Journal of the Association for Information Systems

Volume 23 | Issue 6 Article 7

2022

Too Drained from Being Agile? The Self-Regulatory Effects of the Use of Agile ISD Practices and their Consequences on Turnover Intention

Lea Mueller
Technical University of Darmstadt, mueller@ise.tu-darmstadt.de

Alexander Benlian
Technical University of Darmstadt, benlian@ise.tu-darmstadt.de

Journal of the Association for Information Systems (2022) **23**(6), 1420-1455
doi: 10.17705/1jais.00766

RESEARCH ARTICLE

ISSN 1536-9323

Too Drained from Being Agile? The Self-Regulatory Effects of the Use of Agile ISD Practices and their Consequences on Turnover Intention

Lea Mueller[1], Alexander Benlian[2]

[1]Technical University of Darmstadt, Germany, mueller@ise.tu-darmstadt.de
[2]Technical University of Darmstadt, Germany, benlian@ise.tu-darmstadt.de

Abstract

While much is known about the beneficial effects of agile information systems development (ISD), scholars have largely neglected to address its potential downsides. Specifically, research on this topic has thus far overlooked the ambivalent implications of the specific demands placed on developers working in agile ISD teams, including potentially depleting effects. Drawing on ego depletion theory and the associated literature, we provide a more balanced view and introduce self-regulatory resource depletion triggered by using agile ISD practices—encompassing software development (SD) and project management (PM) practices—as a theoretical perspective on why agile developers experience different levels of work-related fatigue that lead to stronger or weaker turnover intentions. Furthermore, we propose that due to the specific way in which agile ISD methods organize ISD project work, developers' perceived workload influences the intensity by which agile ISD practices affect self-regulatory resources and developers' feelings of fatigue. We examined our research model using a multimethod approach including quantitative and qualitative data. We found that the use of agile SD practices enhances developers' self-regulatory resources and reduces fatigue and turnover intention. Our results also show that perceived workload strengthens the energizing effects of the use of agile SD practices and reveals a depleting effect of the use of agile PM practices, with countervailing implications for turnover intention. This study contributes to agile ISD literature by drawing a more nuanced and balanced picture with both resource-enhancing and resource-draining effects of the use of agile ISD practices. Finally, we give managerial advice regarding factors to consider when designing and managing agile ISD projects.

Keywords: Agile Information Systems Development, Agile Software Development, Agile Project Management, Ego Depletion, Self-Regulation, Fatigue, Perceived Workload, Turnover Intention

Rajiv Sabherwal was the accepting senior editor. This research article was submitted on September 3, 2020 and underwent three revisions.

1 Introduction

Agile methods are omnipresent in contemporary information systems development (ISD). They enable companies to respond quickly to changing conditions caused by technology-centric, turbulent, and uncertain markets (Ramasubbu et al., 2015; Rigby et al., 2018). To date, 97% of companies claim to use agile approaches and while their adoption rate has increased in many business areas, they remain most popular in ISD (VersionOne, 2019). The Agile Manifesto is the common denominator for all agile ISD methods and highlights an iterative, people-centric development approach with self-organizing teams subject to continuous feedback (Fowler & Highsmith, 2001). The various agile ISD methods that have emerged from these basic agile principles, such as Scrum and extreme programming (XP), not only fundamentally influence daily development work but also place various demands on developers, such as short cycle times and permanent responsiveness to change, exercising constant pressure to deliver software increments (Benlian, 2022). The potential consequences of these demands, work exhaustion or fatigue in particular, can constitute essential antecedents to developers' intentions to stay in or leave a company (e.g., Moore, 2000a). In times of IT skill shortages, in which developer churn may come with enormous costs, understanding the demands and reasons for staying or leaving can help organizations retain IT professionals (Pflügler et al., 2018).

Over more than two decades, academic literature has studied the implications of agile ISD methods and their practices (Hoda et al., 2017). Consistent with previous research, we apply the term "practices" to specific agile techniques, such as daily standups and continuous integration, and "methods" to any number of defined, interdependent sets of practices developed by practitioners, such as Scrum or XP (Tripp et al., 2016). Despite the substantive body of work, research has mainly focused on the organizational implementation of agile ISD practices examining method tailoring (e.g., Fitzgerald et al., 2003) and project-level implications, such as the benefits of agile ISD practices for product quality (e.g., Maruping et al., 2009). Even though these insights are valuable because they explain how organizational and project demands can be addressed through the use of agile ISD practices, our understanding of how these practices and their demands influence developers in their daily work is still fragmented (Tripp et al., 2016; Venkatesh et al., 2020). The extant research has generated inconsistent findings concerning the effects of demands inherent in using agile ISD practices, indicating ambivalent implications for individual developers, including positive as well as unintended negative consequences. For example, the empowering nature of agile ISD practices was found to be motivating for developers but may also lead to ineffective decision-making, likely producing stress (Drury et al., 2012; McAvoy & Butler, 2009; McHugh et al., 2011). Similarly, increased communication and knowledge sharing within agile ISD teams facilitates work alignment yet also requires a high level of self-discipline to overcome communication barriers (Ghobadi & Mathiassen, 2016; Hummel et al., 2013).

Given this ambivalence, it is surprising to find that individual-level effects of the use of agile ISD practices only recently started to attract increasing research attention (e.g., Benlian, 2022; Tripp et al., 2016; Tuomivaara et al., 2017; Venkatesh et al., 2020). Tripp et al. (2016), for example, studied how agile software development (SD) practices and agile project management (PM) practices differentially shape developers' job satisfaction via specific job characteristics. The mixed results reported in this study—in the sense that agile SD and PM practices have substantially different direct and indirect effects on job satisfaction—suggest that distinguishing between distinct types of practices seems to be an essential criterion for studying job outcomes in agile ISD, that their use might not have exclusively beneficial effects on developers, and that other factors besides job characteristics seem to be important determinants of job satisfaction.

One such factor could be the effects of the demands placed by the use of agile ISD practices on developers' levels of exhaustion or fatigue as an important prerequisite for job outcomes, such as satisfaction and turnover intention (Moore, 2000a; Rutner et al., 2008). Previous research on the work exhaustion of developers in agile ISD teams addresses the mitigation potential of the use of agile SD practices regarding exhaustion by reducing role ambiguity and role conflict (Venkatesh et al., 2020). Although these results provide important preliminary insights, their focus on agile SD practices leaves aside agile PM practices that are widely used and more people-centric, placing different demands on developers than IS artifact-centric agile SD practices (Baham & Hirschheim, 2022). Integrating agile PM practices in the research on developer exhaustion or fatigue in agile ISD teams may thus bring forth new and potentially diverging mechanisms, which could then influence significant job outcomes.

In light of these research gaps, the objective of this study is to shed light on how the use of agile SD and PM practices—the primary and most widely used categories of agile ISD practices (Baham & Hirschheim, 2022; Tripp et al., 2016)—affects developer fatigue and ultimately turnover intention. We draw on ego depletion theory and the notion of self-regulatory resources to examine how and why using agile ISD practices may invoke both individual benefits and costs for developers, and, more specifically, if developers may need to invest more or less of their personal self-regulatory resources to deal with the implications of agile ISD work structures. This self-regulation view provides a balanced perspective on the underlying psychological mechanisms explaining the positive and energizing consequences of the use of agile ISD practices as well as its unintended negative and depleting consequences.

Given that previous research has repeatedly indicated that workload influences resource-level demands of work (e.g., Diestel & Schmidt, 2012; Prem et al., 2016) and agile proponents emphasize the promotion of sustainable development through a constant and healthy work pace (Fowler & Highsmith, 2001), our study also accounts for whether perceived workload may shape how intensely developers experience the self-regulation effects of the use of agile ISD practices. Tuomivaara et al. (2017) recently showed how agile ISD practices can reduce developers' work exhaustion by leveling out their workload across the entire duration of a project. However, their study does not systematically examine perceived workload as a boundary condition in agile ISD contexts. As a typically scarce resource, developers are frequently exposed to substantial amounts of workload in projects (Huarng, 2001; Moore, 2000a). Because organizations are rarely engaging in exclusively agile work, developers are often part of multiple projects and, even if they are working on one project only, dependencies and the extra effort created by other projects or (functional) departments may severely influence developers' daily workload (Hekkala et al., 2017; Laux & Kranz, 2019), leaving developers with a workload they might not be able to balance.

Failure to fully understand the influences of the use of agile ISD practices on developer fatigue and turnover intention and ignoring the role of perceived workload in this context limits our ability to obtain a more comprehensive picture of how to manage agile ISD projects. With the aim of providing a nuanced and more balanced view of the use of agile ISD practices and its energizing, i.e., resource-enhancing, as well as fatiguing, i.e., resource-depleting, implications for developers, we propose the following overarching research questions:

RQ1: Does the use of agile ISD practices have resource-enhancing and -depleting effects? If so, how do these effects influence developers' turnover intentions?

RQ2: How does developers' perceived workload affect the resource-enhancing and -depleting effects of the use of agile ISD practices?

In our study, we address these research questions by drawing upon ego depletion theory (EDT) and integrating the concept of self-regulation into agile ISD research. We pursued a multimethod approach wherein quantitative and qualitative data were collected and interpreted to better understand the perceptions of developers working in agile ISD teams. In a first study, we conducted a field survey to analyze the impact of the use of agile SD and PM practices on developers' availability of self-regulatory resources, reflected in different levels of work-related fatigue. We examined the role of work-related fatigue as a possible explanation for how agile SD and PM practices affect developers' turnover intentions. Finally, we investigated how developers' perceived workload shaped the effects of agile SD and PM practices on fatigue and turnover intention. In a second study, given the need to enrich and deepen the knowledge on the use of agile ISD practices (Maruping et al., 2009) and in line with our goal to provide a nuanced and comprehensive understanding, we collected qualitative data to explore the antecedents of developers' feelings of fatigue triggered by the use of agile ISD practices in more detail. We conducted interviews with developers working in agile ISD teams, probing how they feel when using agile ISD methods in the context of normal and high workloads, with the aim of corroborating and expanding our findings from the survey study.

