When agile means staying : a moderated mediated model

Setor, Tenace Kwaku; Joseph, Damien

2020

Setor, T. K. & Joseph, D. (2020). When agile means staying : a moderated mediated model.
Journal of Computer Information Systems.
<https://dx.doi.org/10.1080/08874417.2020.1759160>

<https://hdl.handle.net/10356/153990>

<https://doi.org/10.1080/08874417.2020.1759160>

© 2020 International Association for Computer Information Systems. All rights reserved.
This paper was published in Journal of Computer Information Systems] and is made
available with permission of nternational Association for Computer Information Systems.

Downloaded on 26 Apr 2023 17:45:01 SGT

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 (Beck & Andres, 2005; Dyba & Dingsoyr, 2009). 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 (Mathieu et al., 2019).

In practice, the successful redesign of software development methods is predicated on the affective experiences that IT professionals derive from applying the new methods (Mannaro et al., 2004). Work redesign can alter the psychological states of employees, prompt internally motivated work behaviors, and affect employee satisfaction (Peiró et al., 2020; Tims et al., 2016). 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 (Hong et al., 2011). While a substantial body of work (e.g., Dyba & Dingsoyr, 2009; Mannaro et al., 2004; Tessem & Maurer, 2007; Tripp et al., 2016) has found that the use of agile development increases IT professionals' *satisfaction with their jobs*, 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, including

perceptions of employee mobility, considering that 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 (Hong et al., 2011).

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 (Hackman & Oldham, 1976), which delineates the relationships between core dimensions of work and psychological outcomes. 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 (e.g., Dyba & Dingsoyr, 2009; Mannaro et al., 2004; Tessem & Maurer, 2007; Tripp et al., 2016). 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 (Barlow et al., 2011; Kähkönen, 2004). 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 (Barlow et al., 2011). 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 (1976) 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 (Koh & Joseph, 2016; Thatcher et al., 2002; Tripp et al., 2016).

---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 (Hackman & Oldham, 1976). 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)” (Hackman & Oldham, 1976, 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 (Dattero et al., 2017). This focus adds predictability to the development process. The traditional approach mandates extensive documentation of requirements and

training materials (Balaji & Murugaiyan, 2012). 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 (Valacich et al., 2017). 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 (Balaji & Murugaiyan, 2012). 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 (Valacich et al., 2017). 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 (Stoica et al., 2013). At the core of agile development are the mechanisms of iterative and incremental development (Dattero et al., 2017). 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 (Dharmapal & Sikamani, 2014). 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 (Conboy, 2009). 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 (Dattero et al., 2017; Lee & Xia, 2010). 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 (Lee & Xia, 2010). Agile development challenges individual developers to learn and acquire a variety of skills that are different from their preexisting stock of skills (Lee & Xia, 2010).

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 (Beck & Andres, 2005; Maruping et al., 2009). Decision-making power is decentralized and transferred to the developers who carry out the actual tasks. The IT project management literature (Beck & Andres, 2005; Lee & Xia, 2010) supports the notion that agile development offers considerable latitude to software teams in making decisions. 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 (Beck et al., 2001). Customers are considered an integral part of the development team throughout the entire development process (Dattero et al., 2017). Compared to

the traditional plan-driven approach, agile methods enforce customer involvement (Beck & Andres, 2005).

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 (Dattero et al., 2017; Tripp et al., 2016; Valacich et al., 2017).

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 (Hoda & Murugesan, 2016). 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 (Zhang et al., 2010). 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 (Hoda & Murugesan, 2016).

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 (Boehm & Turner, 2004). The variety of hybrid models follow a pattern in which the traditional plan-driven approach serves as the framework that incorporates agile practices (Kuhrmann et al., 2017). 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 (Kuhrmann et al., 2017).

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 (Misra et al., 2009; Sidky et al., 2007; Sureshchandra & Shrinivasavadhani, 2008). 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 (Dyba & Dingsoyr, 2009; Mannaro et al., 2004; Tripp et al., 2016). Tripp et al. (2016) 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 (Tripp et al., 2016). 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. (2005) 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 (e.g., Ilieva et al., 2004; Maruping et al., 2009) that have sampled IT professionals corroborate Wellington et al.'s 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 (Ilieva et al., 2004), and Maruping et al. (2009) 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. (2007) 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 (Tripp et al., 2016).

