinants of information technology compensation: Modeling multilevel and cross-level interactions. Management Science 2002:48(11): 1427-1445.

43. Joseph D, Boh WF, Ang S, Slaughter SA. The career paths less (or more) traveled: A sequence analysis of it career histories, mobility patterns, and career success. MIS Quarterly 2012:36(2): 427-452.

44. Baron RM, Kenny DA. The moderator–mediator variable distinction in social psychological research: Conceptual, strategic, and statistical considerations. Journal of Personality and Social Psychology 1986:51(6): 1173-1182.

45. Hicks R, Tingley D. Causal mediation analysis. Stata Journal 2011:11(4): 1-15.

46. Imai K, Keele L, Tingley D. A general approach to causal mediation analysis. Psychological Methods 2010:15(4): 309-334.

47. Igbaria M, Meredith G, Smith DC. Predictors of intention of is professionals to stay with the organization in south africa. Information and Management 1994:26(5): 245-256.

48. Spector PE, Jex SM. Relations of job characteristics from multiple data sources with employee affect, absence, turnover intentions, and health. Journal of Applied Psychology 1991:76(1): 46-53.

49. Grant AM. Relational job design and the motivation to make a prosocial difference. In. Relational job design and the motivation to make a prosocial difference. Academy of Management Review; 2007.

50. Tripp JF, Armstrong DJ. Agile methodologies: Organizational adoption motives, tailoring, and performance. Journal of Computer Information Systems 2018:17(4): 267-307.

51. Nerur S, Mahapatra R, Mangalaraj G. Challenges of migrating to agile methodologies. Communications of the ACM 2005:48(5): 72-78.

52. Fitzgerald B. Systems development methodologies: The problem of tenses. Information Technology & People 2000:13

53. Dybå T, Dingsøyr T. Empirical studies of agile software development: A systematic review. Information and Software Technology 2008:50(9-10): 833-859.

Funding

This study was partially funded by the Ministry of Education (Singapore) under Tier 1 Grant Number 2017-T1-001-255-0 (RG63/17).