Our study results contribute to research on agile ISD, IT workforce, and work-related ego depletion. By examining resource-enhancing and resource-depleting mechanisms, we move beyond the predominant notions of the use of agile ISD practices as a largely positive phenomenon that enhances work satisfaction and mitigates exhaustion and thus provide a more balanced view on the consequences of using agile ISD practices for individual developers. In response to previous calls for research (Tripp et al., 2016; Venkatesh et al., 2020), we also extend the knowledge on developers' psychological conditions in agile ISD projects by unpacking key processes that underlie the effects of using agile ISD practices on feelings of fatigue. Moreover, complementing prior work on the project level of analysis (Tuomivaara et al., 2017), we introduce perceived workload as a still underinvestigated moderating factor that offers an important boundary condition to the individual-level implications of the use of agile ISD practices. Finally, we contribute to research on the work-based implications of ego depletion (Johnson et al., 2018) by identifying agile ISD practices as specific IT job-design features capable of reducing work-related fatigue. From a practitioner's point of view, our results provide guidance on how to enhance developers' self-regulatory resources and well-being at work and strengthen a company's position in attracting and retaining software development talents.

2 Theoretical Background

2.1 Agile Software Development and Project Management Practices

Agile ISD has been on the rise for almost two decades now. The basic principles are written down in the Agile Manifesto—setting agile boundaries reaching from "individuals and interactions over processes and tools," working product over comprehensive documentation," "customer collaboration over contract negotiation," to "responding to change over following a plan" (Fowler & Highsmith, 2001). Coming from these principles, various agile ISD methods have emerged that contain a wide range of practices. In accordance with previous research (Tripp et al., 2016), we refer to agile ISD "practices" as specific techniques, such as continuous integration, and to agile ISD "methods" as a particular interdependent set of practices, such as Scrum. In our research we focus on agile ISD practices because very few agile practitioners actually adopt all of the practices in a method (Conboy & Fitzgerald, 2010) and many adopt practices from multiple methods

(VersionOne, 2019). At a high level, these agile ISD practices can be distinguished according to their focus on SD and PM (Baham & Hirschheim, 2022; Hummel et al., 2013; Tripp et al., 2016).

Agile SD practices focus on more IS artifact-centric tasks that primarily guide the execution of the software development process itself, emphasizing technical aspects and automating mechanisms (Baham & Hirschheim, 2022). The most popular of these practices stems from XP (Kude et al., 2019; VersionOne, 2019). When implementing unit testing, for example, developers use dedicated test code to (automatically) test the effects of changes to the system. Coding standards provide a set of established norms concerning code-naming and consistency (Beck, 2000), creating a frame for efficiently designing and programming software. Appendix A provides an overview and description of the practices that are currently primarily applied (Tripp et al., 2016; VersionOne, 2019). Next to unit tests and coding standards, continuous integration and refactoring are at the top of this list for agile SD practices.

Agile PM practices relate to more people-centered tasks and focus on facilitating the performance of the actual software development by managing work in teams, establishing customer relationships, and obtaining feedback (Baham & Hirschheim, 2022; Tripp et al., 2016). Scrum has become particularly popular for project management as well as creative teamwork in solving complex problems (Barlow et al., 2011; Rigby et al., 2016). The most important means of agile PM practices are meetings which are preferably held face-to-face. For example, in daily standups, project progress and current tasks are discussed, while in retrospectives, the development team critically reflects on the last iteration and identifies improvement opportunities (Schwaber & Beedle, 2002). Besides daily standups and retrospectives, an iterative delivery approach, including iteration planning and iteration reviews, is among the agile PM practices that are primarily applied in practice (see Appendix A).

2.2 Related Literature on the Effects of the Use of Agile ISD Practices

The consequences and effects of agile ISD have been studied at the organizational, project, team, and individual levels. In the years following the publication of the Agile Manifesto, the adoption of agile ISD methods in organizations, as well as combinations of agile and traditional plan-driven ISD approaches, was of major research interest (Boehm, 2002; Nerur, Mahapatra, & Mangalaraj, 2005; Vinekar et al., 2006). Following this, scholars began to extensively examine project- and team-level effects in agile ISD. The wide range of phenomena and outcomes studied at these levels includes project and team performance, product quality, decision-making, team communication, and knowledge sharing (Coyle et al., 2015; Drury et al., 2012; Hummel et al., 2015; Kude et al., 2019; Maruping et al., 2009a; Maruping et al., 2009b). Only recently has scientific research turned significant attention toward the individual-level implications of agile ISD. The focus of these studies has mainly been on job satisfaction and work exhaustion (Fortmann-Müller, 2018; Tripp et al., 2016; Tuomivaara et al., 2017; Venkatesh et al., 2020). For example, Tripp et al. (2016) conducted a study based on the job characteristics model and found support for the effect of the use of agile PM practices but not for the use of agile SD practices on developer job satisfaction.

Research on the people-oriented implications of the use of agile ISD practices shows ambivalent individual-level consequences for developers emphasizing beneficial effects, while also pointing to negative side effects. Collective decision-making within a self-managed team of developers, for instance, is considered a central advancement of agile ISD. It places decision power where the knowledge is and empowers each team member (Fowler & Highsmith, 2001; Tessem, 2014). While this empowerment has been found to be motivating for developers (McHugh et al., 2011), studies have also reported inefficiencies in decentralized team decision-making that are likely to produce stress for developers (Drury et al., 2012; McAvoy & Butler, 2009). Other examples include the increased communication requirements and the intensive knowledge sharing that, on the one hand, facilitate developers' alignment of work with other stakeholders due to higher transparency (Hummel et al., 2013; Schlauderer et al., 2015). However, on the other hand, developers must invest more time and effort into these communication activities (e.g., in meetings related to planning, grooming, aligning, reviewing), and may need a high-level of self-discipline to do so (Ghobadi & Mathiassen, 2016; Schlauderer et al., 2015). Also, as an important practice of agile ISD, collaborative code ownership contributes to higher product quality and fewer misunderstandings about code (Maruping et al., 2009a) but may also offer a considerable scope for conflicts among developers who may disagree on code structure and quality (Balijepally et al., 2006). These examples showcase that the use of agile ISD practices may be a double-edged sword, generating both benefits and detriments for developers. Yet scholars have rarely adopted a balanced perspective and have not sufficiently discussed potentially unintended negative individual-level effects (Benlian, 2022).

Notable exceptions are studies on work exhaustion and well-being. Tuomivaara et al. (2017) found that an agile PM approach can prevent exhausting phases by balancing the workload in ISD projects. Venkatesh et al. (2020) introduced individual developer skills in their discussion of why agile SD practices are more, or less, effective in reducing role conflict and ambiguity and, eventually, work exhaustion. While these studies provide important insights into the effects of the use of agile ISD practices on developers, they focus primarily on how agile ISD methods can mitigate potential individual-level downsides, emphasizing the generally positive lens through which the consequences of agile ISD method use are examined. Additionally, their results focus on a particular type of agile ISD practices. The study of Tripp et al. (2016), however, shows that different types of agile ISD practices, namely agile PM and SD practices, can potentially trigger divergent consequences for individual developers. Another recent study (Benlian, 2022) showed that agile ISD practices, in general, can simultaneously impair and improve developer well-being. Hence, while this study takes a more balanced view, it also lacks a distinction between types of agile practices, which is necessary to achieve a deeper understanding and illustrate effects on a more granular level. As a consequence, the previous agile ISD literature on work exhaustion and well-being has failed to provide a holistic and nuanced view on the individual-level effects of using agile ISD practices on developers. In sum, individual-level research on agile ISD is still scarce and lacks a balanced perspective that takes both positive and negative implications for job outcomes into account; thus, it misses out on a comprehensive discussion on how the use of agile ISD practices potentially differs in its effects on individual developers. In the following, we turn to an established theory, namely EDT, to build a more thorough and exhaustive understanding of how using agile ISD practices influences developers and their self-regulatory resources.

2.3 Self-Regulation and Feelings of Fatigue

EDT defines self-regulation as the act of exerting control over one's feelings, thoughts, or impulses and adapting behaviors based on various demands (e.g., Baumeister et al., 2006). For example, resisting distractions or coping with stress are demands that require a great deal of control over oneself (Muraven & Baumeister, 2000). According to Muraven et al. (1998), self-regulation relies on a limited pool of resources. When it is used up, one falls into a state of ego depletion, which describes a state of diminished availability of regulatory resources or self-control. Once it is depleted, subsequent tasks in need of self-regulation are less successful than they would be without prior depletion (Baumeister et al., 2006). The literature typically compares the strength of self-control to a muscle that gets exhausted when continuously exerted (e.g., Muraven & Baumeister, 2000). Indeed, fundamental studies of EDT found that performing regulatory tasks results in intensified feelings of fatigue in the sense of being mentally tired, drowsy, or exhausted (Baumeister et al., 1998; Muraven et al., 1998). However, there are circumstances in which regulatory resources can become less depleted or even replenished—in other words, the "muscle" is energized so that one experiences less fatigue. While rest or sleep obviously reduces fatigue and can replenish depleted resources (Muraven & Baumeister, 2000), positive experiences and mood, as well as motivation, have been shown to also counteract the regulatory depletion effect, increase resources, and reduce fatigue (Baumeister & Vohs, 2007; Bono et al., 2013; Gross et al., 2011; Tice et al., 2007). In addition, an environment that supports rather than controls autonomy can help to build regulatory resources through enhanced feelings of subjective vitality (Muraven et al., 2008).