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 (Joseph et al., 2007; Thatcher et al., 2002). IT professionals leave their organizations when their managers fail to design meaningful jobs (Koh & Joseph, 2016). This is particularly the case for many software jobs. Drawing inspiration from job characteristics theory (Hackman & Oldham, 1976), we argue that the skill variety, job autonomy, and feedback that characterize agile development cause IT professionals to experience positive feelings about their work (Tripp et al., 2016), 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 (Tessem & Maurer, 2007). 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 (Montesa García et al., 2019).

Agile development practices prescribe that developers choose their own methods to perform their work, make independent work-related decisions, and schedule their work tasks (Lee & Xia, 2010). These independent work-related and scheduling decisions increase perceptions of job autonomy (Dysvik & Kuvaas, 2011; Tims et al., 2016). 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 (Tessem & Maurer, 2007). 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 (Tripp et al., 2016).

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 (Valacich et al., 2017), 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 (Boehm & Turner, 2004), the general consensus in the software development literature is that the implementation of pure agile practices is often challenging for larger organizations (Kähkönen, 2004). 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 (Hummel et al., 2013). 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 (Ghobadi & Mathiassen, 2017; Turk et al., 2014). Agile teams rely heavily on in-person or face-to-face conversations during the development process (Hummel et al., 2013). While use of videoconferencing technologies has been advocated to improve communication within collocated agile teams (Alzoubi et al., 2015), 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 (Paasivaara et al., 2012). 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 (Singh et al., 2009). 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., Ang et al., 2002), 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 (Joseph et al., 2012). 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 (Baron & Kenny, 1986). 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 ($b = 0.011, p < 0.01, 95\% \text{ CI } [0.010, 0.021]$). 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 (2011) approach. Standard procedures of mediation analysis (e.g., Baron & Kenny, 1986) 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 (Imai et al., 2010). Following Imai et al.'s (2010) recommendation, we employ Hicks and Tingley's (2011) 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" (Hicks & Tingley, 2011, 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 ($b_{TE} = 0.014, p < 0.001$, 95% CI [0.006, 0.020]) as well as a significant indirect effect ($b_{IE} = 0.007, p < 0.001$, 95% CI [0.005, 0.010]).

--- Insert Table 5 About Here ---

The direct effect, however, is not significantly different from zero ($b_{DE} = 0.007, n.s.$, 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 (Δb_{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. (2002) and Igbaria et al. (1994) 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 (1991) 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 (2007) 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 (Igbaria et al., 1994; Thatcher et al., 2002). 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 (Tripp & Armstrong, 2018). 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 (Nerur et al., 2005). 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 (Fitzgerald,

2000). 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 (Valacich et al., 2017). 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 (Lee & Xia, 2010; Tripp & Armstrong, 2018). However, empirical research has shown that the use of agile development practices influences individual IT professional outcomes as well (Dybå & Dingsøyr, 2008; Tripp et al., 2016). 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.