Figure 1: The Theoretical Model

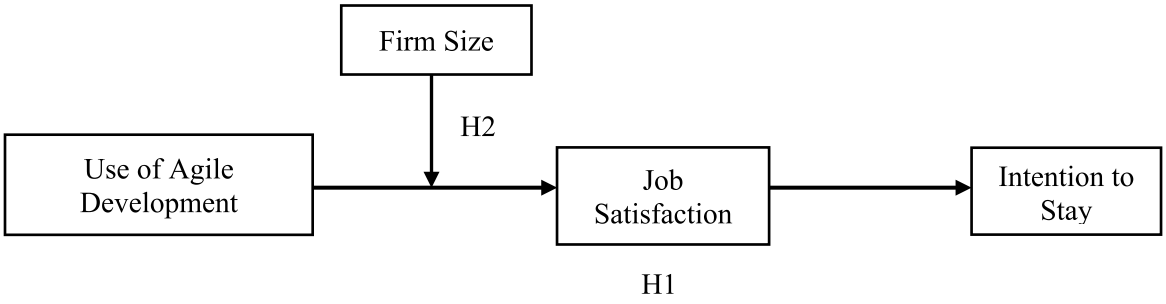

Figure 2 Mediation Effects

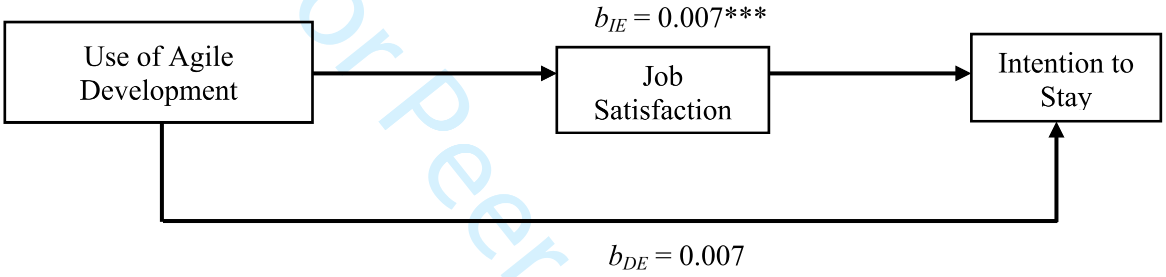

Table 1: IT Jobs, Proportions of Respondents and Educational Level

	N	No formal Education (%)	High School Diploma (%)	Some College Education (%)	Associate Degree (%)	Bachelor's Degree (%)	Postgraduat e Degree (%)
Developers	28,797	0.2	5.0	12.4	3.3	52.1	27.0
Managers	642	0.0	0.0	13.6	1.7	50.0	34.7
Systems administrators	93	1.1	6.4	15.0	6.5	52.7	18.3
Systems analysts	2,857	0.1	3.5	9.1	2.9	40.5	43.9

Table 2: Variables and Corresponding Definitions

Variable	Definition
Dependent variable	
Intention to Stay	Cognitive measure of respondents' intention to stay with the current employer
Independent variable	
Use of Agile Development	Whether agile methods are used for software production in the respondents' firm, with non-agile methods as the reference
Mediator variable	
Job Satisfaction	Affective attachment to a job
Moderator variable	
Firm Size	Measure of employer size
Controls	
Gender	The reported gender identity of respondent
Ethnicity	The reported ethnic affiliation of respondent
Education Level	Level of schooling
IT experience	Tenure in IT profession
Job role	Job type in the IT profession
Salary	Natural logarithmic measure of annual salary (USD)
Computing Workload	Hours per day spent on work computer

Table 3 Results of the Logistic Regression Model

Dependent Variable: Intention to Stay					
	Marginal effect (*b*)	se	z	95%CI	
Use of Agile Development	0.011**	0.005	2.223	0.001	0.021
Male	0.003	0.015	0.205	-0.027	0.033
Female	-0.006	0.017	-0.386	-0.039	0.026
Caucasian	0.029***	0.009	3.410	0.013	0.046
Black	-0.009	0.018	-0.476	-0.045	0.027
Hispanic	0.032**	0.012	2.664	0.008	0.055
Native American	0.020	0.026	0.752	-0.032	0.071
Middle Eastern	0.005	0.015	0.322	-0.024	0.034
Education Level	-0.002	0.002	-1.340	-0.005	0.001
IT Experience	0.002**	0.001	2.435	0.000	0.004
Systems Analyst	-0.010	0.006	-1.580	-0.022	0.002
Manager	-0.010	0.013	-0.791	-0.036	0.015
Systems Administrator	0.032	0.026	1.220	-0.019	0.083
Log Salary	0.019***	0.005	3.800	0.009	0.028
Computing Workload	-0.005*	0.003	-2.062	-0.011	0.000
Country Fixed Effects		Yes			
N		32,389			
Pseudo R Sq.		0.163			
Log Likelihood (*df*)		-10,651.020 (162)			