In the organizational and IS literature, EDT has recently received increasing attention. Studies have drawn on EDT to explain the consequences of self-regulatory demands, hence fatigue, at the workplace or when using technology (e.g., Chan & Wan, 2012; Soror et al., 2015). Such studies have, for example, investigated the resource-draining effects of helping others at work (e.g., Lanaj et al., 2016) and the resource-replenishing effects of specific technology designs in online gaming (e.g., Lee et al., 2016). Given the arguments above, we believe that EDT has the potential to provide a solid foundation for investigating the potentially positive and negative effects of the use of agile ISD practices on individual developers' work-related fatigue.

3 Hypotheses Development

Figure 1 depicts our research model regarding feelings of fatigue and turnover intentions of developers working in agile ISD teams. We draw on EDT to theorize that the use of agile SD and PM practices is associated with lower and higher levels of fatigue, respectively, and that fatigue is a mediator in the relationship between the use of agile SD and PM practices and developers' turnover intentions. We additionally expect perceived workload to moderate the relationships between the use of agile SD and PM practices and developers' feelings of fatigue, in both cases intensifying the main effects in the context of high levels of perceived workload. The development and testing of this (first-stage) moderated mediation model is discussed as follows.

3.1 Agile SD Practices and Self-Regulatory Resources

Two key values of agile SD practices are feedback and simplicity in the process of programming and implementing software (Fitzgerald et al., 2006). They aim for smoothing processes, enhancing understanding, facilitating collaboration, and reducing the level of friction, thereby lowering stress, cognitive demands, and negative feelings among developers.

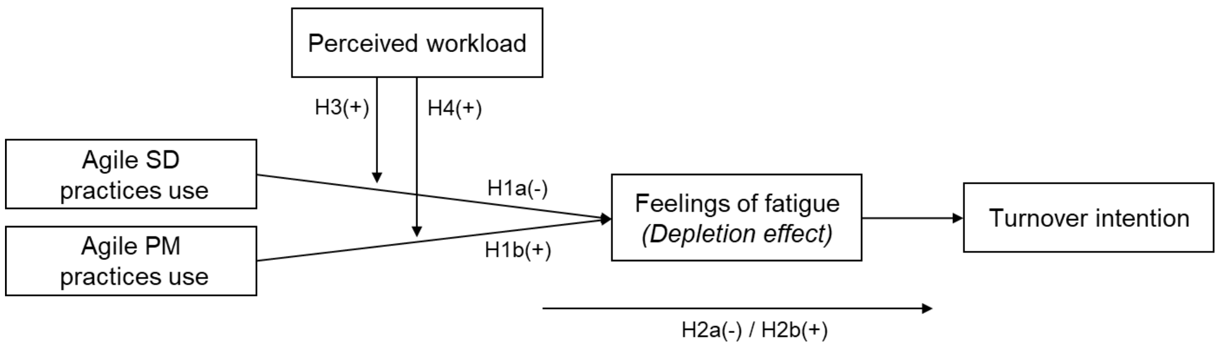

Figure 1. Research Model

In line with Venkatesh et al. (2020), who found that the use of agile SD practices reduces exhaustion through lower role ambiguity and conflict, we argue that as a result of the aforementioned characteristics of agile SD practices, developers are also less likely to feel fatigued due to lower consumption of self-regulation resources. Literature from the field of agile SD as well as ego depletion supports this logic.

A considerable part of feedback in agile SD is automatically generated in terms of failed unit and acceptance tests as well as continuous integration actions on a server machine that is jointly used by all team members (Tripp et al., 2016). The immediate and automated feedback is likely to lead to a seamless and undisturbed workflow conserving developers' internal resources because answering time, that is to say workflow interruption, is minimized (Baumeister et al., 2007; Beal et al., 2005). In addition, the high frequency of testing feedback on small pieces of code—as opposed to the infrequent testing of feedback including large amounts of code—can generate more positive feelings and higher motivation regarding testing because it is perceived as a series of manageable steps (Geister et al., 2006; Hazzan & Dubinsky, 2003). The result is a more comfortable atmosphere that is believed to counteract ego depletion (Baumeister & Vohs, 2007; Tice et al., 2007). Lastly, Ji et al. (2005) stated that the concurrent construction and debugging enabled by frequent feedback generally leads to a higher quality of the system and future development work. Developers likely need fewer resources to cognitively process programming information when the code quality is high (Schmeichel et al., 2003). They are also less exposed to difficult and stressful team situations, such as disagreement and friction, and need fewer regulatory resources to cope with demands (Muraven & Baumeister, 2000).

Applying coding standards and refactoring during software development enhances code clarity and simplicity, enabling a common understanding of software code among developers and thereby supporting effective coordination and collaboration (Beck, 2000; Maruping et al., 2009a; Pikkarainen et al., 2007). Developers' expectations of each other should become much clearer and ambiguity less likely (Venkatesh et al., 2020). Thus, cognitive processing, which is necessary for comprehending and editing code that was written by another developer, becomes easier. Facilitating cognitive processing has been found to build regulatory resources (Aspinwall, 1998; Schmeichel et al., 2003). Also, the possibility of coding errors and misunderstandings is reduced through coding standards, which diminishes the level of stress because team conflicts become less likely. Less stress to cope with, again, should preserve self-regulatory resources (Muraven & Baumeister, 2000). Finally, coding standards allow for a certain degree of behavioral automation in decision-making during the programming process, which is likewise consuming less regulatory resources (Webb & Sheeran, 2003).

In summary, research suggests that agile SD practices facilitate the cognitive processing necessary to work on others' code, create a comfortable atmosphere through motivation and positive feelings, and reduce stress levels and conflict potential because of the high quality of code that is generated. Studies in the field of psychology show that these favorable conditions counteract the resource depletion effect (e.g., Gross et al., 2011; Schmeichel et al., 2003; Tice et al., 2007; Webb & Sheeran, 2003). Thus, the use of agile SD practices holds considerable potential to alleviate resource depletion in developers, which would thus reduce their work fatigue.

H1a: The use of agile SD practices negatively influences developers' levels of fatigue.

3.2 Agile PM Practices and Self-Regulatory Resources

The agile PM practices that are primarily used according to VersionOne (2019) (see Appendix A) focus on short iterations and a variety of meetings during these iterations—for instance, daily standups or iteration planning meetings. Accordingly, agile developers are typically subject to constant time pressure and must deal with many interruptions. Since delivering on these daily demands requires discipline and vigilance, we argue that agile PM practices are likely to trigger resource-depleting effects.

Short cycle times, as proposed by the Agile Manifesto (Beck, 2000), place pressure on developers to frequently deliver software increments (Maruping et al., 2009a). (Agile) ISD in itself is a complex and cognitively demanding task, consuming a high amount of self-regulatory resources (Balijepall et al., 2015; Schmeichel et al., 2003). Feeling constantly pressured to test and deliver software increments would be expected to lead to substantial resource depletion (Muraven et al., 2008). In addition, meetings can present task-related hindrances that disrupt developers' progress on their individual work tasks related to programming software (Lanaj et al., 2016). Trying to stay focused and attentive when the number and frequency of meetings is as high as proposed by agile PM practices requires a high level of self-discipline, or self-regulation (Baumeister et al., 2007; Hagger et al., 2010; Schlauderer et al., 2015). Beyond their interruptive nature, meetings in agile ISD encompass processes that are exhausting. During retrospectives, for example, developers critically reflect on the last iteration and propose and implement possibilities for improvement (Schwaber & Beedle, 2002). Identifying and solving problems regarding task outcomes and processes consume regulatory resources, as these procedures require constant vigilance and contingency planning for both real and imagined problems (Lin & Johnson, 2015; Schmeichel et al., 2003). Moreover, communication can become less effective when there are no formal rules. For example, excessively brief descriptions of development problems (Coyle et al., 2015) may lead to a lack of understanding of the problems, potentially contributing to more intense and demanding problem-solving activities that require the consumption of self-regulatory resources (Schmeichel et al., 2003; Schmidt et al., 2012).

Taken together, the research indicates that when applying agile PM practices, developers need to withstand constant time pressure, muster a high level of self-discipline to stay focused, and be vigilant and careful to efficiently resolve interpersonal and process issues. These self-regulatory demands are likely to cause resource depletion (e.g., Baumeister et al., 2007; Muraven et al., 2008; Schmeichel et al., 2003). Hence, we posit that agile PM practices increase feelings of fatigue in developers.

H1b: The use of agile PM practices positively influences developers' levels of fatigue.

3.3 The Mediating Role of Work-Related Fatigue

The effect of work-related fatigue on turnover intention has received considerable research attention across disciplines. The predominant consensus is that fatigue is positively related to turnover intention (e.g., Cropanzano et al., 2003; Mor Barak et al., 2001; Wright & Cropanzano, 1998). Indeed, among the many implications of high levels of work-related fatigue, job turnover has been suggested as the first thing that most employees consider (Jackson et al., 1986; Leatz & Stolar, 1993; Moore, 2000b). Investigating this relationship explicitly in the context of ISD, IS scholars have confirmed the positive effects of work-related fatigue on turnover intention for IT professionals, both directly (Ahuja et al., 2007; Moore, 2000a) and indirectly, through mediators such as job satisfaction (Korunka et al., 2008; Rutner et al., 2008), organizational commitment (Ahuja et al., 2007), and psychological contracts (Moquin et al., 2019). In addition, results from studies examining IT professionals' intentions to change not only their job but also their profession portend to the positive effect of work-related fatigue (Armstrong et al., 2015; Joia & Mangia, 2017). In the majority of these studies, work-related fatigue in IT professionals has been proposed and identified as a mediating mechanism between stressors or demands at work due to specific job characteristics and employee turnover intention (e.g., Armstrong et al., 2015; Moore, 2000a; Rutner et al., 2008). This mediating role of fatigue has received further support from examinations of actual turnover, revealing that stressful work leads to employee turnover via work-related fatigue (De Croon et al., 2004; Taris et al., 2001). We follow these established arguments when proposing our mediation hypotheses.