Funding

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

References

- Alzoubi, Y. I., Gill, A. Q., & Al-Ani, A. (2015). Distributed agile development communication: An agile architecture driven framework. *Journal of Software*, 10(6), 681–694. <https://doi.org/10.17706/jsw.10.6.681-694>
- Ang, S., Slaughter, S., & Yee Ng, K. (2002). Human capital and institutional determinants of information technology compensation: Modeling multilevel and cross-level interactions. *Management Science*, 48(11), 1427–1445. <https://doi.org/10.1287/mnsc.48.11.1427.264>
- Balaji, S., & Murugaiyan, M. S. (2012). Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. *International Journal of Information Technology and Business Management*, 2(1), 26–30. <https://pdfs.semanticscholar.org/8d2a/ad803781a34a43af447cea410cd710b9f1f5.pdf>
- Barlow, J., Giboney, J., Keith, M. J., Wilson, D., Schuetzler, R., Lowry, P. B., & Vance, A. (2011). Overview and guidance on agile development in large organizations. *Communications of the Association for Information Systems*, 29(2), 25–44. <https://doi.org/10.2139/ssrn.1909431>
- 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. <https://doi.org/10.1037/0022-3514.51.6.1173>
- Beck, K., & Andres, C. (2005). *Extreme programming explained: Embrace change*. Addison-Wesley.
- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). *Manifesto for agile software development: Twelve principles of agile software*. <https://agilemanifesto.org/principles.html>
- Boehm, B., & Turner, R. (2004). Balancing agility and discipline: Evaluating and integrating agile and plan-driven methods. In *Proceedings, 26th International Conference on Software Engineering* (pp. 718–719). IEEE. <https://doi.org/10.1109/icse.2004.1317503>
- Capiluppi, A., Fernandez-Ramil, J., Higman, J., Sharp, H. C., & Smith, N. (2007). An empirical study of the evolution of an agile-developed software system. In *Proceedings, 29th International Conference on Software Engineering* (pp. 511–518). IEEE. <https://doi.org/10.1109/ICSE.2007.14>
- Conboy, K. (2009). Agility from first principles: Reconstructing the concept of agility in information systems development. *Information Systems Research*, 20(3), 329–354. <https://doi.org/10.1287/isre.1090.0236>
- Dattero, R., Galup, S. D., Kan, A., & Quan, J. (2017). It pays to be agile. *Journal of Computer Information Systems*, 57(3), 252–257. <https://doi.org/10.1080/14659891.2016.1184025>
- Dharmapal, S., & Sikamani, K. T. (2014). Applying lean on agile scrum development methodology. *Compusoft*, 3(3), 633.
- Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9–10), 833–859. <https://doi.org/10.1016/J.INFSOF.2008.01.006>
- Dyba, T., & Dingsøyr, T. (2009). What do we know about agile software development? *IEEE Software*, 26(5), 6–9. <https://doi.org/10.1109/MS.2009.145>