* $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$

Table 4 Correlations and Descriptive Statistics

		Mean	SD	1	2	3	4	5	6	7	8	9	10
1	Intention to Stay	0.893	0.309										
2	Use of Agile Development	0.677	0.468	0.020***									
3	Job Satisfaction	3.602	1.098	0.325***	0.042***								
4	Firm Size	4.219	2.145	-0.006	0.072***	-0.020***							
5	Gender: Male	0.927	-	0.006	0.005	-0.001	0.001						
6	Female	0.013	-	-0.008	-0.003	0.003	-0.002	-0.896***					
7	Other	0.060	-	0.003	-0.003	-0.005	0.003	-0.417***	-0.029***				
8	Ethnicity: White	0.744	-	0.133***	0.011*	0.096***	0.016**	0.016**	-0.019***	0.002			
9	Black	0.019	-	-0.022***	0.007	-0.020***	-0.013*	-0.029***	0.029***	0.005	-0.198***		
10	Asian	0.138	-	-0.114***	-0.004	-0.062***	0.023***	-0.016**	0.023***	-0.011*	-0.571***	-0.051***	
11	Hispanic	0.062	-	-0.004	0.024***	-0.015**	-0.009	0.006	-0.004	-0.006	-0.366***	-0.033***	-0.094***
12	Native American	0.004	-	0.001	0.003	0.001	0.000	-0.007	0.002	0.010	-0.102***	-0.009	-0.026***
13	Middle Eastern	0.033	-	-0.020***	-0.013*	-0.027***	-0.041***	0.001	-0.003	0.003	-0.262***	-0.023***	-0.068***
14	Education Level	3.884	1.127	-0.018***	0.027***	0.001	0.105***	-0.033***	0.047***	-0.022***	-0.061***	-0.009	0.086***
15	IT Experience	3.034	2.150	0.057***	0.066***	0.023***	0.076***	0.083***	-0.083***	-0.017**	0.162***	-0.042***	-0.143***
16	Job Role: Developer	0.889	-	0.004	0.132***	-0.021***	-0.032***	0.028***	-0.025***	-0.012*	-0.028***	0.001	0.020***
17	Systems Analyst	0.089	-	-0.008	-0.142***	0.012*	0.046***	-0.038***	0.034***	0.016**	0.027***	-0.008	-0.019***
18	Manager	0.020	-	0.001	0.023***	0.022***	-0.046***	0.013*	-0.010	-0.008	0.002	0.008	-0.005
19	Systems Administrator	0.002	-	0.003	-0.057***	-0.006	0.021***	0.006	-0.009	0.003	0.007	0.014**	-0.004
20	*Log* Salary	4.706	0.501	0.117***	0.111***	0.096***	0.172***	0.003	-0.008	0.009	0.339***	-0.038***	-0.248***
21	Computing Workload	2.830	0.660	-0.018***	0.007	-0.025***	-0.053***	0.017**	-0.031***	0.026***	-0.028***	0.013*	0.024***

* $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$

Table 4 (Continued)

		12	13	14	15	16	17	18	19	20
1	Intention to Stay									
2	Use of Agile Development									
3	Job Satisfaction									
4	Firm Size									
5	Gender: Male									
6	Female									
7	Other									
8	Ethnicity: White									
9	Black									
10	Asian									
11	Hispanic									
12	Native American									
13	Middle Eastern	-0.012*								
14	Education Level	-0.023***	0.004							
15	IT Experience	0.011*	-0.029***	-0.004						
16	Job Role: Developer	-0.011*	0.008	-0.070***	-0.001					
17	Systems Analyst	0.016**	-0.005	0.071***	-0.045***	-0.854***				
18	Manager	-0.004	-0.005	0.010	0.092***	-0.392***	-0.044***			
19	Systems Administrator	-0.004	0.000	-0.014**	-0.015**	-0.145***	-0.016**	-0.007		
20	*Log* Salary	0.029***	-0.068***	0.039***	0.354***	-0.033***	0.004	0.058***	-0.007	
21	Computing Workload	0.009	0.016**	-0.067***	-0.052***	0.029***	-0.025***	-0.011*	-0.001	-0.074***

* $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$

Table 5 Results of the Mediation

	b	95%CI	
Total Effect (TE)	0.014***	0.006	0.020
Direct Effect (DE)	0.007	-0.002	0.010
Indirect Effect (IE)	0.007***	0.005	0.010

* $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$

Table 6 Results of the Moderated Mediation Analysis

	Level	Conditional indirect effect	95%CI	
Firm Size	Large	0.006*	0.001	0.010
	Small	0.010***	0.006	0.010
H3	$\Delta b_{\text{IE (Large – Small)}}$	-0.004*	-0.011	-0.001

* $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$; Results are based on 1000 simulations using the quasi-Bayesian Monte Carlo method.

UCIS-2020-0030
When Agile Means Staying: A Moderated Mediated Mode
Responses to Review Comments

Responses to Editor-in-Chief's Comments

Based on a thorough blind review and my own assessment, your manuscript entitled "When Agile Means Staying: A Moderated Mediated Model" has been conditionally accepted for publication in the Journal of Computer Information Systems. Please know that your manuscript is designated as requiring "major revision". Therefore, final publication decision depends upon successfully revising your manuscript to address the reviewer(s)' comments. Furthermore, your revised manuscript will be reviewed and determined whether it requires another round of revision.