As discussed above, higher levels of the use of agile SD practices can build regulatory resources by inducing positive feelings, reducing stress, and facilitating cognitive processing. Feeling less fatigued as a result of having more available resources will likely cause developers to think less about quitting their jobs. On the basis of the foregoing discussion, we posit that the use of agile SD practices reduces developers' turnover intentions because these practices help them enhance or replenish self-regulatory resources, leading to lower levels of fatigue.

H2a: Feelings of fatigue mediate the negative effect of the use of agile SD practices on developer turnover intention such that using agile SD practices reduces turnover intentions by negatively impacting feelings of fatigue.

In contrast, we have discussed how developers experience high levels of fatigue when using agile PM practices because daily development and collaboration activities drain their self-regulatory resources. Acknowledging previous empirical findings regarding the positive effect of fatigue on turnover intention, we suggest that the use of agile PM practices enhances developers' willingness to search for a new job because it consumes their regulatory resources and triggers fatigue.

H2b: Feelings of fatigue mediate the positive effect of the use of agile PM practices on developer turnover intention such that using agile PM practices increases turnover intentions by positively impacting feelings of fatigue.

3.4 The Role of Perceived Workload in Agile ISD

Agile ISD is usually organized into projects. Higher perceived workloads by individuals working in agile ISD teams can therefore be primarily considered in two ways: an individual takes part in various projects, which necessitates more frequent switching between tasks and topics, or projects get more intense in the sense that more tasks need to be accomplished in the same amount of time. The latter argument can certainly be challenged by agile proponents assuming that the workload in agile ISD projects ideally remains the same when agile ISD practices are used correctly and completely (Fowler & Highsmith, 2001; Tuomivaara et al., 2017). However, conditions are rarely ideal for agile ISD teams because they are often part of a larger organization that does not exclusively use agile methods, which creates dependencies and extra effort even if agile practices are faithfully used (Hekkala et al., 2017; Laux & Kranz, 2019; VersionOne, 2019). Additionally, the perception of workload is subjective and might vary even between agile ISD team members leading to different assessments of high and acceptable levels of workload.

Higher workloads, in general, are likely to produce stress and typically require attentional focus and higher cognitive processing, thereby demanding self-regulatory resources (e.g., Beal et al., 2005; Prem et al., 2016; Schmeichel et al., 2003). However, we believe that due to their specific way of organizing project work, agile ISD practices can have varying effects on developers' level of fatigue in high-workload situations. In particular, we believe that both the resource-enhancing effect of agile SD practices and the resource-draining effect of agile PM practices are intensified in such situations.

We have already suggested that the simplicity and automation potential inherent to agile SD practices have energizing effects on software developers (Beck, 2000; Fitzgerald et al., 2006; Schmeichel et al., 2003; Tripp et al., 2016; Webb & Sheeran, 2003). We additionally propose that the unique characteristics of agile SD practices reveal their full potential in high workload situations when time pressure is high or switching between projects becomes prevalent, reinforcing the enhancement of developers' self-regulatory resources. We make this argument because developers would be expected to benefit more from the use of agile SD practices when they perceive their workload to be high. In fact, we believe that developers do not fully realize the benefits of using agile SD practices until their perceived workload is high. For example, automated tests and clear coding standards exhibit automation advantages under high-workload conditions through enhanced consistency and simplicity. Because of these advantages, the effort to cope with additional development tasks is likely to increase only slightly. In addition, task and test automation can enable developers to speed up their programming work significantly when necessary and facilitate accomplishing a higher number of tasks in the same amount of time. Recognizing and appreciating the ability to increasingly exploit the efficiency provided by agile SD practices in high-workload situations may trigger positive feelings toward work in developers. Positive emotions directly counteract ego depletion and can therefore facilitate and accelerate the replenishing effect of using agile SD practices on effective self-control (Aspinwall, 1998; Tice et al., 2007). In addition, positive feelings may lead to higher levels of motivation at work (Isen & Reeve, 2005). Motivated developers are less likely to experience a depletion of self-regulatory resources (Baumeister & Vohs, 2007; Muraven & Slessareva, 2003). Hence, we propose that developers working on projects using agile SD practices can beneficially leverage the respective practices in high-workload situations and thereby substantially reduce their levels of fatigue.

H3: Perceived workload negatively moderates the relationship between the use of agile SD practices and developers' feelings of fatigue such that when the perceived workload is high (vs. low), developers experience fewer feelings of fatigue from using agile SD practices.

In contrast, for developers working on projects using agile PM practices, higher perceived workloads likely imply greater drain on regulatory resources due to more disruptions and substantial time pressure. Looking at the consequences of different workload levels in this context, we draw on findings from research on work events (e.g., Lanaj et al., 2016; Zohar et al., 2003). As mentioned earlier, agile PM practices largely focus on various meetings (e.g., daily standups, retrospectives) to provide feedback and improve work practices. These meetings place high social demands (e.g., due to intense face-to-face interactions) on developers and cannot be simply automatized or skipped in high-workload situations, thus increasing the likelihood that such meetings are experienced as disruptions to developers' daily work progress (Lanaj et al., 2016). When workloads are high, feelings of fatigue caused by disruptive events may thus intensify due to a limited capacity of regulatory resources (Beal et al., 2005; Zohar et al., 2003). In addition, when developers have to join more than one project using agile PM practices, the number of meetings they have to attend across projects increases substantially. As a result, they have less time to complete regular programming tasks, leading to severe time pressure—a situation that is likely to increase stress and deplete self-regulatory resources (Prem et al., 2016). Accordingly, we propose that in situations of high perceived workload, the resource-draining effect of using agile PM practices becomes stronger.

H4: Perceived workload positively moderates the relationship between the use of agile PM practices and developers' feelings of fatigue such that when the perceived workload is high (vs. low), developers experience more feelings of fatigue from using agile PM practices.

4 Research Approach

To examine the hypotheses proposed above, we employed a multimethod approach using two independent studies. The first was a cross-sectional field survey among agile ISD professionals in the United States to test our proposed relationships with a representative sample and achieve generalizable results. We then collected qualitative data in semi-structured interviews with agile ISD professionals in Germany to unpack the relationships in detail with the aim of revealing what is actually occurring and why. Figure 2 depicts this research approach.

We chose this research design to fulfill two purposes of multimethod research: confirmation and expansion (Jick, 1979; Mingers, 2001; Venkatesh et al., 2013). First, conducting two studies with different methods and samples allowed for the triangulation of our findings regarding the core theoretical associations between agile SD and PM practices and levels of fatigue as well as the influence of the moderator, perceived workload. Hence, we were able to strengthen the validity of the inferences made in each study. Second, the qualitative study expanded the survey study by empirically uncovering the mechanisms underlying the self-regulation effects of agile ISD. We were thereby able to gain additional insight into the nature and causes of the hypothesized relationships in line with our aim of providing a more nuanced view of the individual-level consequences of using agile ISD practices.

5 Quantitative Field Survey Study

5.1 Sample and Data Collection

We recruited our survey sample using Empanel Online, a panel company that specializes in conducting internet-based surveys in a B2B context. Many studies have already shown that data can be successfully collected via panel companies (e.g., Tripp et al., 2016; Venkatesh et al., 2019; Wiener et al., in press). Our survey was conducted among ISD professionals in the United States. Empanel Online provided 761 potential respondents. In order to ensure that the participants work with agile ISD methods, a screening question at the beginning of the survey asked whether they currently work in ISD projects using agile practices to a large degree. The screening resulted in 396 suitable respondents.

We incorporated quality assurance questions throughout the survey to ensure participants' engagement—for example using attention checks such as "If you are paying attention, please select Daily." When these quality checks were improperly answered, respective participants were immediately disqualified and prevented from completing the survey. 228 respondents passed our quality assurance procedure. After removing incomplete data sets and cases in which participants gave the same answer to every question ignoring reverse formulations, the final sample consisted of 207 valid responses. We accounted for nonresponse bias by comparing early and late respondents (first and last 50) based on their sociodemographics (Armstrong & Overton, 1977). The means of each sample showed no significant difference ($p > 0.05$) suggesting that a nonresponse bias is unlikely to have had an effect on the study results.

Of the 207 respondents, 65.7% were male. This percentage is in line with the higher rate of men in computing occupations (Ashcraft et al., 2016). Most developers in this sample were between 31 and 55 years old (71.0%). The experience with agile ISD varied but peaked at 3-5 years (32.4%). Sample statistics are summarized in Appendix B.

5.2 Measurement

The items used in the survey are based on established scales from previous research (see Appendix B) and were measured with 7-point Likert scales. All latent constructs in our study were measured reflectively with multiple items. The use of agile ISD practices was measured based on items from Tripp et al. (2016). The authors' list was updated with respect to current statistics on the most widely used agile ISD practices (VersionOne, 2019), resulting in the following eight constructs to be measured: iterative delivery, daily standups, retrospectives, short iterations, unit testing, continuous integration, coding standards, and refactoring. We tested and validated these practices by conducting a small-sized pretest with agile ISD developers. Based on the results of this pretest and recommendations from the panel company, we shortened the scale from Tripp et al. (2016). All agile ISD practice items were measured on an agreement scale with "I don't know" as an eighth option. In line with Tripp et al. (2016), we modeled the variables for the use of agile ISD practices as reflective-formative second-order constructs composed of the respective practices (Hair et al., 2017).