- Dysvik, A., & Kuvaas, B. (2011). Intrinsic motivation as a moderator on the relationship between perceived job autonomy and work performance. *European Journal of Work and Organizational Psychology*, 20(3), 367–387.
<https://doi.org/10.1080/13594321003590630>
- Fitzgerald, B. (2000). Systems development methodologies: The problem of tenses. *Information Technology & People*, 13(3), 174–185. <https://doi.org/10.1108/09593840010377617>
- Ghobadi, S., & Mathiassen, L. (2017). Risks to effective knowledge sharing in agile software teams: A model for assessing and mitigating risks. *Information Systems Journal*, 27(6), 699–731.
<https://doi.org/10.1111/isj.12117>
- Grant, A. M. (2007). Relational job design and the motivation to make a prosocial difference. *Academy of Management Review*, 32(2), 393–417. <https://doi.org/10.5465/amr.2007.24351328>
- Hackman, J. R., & Oldham, G. R. (1976). Motivation through the design of work: Test of a theory. *Organizational Behavior and Human Performance*, 16(2), 250–279. [https://doi.org/10.1016/0030-5073\(76\)90016-7](https://doi.org/10.1016/0030-5073(76)90016-7)
- Hicks, R., & Tingley, D. (2011). Causal mediation analysis. *The Stata Journal*, 11(4), 1–15.
<https://doi.org/10.1177/1536867X1201100407>
- Hoda, R., & Murugesan, L. K. (2016). Multi-level agile project management challenges: A self-organizing team perspective. *Journal of Systems and Software*, 117, 245–257.
<https://doi.org/10.1016/j.jss.2016.02.049>
- Hong, W., Thong, J. Y. L., Chasalow, L. C., & Dhillon, G. (2011). User acceptance of agile information systems: A model and empirical test. *Journal of Management Information Systems*, 28(1), 235–272.
<https://doi.org/10.2753/MIS0742-1222280108>
- Hummel, M., Rosenkranz, C., & Holten, R. (2013). The role of communication in agile systems development: An analysis of the state of the art. *Business and Information Systems Engineering*, 5(5), 343–355. <https://doi.org/10.1007/s12599-013-0282-4>
- Igbaria, M., Meredith, G., & Smith, D. C. (1994). Predictors of intention of IS professionals to stay with the organization in South Africa. *Information & Management*, 26(5), 245–256.
[https://doi.org/10.1016/0378-7206\(94\)90063-9](https://doi.org/10.1016/0378-7206(94)90063-9)
- Ilieva, S., Ivanov, P., & Stefanova, E. (2004). Analyses of an agile methodology implementation. In *Proceedings of the 30th EUROMICRO Conference* (326–333). IEEE Computer Society. <https://doi.org/10.1109/EURMIC.2004.1333387>
- Imai, K., Keele, L., & Tingley, D. (2010). A general approach to causal mediation analysis. *Psychological Methods*, 15(4), 309–334. <https://doi.org/10.1037/a0020761>
- Joseph, D., Boh, W. F., Ang, S., & Slaughter, S. A. (2012). The career paths less (or more) traveled: A sequence analysis of IT career histories, mobility patterns, and career success. *MIS Quarterly*, 36(2), 427–452. <https://doi.org/10.2307/41703462>
- Joseph, D., Ng, K.-Y., Koh, C., & Ang, S. (2007). Turnover of information technology professionals: A narrative review, meta-analytic structural equation modeling, and model development. *MIS Quarterly*, 31(3), 547–577. <https://doi.org/10.2307/25148807>
- Kähkönen, T. (2004). Agile methods for large organizations: Building communities of practice. In *Proceedings of the Agile Development Conference* (2–10). IEEE.
<https://doi.org/10.1109/adevc.2004.4>
- Koh, C., & Joseph, D. (2016). Experienced meaningfulness and calling. In *SIGMIS-CPR '16: Proceedings of the 2016 ACM SIGMIS Conference on Computers and People Research* (97–103). ACM. <https://doi.org/10.1145/2890602.2890615>
- Kuhrmann, M., Diebold, P., Münch, J., Tell, P., Garousi, V., Felderer, M., Trektore, K., McCaffery, F., Linssen, O., Hanser, E. & Prause, C.R. (2017). Hybrid software and system development in practice: waterfall, scrum, and beyond. In *Proceedings of the 2017 International Conference on Software and System Process* (30–39). <https://doi.org/10.1145/3084100.3084104>
- Lee, G., & Xia, W. (2010). An integrated analysis of quantitative and qualitative field data on software development agility. *MIS Quarterly*, 34(1), 87–114. <https://doi.org/10.2333.243.235>