When you revise your manuscript please highlight the changes you make in the manuscript by using the track changes mode in MS Word or by using bold or colored text.

Once again, thank you for submitting your manuscript to Journal of Computer Information Systems and I look forward to receiving your revision.

Response

Thank you very much for conditionally accepting our manuscript. We take this opportunity to thank you and the review team for the excellent review comments that have greatly improved the quality of the manuscript. In this round of review, we enacted the following suggestions:

1. Agile, Traditional and Hybrid Approaches: We provided a balanced review of agile, traditional and hybrid approaches to developing software.
2. Large vs small organizations and Agile: We strengthened the theoretical arguments leading to our hypothesis that firm size moderates the mediated relationship between the use of agile development and intention to stay via job satisfaction.
3. Future research: We called on future research to examine the outcomes of hybrid approaches.
4. Editorial:
   a. Updated the citations to be current.
   b. Corrected spelling and grammatical errors
   c. The document has been professional edited.
   d. Adhered to the author guidelines including the selecting the appropriate reference style and maximum word count.

We do hope that these revisions have met with the review team's expectations. Once again, thank you for the excellent guidance on this manuscript.

Responses to Reviewer 1's Comments

R1.1

While I love the premise of your paper and think your research questions are vital, you need to take another step and ensure that you have bracketed out your bias because as I read this paper I immediately concluded, "These authors love agile," which is great, until I started to conclude as I read it, "These authors came into this research with a forgone conclusion that agile is the only way to go." I get that you've bought in to agile and I applaud that, but you need to assume a more objective stance when mounting a research study.

Response

Thank you for highlighting this issue and allowing us the opportunity to clarify our thinking and writing. It is not our intention to position agile as a silver bullet. In the current version of the manuscript we provide a more balanced narrative of agile methodology by highlighting its strengths and weaknesses.

R1.2

Now, I must confess I am a PMI method wonk and a 26 year veteran of projects, programs and portfolios delivering large business intelligence and analytics systems to clients, but internally and in a consulting role. I don't share this to tout what I'm about, I share this because it reinforces my first paragraph. For instance, I have seen agile work well in large organizations so your conclusion that agile only works in small organizations is unfounded. Sure, your point about requiring additional infrastructure is a valid one, as is your point affirming that agile is designed with collocation in mind. That said - and whether you intended it or not - your emphasis on small companies appears biased.

Response

Thank you for your comments. In the current version of the manuscript, we acknowledge that agile practices work well in large organizations. Nonetheless, we offer a more nuanced explanation, based on the current literature, of how the implementation and use of agile practices pose a greater challenge to larger organization relative to smaller ones. This challenge revolves around the potential risk of communication breakdown in larger, globally distributed teams that are commonly found in larger organizations. Our revised manuscript (page 11) now reads as follows:

> "While larger organizations have adapted agile methods to include hybrid models[25], the general consensus in the software development literature is that the implementation of pure agile practices is often challenging for larger organizations.[35] Larger organizations tend to work on complex IT projects by deploying multiple teams that must engage in cross-function activities. The larger number of teams adds complexity in managing team–team and team–client interactions.[36] This complexity becomes even more challenging when deploying globally distributed software teams, as is commonly done by larger organizations.
>
> We argue that in larger organizations, cross-team communication and interaction—a critical requirement of agile practices—is difficult to achieve, although not impossible.[37,]

> [38] Agile teams rely heavily on in-person or face-to-face conversations during the development process.[36] While use of videoconferencing technologies has been advocated to improve communication within collocated agile teams[39], these technologies may not be as effective as in-person meetings. There is an increased risk of communication breakdowns in these software teams (and with their clients) when agile methods are used in larger organizations.[40]"

R1.3
Then, while I give you kudos for mentioning predictive/waterfall models, you appear to dismiss them and incorrectly assert that agile approaches are free wheeling and lack structure.