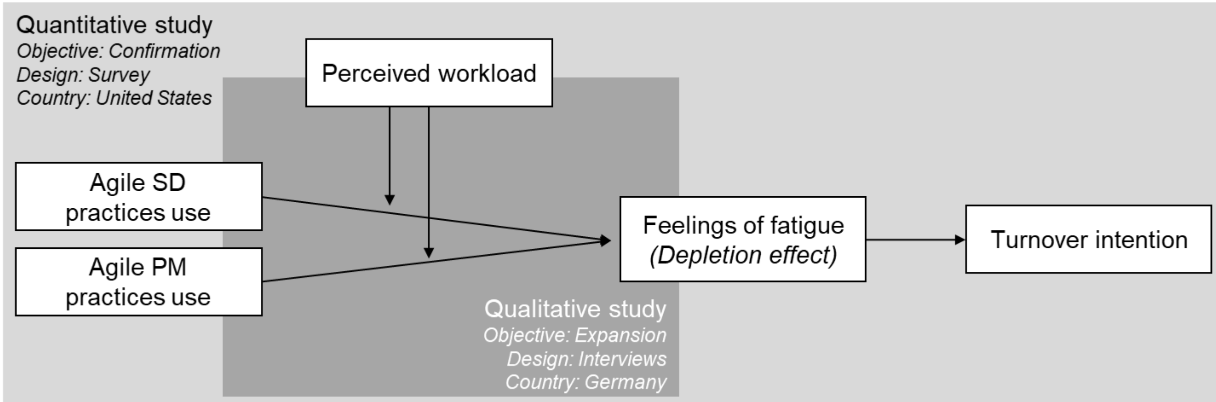

Figure 2. Multimethod Research Approach

The depletion effect was measured with five items adapted from van Yperen and Hagedoorn (2003) and Chan and Wan (2012) to capture the level of developers' fatigue specifically caused by conditions at work. In our survey instructions, we stressed that participants should answer questions concerning feelings of fatigue in the context of agile ISD projects. The scale was anchored at (1) never and (7) daily. Turnover intention was measured with two items adapted from Leiter et al. (2011). The correlation between the two items was very high (r = 0.86). For the moderator variable of *perceived workload*, we drew on two items used by Moore (2000a). These items also correlated strongly (r = 0.75). For both the dependent and the moderator variable, agreement scales were applied.

To account for alternative explanations, we included a number of control variables in the prediction of the dependent variable. In accordance with prior studies concerning IT turnover and work-related fatigue, we controlled for demographic data, such as gender and age, as well as for organizational tenure and negative affect (Ahuja et al., 2007; Chan & Wan, 2012; Joseph et al., 2007; Moore, 2000a). Negative affect was measured with three items adapted from the Positive and Negative Affect Schedule (PANAS) (Watson et al., 1988). We selected the three items from the PANAS that had the highest factor loadings compared to relevant prior studies (Tripp et al., 2016). Finally, we also included items concerning project characteristics, such as team size and team dispersion.

5.3 Data Analysis and Results

The hypotheses were tested using partial least squares structural equation modeling (PLS-SEM) with SmartPLS 3 (Ringle et al., 2015). Because of the relatively complex model, including moderated mediation effects and aggregate second-order constructs, PLS-SEM is particularly suitable for our analysis (Hair et al., 2017; Sarstedt et al., 2020). In addition, the approach is robust to relatively lean sample sizes and makes no distributional assumptions for the data (Chin, 1998). The minimum sample size for a robust calculation is ten times the maximum number of paths that are directed toward a certain construct in the structural model (Hair et al., 2017). Our sample size (N = 207) exceeds this threshold. We applied a bootstrapping procedure with no sign changes and 5,000 subsamples, as suggested by Hair et al. (2017), to assess the paths' significance levels. When modeling the hierarchical structural model, we used the repeated indicator approach Mode B (Becker et al., 2012) for the reflective-formative second-order constructs (agile SD practices and agile PM practices).

Table 1 presents the descriptive statistics and correlations of our latent variables. We evaluated convergent validity following Gefen and Straub (2005), using three criteria proposed by Fornell and Larcker (1981). All factor loadings were greater than 0.78 and significant. Composite reliabilities (CR) and average variances extracted (AVE) exceeded 0.82 and 0.70 respectively. Discriminant validity was assessed using the Fornell-Larcker-criterion (Hair et al., 2017). In our model, the value of the square root of a construct's AVE was always larger than the correlation of the construct with any construct in the model (see Table 1). In addition, we performed a cross-loading analysis. All items of the model loaded higher on their intended construct than on any other construct with the difference exceeding 0.10 (Gefen & Straub, 2005) (see Appendix C). Collectively, despite using shortened scales, the convergent and discriminant validity criteria indicated a satisfactory quality of the measurement models. Additionally, the correlations in Table 1 do not indicate any systematic relationships between the extent of the use of agile ISD practices and perceived workload supporting our argument for perceived workload as a valuable independent moderator in our model.

Table 1. Descriptive Statistics of Latent Variables

ID	Latent variables	Mean	*SD*	Loading range	CR	AVE	Correlations and square roots of AVEs of latent variables											
							1	2	3	4	5	6	7	8	9	10	11	12
1	Coding Standards	5.73	1.19	.918-.922***	.917	.847	.920											
2	Continuous Integration	5.39	1.20	.828-.852***	.828	.706	.560	.840										
3	Daily standups	5.68	1.32	.896-.918***	.903	.822	.417	.579	.907									
4	Feelings of Fatigue	3.06	1.23	.785-.856***	.917	.687	-.170	-.254	-.156	.829								
5	Iterative delivery	5.59	1.18	.847-.879***	.896	.741	.571	.592	.590	-.191	.861							
6	Negative affect	2.23	1.35	.872-.919***	.928	.810	-.301	-.324	-.201	.685	-.261	.900						
7	Perceived workload	3.76	1.50	.808-.923***	.858	.752	-.132	-.249	-.219	.568	-.279	.463	.867					
8	Refactoring	5.58	1.21	.854-.881***	.859	.753	.632	.631	.458	-.267	.589	-.259	-.254	.868				
9	Retrospectives	5.55	1.18	.857-.889***	.865	.762	.506	.596	.579	-.146	.677	-.163	-.177	.596	.873			
10	Short iterations	5.35	1.26	.829-.856***	.831	.710	.458	.614	.573	-.142	.591	-.214	-.137	.542	.571	.843		
11	Turnover intention	2.14	1.36	.912-.943***	.925	.860	-.414	-.389	-.363	.485	-.348	.537	.361	-.381	-.321	-.284	.928	
12	Unit tests	5.32	1.35	.867-.888***	.870	.770	.547	.535	.475	-.187	.526	-.233	-.146	.505	.441	.527	-.273	.878

Note: Abbreviations: CR, Composite reliability; AVE, Average variance extracted. Square roots of the AVE depicted on the diagonal. ***$p < 0.00$

We addressed the potential threat of common method bias by performing Harman's single-factor test (Podsakoff et al., 2003). The highest level of covariance explained by a single factor is 21.63%, reducing the concern of common method bias. Moreover, we conducted a full collinearity test according to Kock (2015). The variance inflation factors (VIFs) calculated with our data are well below the threshold of 3.3 giving us further confidence that there is no issue with common method bias in our study (see Appendix C).

The results of our analysis are shown in Table 2. Models 1 to 3 display the results predicting feelings of fatigue. Model 1 represents the baseline model, Model 2 represents the main effects only, and model 3 represents the moderation effects. Models 4 and 5 show the results of the mediation analysis. We predicted in H1a that the use of agile SD practices would have a resource-enhancing effect on developers, and in H1b, we predicted that the use of agile PM practices would have a resource-depleting effect on developers. The results show a significant negative effect of the use of agile SD practices on fatigue (β = -0.23; $p < 0.05$), hence supporting H1a. However, the data does not support our hypothesis of the depletion effect of agile PM practices and we thus had to reject H1b. When analyzing the data with regard to our moderation hypotheses H3 and H4, we found significant interaction effects between the use of agile SD practices and perceived workload (β = -0.18; $p < 0.05$) and between the use of agile PM practices and perceived workload (β = 0.23; $p < 0.01$).

In order to understand the nature of these interactions, we conducted a simple slope analysis. Figure 3a illustrates the simple slopes of the use of agile SD practices on developers' levels of fatigue at high and low levels of perceived workload (i.e.,1 and 2 StD above and below the mean). Figure 3b shows the marginal effect of perceived workload on the interaction effect. As can be seen, perceived workload moderates the effect of the use of agile SD practices such that the more developers perceive their workload as high, the stronger and more negative the effect on their levels of fatigue will be. Higher perceived workload seemingly amplifies the energizing effect of the use of agile SD practices. By contrast, agile SD practices has no significant effect on fatigue when developers perceive their workload as low.

The simple slope analysis depicted in Figures 4a and 4b shows the effect of using agile PM practices on developers' levels of fatigue at high and low levels of perceived workload (i.e., 1 and 2 StD above and below the mean). It indicates that the use of agile PM practices is significant and strongly positively related to developers' levels of fatigue when developers experience higher workload levels. Hence, a higher perceived workload seems to invoke a depleting effect related to the use of agile PM practices. Here again, at low levels of perceived workload, no significant effects of using agile PM practices on feelings of fatigue are observed. Overall, these results of the moderation analysis provide support for hypotheses H3 and H4. Models 2 and 3 in Table 2 show significant improvements in explained variance, which provides further support for H1a, H3, and H4. In total, the moderated model explains 38% of the variance.