- Mannaro, K., Melis, M., & Marchesi, M. (2004). Empirical analysis on the satisfaction of IT employees comparing XP practices with other software development methodologies. In J. Eckstein & H. Baumeister (Eds.), *Extreme programming and agile processes in software engineering* (166–174). Springer. https://doi.org/10.1007/978-3-540-24853-8_19
- 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. <https://doi.org/10.1287/isre.1090.0238>
- Mathieu, M., Eschleman, K. J., & Cheng, D. (2019). Meta-analytic and multiwave comparison of emotional support and instrumental support in the workplace. *Journal of Occupational Health Psychology*, 24(3), 387–409. <https://doi.org/10.1037/ocp0000135>
- Misra, S. C., Kumar, V., & Kumar, U. (2009). Identifying some important success factors in adopting agile software development practices. *Journal of Systems and Software*, 82(11), 1869–1890. <https://doi.org/10.1016/j.jss.2009.05.052>
- Montesa García, D., Molina, I. R., & Marco, J. M. (2019). The effect of job characteristics on intrinsic and extrinsic job satisfaction: The moderating role of psychological capital, 1. *Psychologica*, 62(1). https://doi.org/10.14195/1647-8606_62-1_2
- Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, 48(5), 72–78. <https://doi.org/10.1145/1060710.1060712>
- Paasivaara, M., Lassenius, C., & Heikkilä, V. T. (2012). Inter-team coordination in large-scale globally distributed Scrum. *Proceedings of the ACM–IEEE International Symposium on Empirical Software Engineering and Measurement* (235–238). ACM. <https://doi.org/10.1145/2372251.2372294>
- Peiró, J. M., Bayona, J. A., Caballer, A., & Di Fabio, A. (2020). Importance of work characteristics affects job performance: The mediating role of individual dispositions on the work design–performance relationships. *Personality and Individual Differences*, 157, 109808. <https://doi.org/10.1016/j.paid.2019.109808>
- Sidky, A., Balci, O., Bohner, S., Pérez-Quinones, M. A., & Wallace, L. (2007). A structured approach to adopting agile practices: The agile adoption framework. *Innovations in Systems and Software Engineering*, 3(3), 203–216. <https://doi.org/10.1007/s11334-007-0026-z>
- Singh, A., Taneja, A., & Mangalaraj, G. (2009). Creating online surveys: Some wisdom from the trenches tutorial. *IEEE Transactions on Professional Communication*, 52(2), 197–212. <https://doi.org/10.1109/TPC.2009.2017986>
- Spector, P., & Jex, S. (1991). Relations of job characteristics from multiple data sources with employee affect, absence, turnover intentions, and health. *Journal of Applied Psychology*, 76(1), 46–53. <https://doi.org/10.1037/0021-9010.76.1.46>
- Stoica, M., Mircea, M., & Ghilic-Micu, B. (2013). Software development: Agile vs. traditional. *Informatica Economică*, 17(4), 64–76. <https://doi.org/10.12948/issn14531305/17.4.2013.06>
- Sureshchandra, K., & Shrinivasavadhani, J. (2008). Adopting agile in distributed development. In *IEEE International Conference on Global Software Engineering* (217–221). IEEE. <https://doi.org/10.1109/ICGSE.2008.25>
- Tessem, B., & Maurer, F. (2007). Job satisfaction and motivation in a large agile team. In G. Concas, E. Damiani, M. Scotto, & G. Succi (Eds.), *Agile Processes in Software Engineering and Extreme Programming* (54–61). Springer. https://doi.org/10.1007/978-3-540-73101-6_8
- Thatcher, J., Stepina, L., & Boyle, R. (2002). Turnover of information technology workers: Examining empirically the influence of attitudes, job characteristics, and external markets. *Journal of Management Information Systems*, 19(3), 231–261. <https://doi.org/10.1080/07421222.2002.11045736>
- Tims, M., Derks, D., & Bakker, A. B. (2016). Job crafting and its relationships with person–job fit and meaningfulness: A three-wave study. *Journal of Vocational Behavior*, 92, 44–53. <https://doi.org/10.1016/j.jvb.2015.11.007>
- Tripp, J. F., & Armstrong, D. J. (2018). Agile methodologies: Organizational adoption motives, tailoring, and performance. *Journal of Computer Information Systems*, 58(2), 170–179.

<https://doi.org/10.1080/08874417.2016.1220240>

- Tripp, J. F., Riemenschneider, C., & Thatcher, J. B. (2016). Job satisfaction in agile development teams: Agile development as work redesign. *Journal of the Association for Information Systems*, 17(4), 267–307. <https://doi.org/10.17705/1jais.00426>
- Turk, D., France, R., & Rumpe, B. (2002). Limitations of agile software processes. In *Proceedings of the 3rd International Conference on eXtreme Programming Agile Processes in Software Engineering* (43–46). Publisher.
- Valacich, J. S., George, J. F., & Valacich, J. S. (2017). *Modern systems analysis and design*. Pearson Education.
- Wellington, C. A., Briggs, T., & Girard, C. D. (2005). Comparison of student experiences with plan-driven and agile methodologies. In *Proceedings, Frontiers in Education 35th Annual Conference* (T3G-18–T3G-23). IEEE. <https://doi.org/10.1109/FIE.2005.1611951>
- Zhang, Z., Arvela, M., Berki, E., Muhonen, M., Nummenmaa, J., & Poranen, T. (2010). Towards lightweight requirements documentation. *Journal of Software Engineering and Applications*, 3(9), 882–889. <https://doi.org/10.4236/jsea.2010.39103>