Response
Thank you for this comment. We have revised the current manuscript to highlight the value propositions offered by the predictive/waterfall model. We argue that the waterfall model adds predictability to the development process and enables developers maintain a robust project scope. We have added more details on the advantages of using the traditional plan-driven approaches in following manner (pg. 5)

> "A value proposition of the traditional plan-driven approach is that it allows the development team to maintain a more detailed and robust project scope.[16] As a result, developers adhere to stringent development procedures. Another benefit of using the traditional approach is that, due to its extensive documentation requirement, the approach is well-suited to large projects that are joined by new individuals at different points in the development lifecycle.[15] New team members can readily refer to the documentation to get up to speed with the development project."

We agree that agile approaches are not entirely freewheeling and lack structure. According to the agile manifesto the core value of agile is individuals and interactions over people and tools. Agile methods are designed to use minimum documentation in order to facilitate responsiveness to changing user needs. This implies less planning and more flexibility is required in agile projects than in traditional plan-driven project management (Serrador and Pinto 2015). In the revised manuscript we convey this difference in planning and flexibility between agile and traditional approaches on page 6:

> "Relative to the traditional plan-driven approach, agile development is a fluid and self-adaptive approach to developing software.[17]"

> "In contrast to the plan-driven approach, agile development emphasizes individuals rather than processes and tools.[20]"

R1.4
As I am sure you will agree, the only way to be effective in delivering on agile projects is to freeze scope at the beginning of a sprint or you're sunk. Moreover, again looking at this through my "PMI lens," senior executives could care less whether or not you're using agile, waterfall or hybrid approaches as long as you're delivering.

Response

Thank you for this comment. We agree that it is common practice to apply lean thinking to agile development. The lack of "scope freeze" in agile is considered a common challenge in Agile Scrum methodology. A lean solution proposed for this challenge is to freeze scope at the beginning of a sprint or scrum. This is essential to deliver the product faster. In the revised manuscript we note this point on page 6:

> "Lean principles are sometimes incorporated into agile practices to facilitate code and functional freezes.[18]"

In regard to your comment about senior executives, the choice of using a particular methodology is organization and project-specific. While the use of traditional approaches of development continues to dominate in the industry, a 2019 Gartner survey of senior executives found that "the majority of organizations use or plan to use agile approaches on the business side and for implementing enterprise software" (Gartner 2019, pg. 6).

R1.5

Which brings me to the concept of hybrid projects. Folks, its not a binary. We don't have to choose agile over waterfall or vice versa. Hybrid approaches allow us to mix and match between the two (or more). So for instance, I used to develop large data warehouses with my teams (now, of course, re-branded as data lakes) where the back ends were more logically developed using a waterfall approach to minimize change and risk while the front end dashboards and reports were developed with agile methods. You should acknowledge this to back off the bias that appears, perhaps by accident, to be present.

Response

Thank you for this suggestion. In the current version, we acknowledge the existence of the hybrid models in the following manner the following pages:

On p. 8

> "To address the challenges associated with adopting a specific development approach, teams and organizations adopt hybrid approaches by combining well-structured development processes and flexible agile practices.[25] The variety of hybrid models follow a pattern in which the traditional plan-driven approach serves as the framework that incorporates agile practices.[26] Hybrid approaches are often neither planned nor designed but are a result of a natural evolution of different development practices that are known to work.[26]"

On p. 11:

> "While larger organizations have adapted agile methods to include hybrid models[25], the general consensus in the software development literature is that the implementation of pure agile practices is often challenging for larger organizations.[35]"

On p. 21:

> "Rather, software developers employ a variety of development methods, including hybrid models of agile and traditional approaches. Our measure of the use of agile development, which includes Extreme programming, Scrum, Kanban, Lean, Pair, and Mob programming, effectively captures agility in the development process.[16] Together, these practices fulfill the core tenets of the agile manifesto. Future research may replicate the findings of the present research using a more refined conceptualization and measurement of the use of agile development as well as hybrid models."