In order to test for the mediation effects posited in H2a and H2b, we followed the two-step approach by Preacher and Hayes (2008) and first analyzed the direct effects of the independent variables on the dependent variable (Model 4 in Table 2).

Table 2. Results of PLS-SEM Analysis Predicting Feelings of Fatigue and Turnover Intention

	Feelings of fatigue			Turnover intention	
	Model 1	**Model 2**	**Model 3**	**Model 4**	**Model 5**
Block 1: Controls					
Gender	-05	-.05	-.01	.01	.01
Age	-.05	-.02	.06	-.06	-.07
Team size	-.16°	-.05	-.08		
Team dispersion	.00	-.02	-.02		
Negative affect				.43**	.28**
Organizational tenure				.06	.06
Block 2: Main effects					
Use of agile PM practices		.13	.10	-.15	-.16
Use of agile SD practices		-.23*	-.22*	-.20	-.19
Perceived workload		.54**	.54**		
Block 3: Interaction effects					
Use of agile PM practices × Perceived workload			.21**		
Use of agile SD practices × Perceived workload			-.18*		
Block 4: Mediator					
Feelings of fatigue					.21**
R^2	.03	.35	.38	.38	.41
ΔR^2		.32	.03		.03
Note. $p < 0.1$, $*p < 0.05$, $**p < 0.01$					

Figure 3a

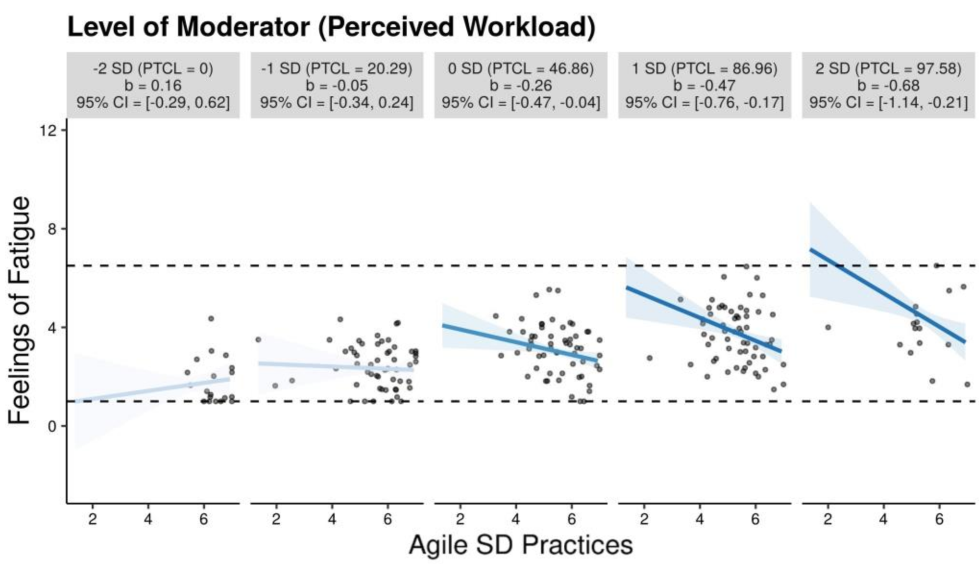

Figure 3b

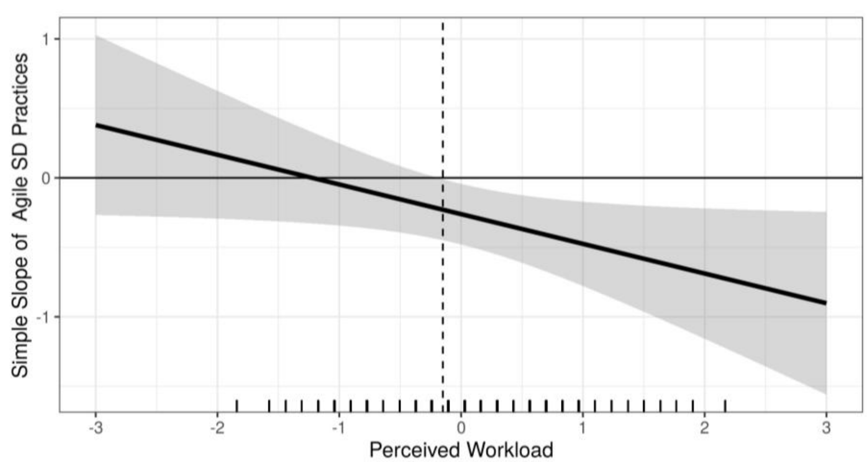

Figure 3. Simple Slope Analysis for Agile SD Practices

Figure 4a

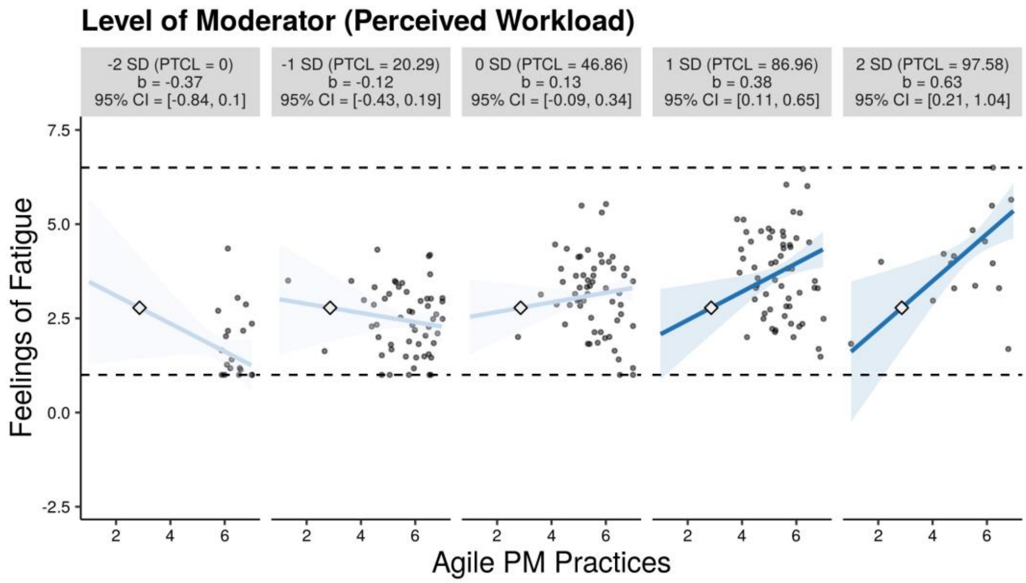

Figure 4b

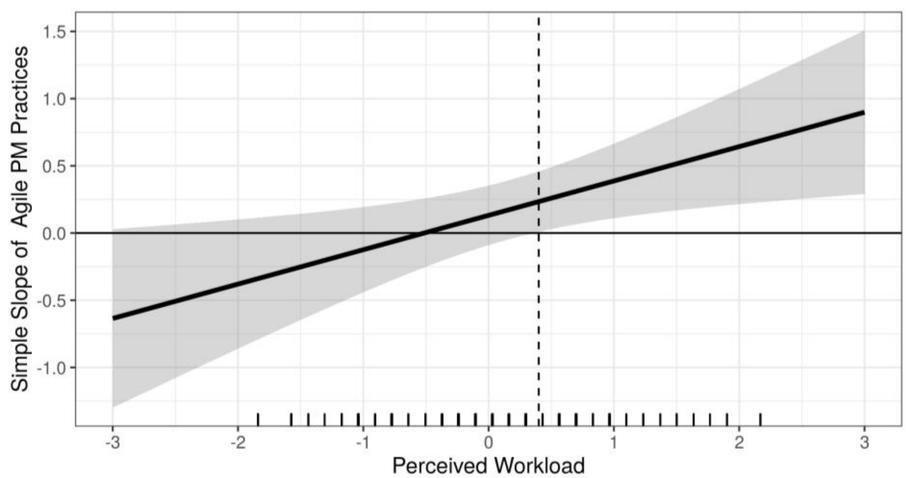

Figure 4. Simple Slope Analysis for Agile PM Practices

The model explained 38% of the variance in turnover intention. The path coefficients for the direct effects of using agile SD practices and using agile PM practices on turnover intention were not significant. In Step 2, we introduced the mediator, feelings of fatigue (Model 5 in Table 2). The explained variance in turnover intention increased to 41%. The effect of feelings of fatigue on turnover intention was positive and significant ($\beta = 0.21$; $p < 0.01$). The effect of the use of agile SD practices on feelings of fatigue was negative and significant ($\beta = -0.29$; $p < 0.05$) leading to a significant indirect effect of the use of agile SD practices on turnover intention via feelings of fatigue ($\beta = -0.06$; $p < 0.05$) (Hair et al., 2017). The direct effect of agile SD practices on turnover intention remains insignificant. Thus, feelings of fatigue fully mediate the relationship between the use of agile SD practices and turnover intention, supporting hypothesis H2a. Since the use of agile PM practices has a positive but nonsignificant effect on feelings of fatigue, H2b had to be rejected. We remark that the control variable *negative affect* had a significant effect on turnover intention. However, its path coefficient was lower when the model included feelings of fatigue. This result gives us reason to believe that the use of agile ISD practices through developers' levels of fatigue has a significant impact on their intention to change jobs beyond affective effects and mood.