R1.6
Case in point is your networking example. I've installed labs, server farms and networks. You appear to question the reality that the average networking project is done using a waterfall approach. Its logical to do that, again to minimize risk. Consider a server farm installation, for example. Each server has the same configuration, is installed the same way and is networked in identical fashion. That's waterfall, folks.

Response
Thank you for the feedback. We have revised the networking example to reflect the reality that the different networking lifecycle models (e.g., Plan-Design-Manage and Prepare-Plan-Design-Implement-Operate-Optimize) follow the waterfall approach. The revised manuscript includes that following narrative on page 19:

> "Our study underscores the need to examine how specific work methods in other IT domains impact job attitudes and perceptions. For example, the network development lifecycle specifies the process of building computer networks. There are several different network lifecycle models (e.g., Plan-Build-Manage, Prepare-Plan-Design-Implement-Operate-Optimise) that follow the waterfall approach. How do the differences in the lifecycle models lend themselves to determining individual work attitudes and perceptions? This question deserves scholarly attention in order to further our understanding of designing IT-specific jobs."

R1.7
Its clear to me that one or more of the authors work for a small IT consulting firm. I did too several times in my day and I rejoice in that. That said, again, you need to be careful to bracket out your bias. Here's my suggestion for adjusting your theoretical model so that it is more neutral and still moves the field forward: switch your independent variable and your mediator variable so that job satisfaction is your independent variable and your mediator variable is agile development. If you think about it, in your context intention to stay is dependent on job satisfaction, not on whether or not developers are using agile.

Response
Thank you for suggesting the alternative model. The theoretical arguments explicating our research model is that the use of agile development practices is associated with higher levels of job satisfaction which then prompts intentions to stay.

We specified the alternative model (Job Satisfaction – Agile Development – Intention to Stay) and rerun the analysis. First, we regressed Use of Agile Development on Job Satisfaction and performed the mediator analysis. The results of the regression model are shown in Table R1. The results of the alternative model indicate an R sq. of 0.026 compared to the research model's R sq. of 0.163 (Table 3). This indicates that our research model has a higher explanatory power than the alternative model, suggesting a better fit of the data.

Table R1: Results of the alternative model

Use of Agile Development			
	Marginal Effect (*b*)	se	z
Job Satisfaction	0.007**	0.003	2.333
Male	0.005	0.017	0.294
Female	-0.010	0.016	-0.625
Caucasian	0.025	0.020	1.250
Black	-0.007	0.007	-1.000
Hispanic	0.022	0.015	1.467
Native American	0.020	0.023	0.870
Middle Eastern	0.025	0.021	1.191
Education Level	-0.005	0.004	-1.250
IT Experience	0.019***	0.004	4.750
Systems Analyst	-0.009	0.020	-0.450
Manager	-0.007	0.015	-0.467
Systems Administrator	0.043*	0.022	1.955
Log Salary	0.025	0.049	0.510
Computing Workload	-0.006	0.006	-1.000
Country Fixed Effects	Yes		
N	32,389		
Pseudo R Sq.	0.026		
Log Likelihood (*df*)	-17,634.992 (162)		

*** $p < 0.001$; ** $p < 0.01$; * $p < 0.05$

R1.8
This opens up possibilities for you for subsequent research as well because you can say in your section about future research that you can assess predictive (waterfall) and hybrid approaches in subsequent research.

Response
Thank you for this suggestion. In the revised manuscript, we call for future research to assess the developer outcomes of hybrid models in the following manner (page 21):

> "Second, our measure of the use of agile development is not perfect. Rigorous adherence to a specific software development method is low in the developer community.[52] Rather, software developers employ a variety of development methods, including hybrid models of agile and traditional approaches. Our measure of the use of agile development, which includes Extreme programming, Scrum, Kanban, Lean, Pair, and Mob programming, effectively captures agility in the development process.[16] Together, these practices fulfill the core tenets of the agile manifesto. Future research may replicate the findings of the present research using a more refined conceptualization and measurement of the use of agile development as well as hybrid models."