We next calculated the indices of moderated mediation following Hayes (2015) and Hair et al. (2017) in order to examine the effects of the use of the agile ISD method on turnover intention in more detail. The index of moderated mediation shows the effect of the moderating variable on the indirect effect of the independent on the dependent variable via the mediator (Hayes, 2015). For the relationship between agile PM practices and turnover intention the index of moderated mediation is positive and significant ($\beta = 0.05$; $p < 0.05$). Hence, when the perceived workload increases, the positive indirect effect of agile PM practices on turnover intention via feelings of fatigue becomes stronger. The index of moderated mediation for the relationship between agile SD practices and turnover intention is negative and marginally significant (β = -0.04; $p < 0.1$), indicating a stronger negative indirect effect of the use of agile SD practices on turnover intention via feelings of fatigue when the perceived workload increases. These results provide additional support for our hypothesized moderating effects of perceived workload and show that the mediated effects of the use of agile ISD practices and their interaction with perceived workload entail significant implications for developers' feelings of fatigue.

6 Qualitative Interview Study

6.1 Data Collection and Analysis

Semi-structured interviews formed the basis of our qualitative data collection. We employed purposeful sampling and selected software developers currently working in agile ISD teams and used most of the agile ISD practices analyzed in the quantitative study (Palinkas et al., 2015). We contacted the developers via social network sites (e.g., LinkedIn) and invited our personal and research networks to forward our call to appropriate candidates. Fifteen interviews with software developers who work in agile ISD teams in Germany were conducted face-to-face or via phone and lasted between 30 and 60 minutes each. After the fifteenth interview, we observed that the new data did not provide substantial additional insights related to our study and we had reached saturation (Keutel et al., 2014). Our interview partners came from various industries, had an average age of 34, and reported an average experience with agile practices of four years. Appendix D summarizes the information about our interviewees.

The semi-structured interviews followed established guidelines for qualitative interviews (Myers & Newman, 2007). Our interview guideline was structured with regard to the agile ISD practices we focused on during our quantitative data collection. We asked the developers about their attitudes and feelings toward the different agile PM and SD practices under normal- and high-workload conditions in order to gain additional insights into the nature and causes tested in the quantitative data analysis.

In line with our aim to triangulate and expand our quantitative findings, we analyzed the qualitative data using ego depletion theory as a lens to interpret the results (Sarker et al., 2018; Walsham, 1995). We formally coded each interview to gain insights into the mechanisms potentially causing self-regulation effects when using agile ISD methods. To accomplish this, three knowledgeable researchers read the transcripts and marked each sentence that referred to feelings of depletion or energy related to the use of agile ISD methods or specific practices. Two of these researchers were unfamiliar with the quantitative study, thereby controlling for a potential bias through prior empirical knowledge. We also remained open for themes not affecting a complete agile ISD method but spanning multiple agile ISD practices (e.g., meetings). The coding was compared and discussed in case of discrepancies until a shared interpretation of the interviews was reached. During this process, we organized the resource-depleting and resource-replenishing mechanisms according to agile SD and PM practices and the level of workload indicated by the interview partners. We also follow this structure in the presentation of the qualitative results.

6.2 Findings

6.2.1 Relationship between the Use of Agile SD Practices and Feelings of Fatigue

With regard to agile SD practices, the answers from the interview partners were overall very positive. The consequences of their use mentioned by the developers tended generally toward making everyday work easier. Based on the interviews, the automation of processes and tests that continuous integration practices and unit tests provide greatly facilitate developers' work. This automation allows for quick checks when needed and eliminates the need for developers to perform tedious tasks. Two interview partners indicate the benefits of automation on daily work routines in the following quotes:

> *Testing is very annoying, especially manual testing is very annoying. So, [with unit tests] you can run the automated tests as often as you want, this can be done quickly in case of doubt.* (D5)

> *And then processes are automatically started that somehow deploy this to the system. It makes life a lot easier.* (D3)

Further, some interviewees stated that the immediate feedback inherent in automated unit tests and continuous integration practices also gives the developers a feeling of certainty while coding:

> *The great thing is that the unit tests in the API [application programming interface] will show you very quickly if you accidentally break something. It's very reassuring when you've made a lot of changes.* (D10)

> *To some extent, it [continuous integration] gives you the certainty that you have done everything right.* (D8)

Moreover, the interviews indicated that the quality of the product benefits greatly from agile SD practices. In particular, the interviewed developers mentioned that running unit tests and performing conscientious refactoring lead to quality enhancements. Additionally, they stated that refactoring sets the stage for a higher speed of development later on because code snippets become modular and reusable. Two developers succinctly emphasized these benefits:

> *Unit tests increase the quality of the software tremendously.* (D5)

> *If you refactored a module, it looks like before but with a more solid base and later features can be programmed much faster.* (D11)

The interviews indicated that coding standards add to the positive attitude toward agile SD practices; besides improving code consistency, they provide clear and objective guidelines for developers. These guidelines make the code understandable for every existing or new team member. Hence, coding standards facilitate discussions on code between developers and speed up the onboarding of new colleagues. As our interview partners explained:

> *I think it's good because you stay clean and consistent from the start, making the code easier to read.* (D5)

> *There is no more discussion how it should look like, the linter shows you what is wrong. I find it very comfortable. ... If you don't have such a thing, it certainly leads to confusion.* (D7)

> *Standards make it easier for people to get started, because they know how it is done. It is readable and understandable.* (D12)

In addition to being generally very positive about agile SD practices, some developers were also somewhat critical, particularly concerning the practice of refactoring. Apart from improving product quality, they reported that refactoring code is a very time-consuming task. Since developers therefore cannot code as much as they would like to, they eventually experience frustration. As one interview partner stated:

> *Sometimes I get frustrated when I have to restructure the code before I can do my job.* (D9)

The refactoring conflict between more time spent and higher product quality, which facilitates code writing, especially later on, is illustrated in the following quote. The interviewee makes clear that developers only feel appreciated for new features, not for refactored features:

> *A programmer is always a little bit more appreciated from the outside when he produces features. That's why you might be a bit more eager to generate new features and maybe not pay as much attention to how the code is written... It will always be a battle between product owners and the coders. Between "Okay, we have to refactor this" and the product owner who says: "It doesn't matter now. The important thing is that we deliver the feature." ... Clearly, refactoring is very time-consuming at the beginning when you do the feature but, overall, you save time because the product becomes much more modular, reusable, and stable.* (D14)

Overall, however, the benefits of automation, continuous feedback, quality, and consistency provided by the various agile SD practices clearly prevailed during the interviews. One interview partner summarized characteristics that help developers to do their job successfully as follows:

> *Coming to XP, all the technical stuff we have helps us a lot. Every test that runs automatically, every integration, every time a deployment happens automatically and not by hand if I don't have to manually move files to the server. That happens automatically after every change, saves time, prevents sources of error, and leads to a better product, to better software. Also, standards, much less own opinion, much less subjectivity ... all that reduces sources of error.* (D1)

The benefits mentioned by the interviewed developers support a resource-enhancing view of applying agile SD practices. Automation and continuous feedback are mechanisms that can reduce workflow interruption and the effort to attract attention, thereby conserving self-regulatory resources (Baumeister et al., 2007; Beal et al., 2005). The high quality and consistency of the code facilitate the cognitive processing of information and reduce the number of conflicts in the team, diminishing the need to build self-regulatory resources (Muraven & Baumeister, 2000; Schmeichel et al., 2003).

6.2.2 Relationship between the Use of Agile PM Practices and Feelings of Fatigue

With regard to the use of agile PM practices, we observed less of a clear trend in the interviews. Our data mainly elucidates that the use of these practices can facilitate collaboration but, at the same time, they can cause distractions and negative feelings. In particular, our participants criticized the high number of meetings necessitated by such practices. They mentioned that agile meetings lead to less progress in their work and disrupt their workflow. This is described by two interview partners as follows:

> *Way too many [meetings]. Just yesterday was retro[spective] day, which means we had a retro in the morning, which I think lasted two hours in the end. After that we had a refinement, and after that we had another meeting about design changes. That was a half-hour meeting, and it came out that I had to change one word. You could have told me that on the go. I don't think I opened my laptop before 4 pm. And I was in the office at 8 am.* (D1)

> *If you do sprints of two weeks, you have the feeling that you only do sprint plannings, sprint reviews and retrospectives. You have just started, but you are already back to [the] retrospective. ... Because of the distraction from normal work, as I said, when I do sprints for a week or two, we used to feel like we were just starting and then we had to stop again. ... I realized then that two weeks suck.* (D2)

In addition, some developers clearly adopt a fatalistic view and experience agile meetings as futile. They do not see any added value, especially in the daily standups, and just endure them. As two developers explained:

> *There are parties who are very communicative and talk about something for 10 hours and then the others fall asleep. Then a certain fatalism develops. Then you go to the sprint meeting and let the half hour, 15 minutes pass and then you leave again. And it didn't do you any good.* (D2)

> *I'd say I've never had a cool daily. Well, it was always so-so. When you're on a [team] anyway, you know what's going on. I don't understand what kind of information flow there should be.* (D11)

Others referred to the retrospective as a type of meeting they think is overvalued. They get annoyed by continuously being forced to evaluate the team dynamics and processes. One interview partner even reported making things up just to get the meeting finished and avoid further discussion:

> *I personally find a very frequent retrospective, namely after every sprint, very annoying. You start making up things to say something or people get annoyed when they have nothing to say.* (D5)

From the statements above we can derive a series of self-regulatory tasks and feelings. Directing attention back to the development task after being distracted and dealing with negative emotions in meetings that do not provide developers with any benefit are actions that require self-regulation and could lead to depletion and feelings of fatigue (Baumeister et al., 2007; Muraven & Baumeister, 2000).