R1.9
After mentioning all this, I don't think you need to do a lot with this paper. Simply make more mention of the reality that some projects benefit from waterfall and hybrid approaches, change the model to reflect that without changing your findings, and back off the bias against large companies and toward agile.

Response
Thank you for your inputs that has caused us to revise the current manuscript to reflect the reality that some projects benefit from waterfall and hybrid approaches. On page 8, under the section Software Development Approaches: Traditional and Agile Approaches, we have included a paragraph about hybrid approaches (on p. 8).

> "To address the challenges associated with adopting a specific development approach, teams and organizations adopt hybrid approaches by combining well-structured development processes and flexible agile practices.[25] The variety of hybrid models follow a pattern in which the traditional plan-driven approach serves as the framework that incorporates agile practices.[26] Hybrid approaches are often neither planned nor designed but are a result of a natural evolution of different development practices that are known to work.[26]"

In addition, we argue that agile practices can be successfully implemented in large organization but the implementation of agile practices in large organization in comparison to smaller organizations come with its unique challenges. We argue that in larger organizations, where larger distributed teams build complex systems, there is the potential risk of communication breakdown when agile practices are adopted. We explain this potential risk on page 11 as follows:

> "We argue that in larger organizations, cross-team communication and interaction—a critical requirement of agile practices—is difficult to achieve, although not impossible.[37, 38] Agile teams rely heavily on in-person or face-to-face conversations during the development process.[36] While use of videoconferencing technologies has been advocated to improve communication within collocated agile teams[39], these technologies may not be as effective as in-person meetings. There is an increased risk of communication breakdowns in these software teams (and with their clients) when agile methods are used in larger organizations.[40]"

R1.11

Notice this in no way adds more incremental work for you, it just gives you an air of objectivity that is lacking in your current iteration. Don't change your analysis. Do remove your bias and open up your model for additional research. The result will be even more powerful than it currently is, and I am happy to tell you what you have here is definitely positive and powerful.

Response

Thank you for your comments and review of our manuscript which has caused us to revise the current manuscript to provide a balanced narrative on agile, predictive and hybrid models of development. We hope the revised manuscript allays your concerns.

Responses to Reviewer 2's Comments

R2.1
Writing Style: The topic of this paper is a relevant one. However, it should be improved on readability with more focus on sentence structure, punctuation, and grammar. Several issues with clarity, sentence structure, awkward phrasing.

Response
Thank you for your feedback. The revised manuscript has been professionally edited.

R2.2
Grammar/usage/mechanics: This paper could have been well served with an edit focused on sentence structure, subject-verb agreement, punctuation to name few

Response
Thank you for your feedback. The revised manuscript has been professionally edited.

R2.3
The content could have been richer by going into some more analytic depth of explanation of the detection of duplicated data while presenting some empirical studies that used this framework.

Response
Thank you for this comment. In the revised document we explain how the survey design tackles the issue of duplicated data, which could pose some validity issues to the data. On page 12, we explain that:

> "Stack Overflow contacted the entire population of its registered members via email to ask them to respond to the survey. The email invitations contained unique links to minimize the possibility of respondents submitting multiple responses.[41]"

R2.4
The references were somewhat old. I only spotted few recent ones.

Response
Thank you for your comment. We have updated the citations to be current.

R2.5
The sampling technique was not clear too

Response
Thank you for your feedback about the sampling technique. Stack Overflow uses a total population sampling technique for its surveys. This technique is a purposive sampling approach that involves examining the entire population of interest. The population of interest for the survey are the registered members of the Stack Overflow platform. In the revised document, we clarify the sampling technique on page 12:

> "Stack Overflow contacted the entire population of its registered members via email to ask them to respond to the survey."

Additional References

Gartner (2019). Results Summary: Agile in the Enterprise. Retrieved from https://circle.gartner.com/Portals/2/Resources/pdf/Agile%20in%20the%20Enterprise%202019%20-%20Results%20Summary%20(updated).pdf

Serrador, P., & Pinto, J. K. (2015). Does Agile work?—A quantitative analysis of agile project success. *International Journal of Project Management*, *33*(5), 1040-1051.