Apart from these critical views, we also observed very positive reactions and opinions regarding the use of agile PM practices. Some interviewees reported that they appreciated the constant information flow and easy communication enabled by agile PM practices. In general, they stated that knowing what is going on gives them a feeling of direction. One developer stated:

> *I think it is very useful for a developer to get constant feedback. You also feel more secure because you know it's going in the right direction* (D12)

Further, some interview partners named the daily standups, in particular, as well as the iteration planning meetings as sources of information that are helpful for their daily work. They stated that the discussions in these meetings create a high level of transparency, which saves effort and prevents unnecessary work. As two developers explained:

> *It's really good so that the others always know what's going on every day. Especially if you have bigger stories or you are working with several people on a story, the discussion is very important. ... Today in the stand-up I got the information that my colleague had gotten so far with the topics that I can start with my story. So, I had the information immediately and didn't have to approach him myself first.* (D7)

> *We hold these [iteration planning] meetings together because we've noticed that otherwise there will be too much frictional loss. For example, that app developers program things that already exist or that they lack information about how the marketplace works.* (D6)

Additionally, some interviewees reported that retrospectives contribute to good feelings and a common understanding among developers. Whereas other meetings tend to be rather loaded with technical information important for the project's progress, they stated that the discussions in the retrospectives build a positive atmosphere and help to strengthen team cohesion. In the words of one interviewee:

> *When we introduced it, it was really well received and it is still important. There's a lot of discussion and because it's not so technical, code heavy, but more about how the developers feel, I think it's really important and good.* (D11)

Moreover, the different project structure, namely the iterative approach with short iterations of between two and four weeks, was described by some of our interviewees as leading to a great sense of achievement and self-efficacy because developers frequently see what they have accomplished. The visibility of progress spurs individual motivation and enthusiasm, as illustrated by the following two quotes:

> *I really think you have more of a sense of achievement, because I always have the feeling that we have completed a sprint and we have speeded up again. ... Everyone claps and congratulates each other. I actually find the flow quite nice.* (D4)

> *I can tell we're getting things done, which is incredibly motivating.* (D9)

In contrast to the resource-depleting factors we mentioned earlier, the positive feelings, motivation, and shared understanding among developers invoked by the implementation of agile PM practices likely have a positive impact on developers' levels of self-regulatory resources. Research has shown that these characteristics can counteract the depletion effect and lead to the replenishment of personal resources (Baumeister & Vohs, 2007; Baumeister et al., 2007; Tice et al., 2007).

Overall, the results of our qualitative study reveal no trend concerning positive or negative consequences of agile PM practices with regard to developers' levels of regulatory resources. We observed depleting as well as replenishing mechanisms triggered by the implementation of each of these practices. One general quote regarding the culture of continuous feedback inherent in agile PM practices nicely illustrates this self-regulation ambivalence:

> *Usually I appreciate feedback, even if it is not always tasty. As long as it is respectful and has a comprehensible reason. But actually, it is also the case that I often have difficulties to keep my ego out of it. And to say, yes you might be right, yes I can also work on it.* (D13)

6.2.3 Relationship between the Use of Agile SD and PM Practices and Feelings of Fatigue during Periods of High Perceived Workload

When asked about their opinions concerning the use of agile ISD practices during periods of high perceived workload, the responses of the interviewed developers differed depending on whether they used agile SD or agile PM practices. Applying *agile SD practices* seems to have beneficial effects during peak times. The automation of tasks during the process of programming and the clear guidance provided by coding standards facilitate developers' everyday work. The practices ensure that developers can keep up with the higher number of tasks when the workload is high, thus ensuring that product quality is maintained. In addition, the code consistency established through following standards and refactoring code pays off in the long run because it makes higher workloads easier for developers to handle. Two interviewees explained the importance of continuous integration and coding standards in the following quotes:

> *Continuous integration is already there, you don't have to do anything but press a button, it always works.* (D10)

> *Especially when you have a high workload, mistakes happen. The good standard makes sure that the errors are eliminated. This means that especially when you have a high workload, you should follow the rules. I'm aware of that.* (D4)

One developer pointed out that agile SD practices not only facilitate work but also speed up task accomplishment. This is particularly favorable when the number of tasks accumulates and the perceived workload is high:

> *So, it speeds up the work considerably, because the overview is provided. Especially then, when you have a lot to do, you still keep the overview. ... It accelerates the next task at the same place.* (D7)

Performing refactoring has already been characterized as a time-consuming task when workloads are normal. Even though a generally well-refactored code enhances quality and consistency, developers do not engage in this task when the workload is high because, at such times, delivering working code is more important to the customer. As one interviewee stated:

> *The problem with stressful situations is of course that people say, yes, they have to get the features out now. No one of the bosses would say, yes, just do a refactoring for a month* (D13)

To summarize, the interviewed developers provided reasons to believe that developers can benefit significantly from applying agile SD practices when the workload is perceived to be high. The positive feelings that arise when one gets things done intensify during high-workload situations because developers appreciate the ability to exploit the efficiency provided by these practices even more than they do during periods of normal workloads. This can accelerate the replenishing effect on self-regulatory resources (Baumeister et al., 2007; Muraven & Baumeister, 2000; Schmeichel et al., 2003; Tice et al., 2007).

In contrast, with regard to the use of *agile PM practices*, the interview partners stressed the disruptions that they experience due to plentiful meetings. These meetings cause feelings of displeasure to arise because developers have difficulties completing their programming tasks even though frequent results are expected. Two developers expressed their dissatisfaction with the strict adherence to agile PM practices in high-workload situations as follows:

> *When I have a lot to do, which is often the case, then it is annoying. Everybody wants to finish his tasks in the sprint first. And I think, okay, I don't have 15 minutes for a daily and I don't want to have a review or a retrospective because I want to get my tasks done.* (D14)

> *Especially when it comes to the end of the sprint it would be better to skip it, because then the work accumulates, ... because often there are things that have to be done, and the daily is just a time waster.* (D15)

Other developers reported investing high levels of personal effort to maintain concentration during the meetings and when processing their tasks. They reported having to invest a lot of self-control to stick to the agile PM practices when workloads are high. One interview partner explained:

> *In particular, if there are urgent things and you have a lot to do, then sometimes you have to ask yourself: do we have to do the one-hour retro now, because actually everybody has so much to do and there is not so much to talk about. ... Sometimes then I have to pull myself together to play by the rules.* (D4)

In addition, our interview partners reported that the time pressure inherent to higher-workload periods reduces discipline and thereby inhibits necessary process improvements. Frustration emerges because meetings are not only disruptive but also futile in such cases. One developer expressed his opinion vividly:

> *[During high workload] we can't really change anything from any retro. So, we start from scratch every time. ... It might last for a week and then we get back to the usual mode because the workload is too high. We identify very large pain-points with the help of a retro and address them. We have really good action items to counteract them. But we don't stick to that. We only have the discipline for a week, and then we don't have the time to stick to it. I always think of a metaphor, like you get in cartoons. We have big weights on both legs hanging from a chain on the foot ... and we're so busy racing that we don't have time to saw off the weights.* (D1)

Overall, the developers' experiences with using agile PM practices in high-workload situations indicate intensified negative feelings toward them and a higher need to control both feelings and actions. These consequences likely lead to higher consumption of self-regulatory resources and feelings of fatigue (Beal et al., 2005; Lanaj et al., 2016; Muraven & Baumeister, 2000).

Comparing the interview partners' responses regarding the use of agile SD and PM practices during high-workload periods with the results from the quantitative study, we can infer a similar trend. Quantitatively, we observed that agile SD practices enhance and agile PM practices deplete developers' self-regulatory resources. Indicating similar effects of agile SD and PM practices, our qualitative findings complement the quantitative results by providing explanations for why these opposite consequences occur. One interview partner succinctly points out the difference:

> *To sum it up, if the workload is high, then the Scrum meetings are rather inconvenient and the XP methods are mostly beneficial.* (D15)

6.2.4 Integration of Qualitative and Quantitative Study Results

Table 3 synthesizes our findings and incorporates the various mechanisms mentioned by the interview partners that potentially lead to the depletion or enhancement of self-regulatory resources. The qualitative study thereby enhances our understanding of the reasons for the (un-)intended consequences of the use of agile SD/PM practices for developers and adds to a more nuanced view of agile concepts in ISD. Table 3 also depicts how the qualitative findings map onto the results of our quantitative field study with a focus on the direct (i.e., H1a, H1b) and moderating effects (i.e., H3 and H4) formulated in our hypothesis development. Next, we discuss our overall results and their implications.

7 Discussion

We set out to provide a more balanced view of the individual-level effects of the use of agile ISD practices for developers. Drawing on EDT, we developed and tested a model that suggests that the use of agile SD practices reduces and the use of agile PM practices enhances developers' feelings of fatigue, thereby respectively decreasing and increasing their intention to change jobs. We additionally theorized that both effects are stronger when the workload is perceived to be high.

Table 3. Mapped Results of Qualitative and Quantitative Studies

Hypotheses	Quantitative study result	Qualitative study findings: Resource-enhancing factors	Qualitative study findings: Resource-depleting factors
H1a: The use of agile SD practices negatively influences developers' levels of fatigue.	Supported	• Automation • Code consistency • Feeling of certainty • Immediate feedback • Fewer discussions • Product quality • Short onboarding • Speed of development	• Feels time-consuming • Frustration
H1b: The use of agile PM practices positively influences developers' levels of fatigue.	Rejected	• Common understanding • Constant feedback • Easy communication • Motivation • Self-efficacy • Sense of achievement • Transparency	• Displeasure • Distraction • Fatalism • Futility • No progress • Self-control
H3: Perceived workload negatively moderates the relationship between the use of agile SD practices and developers' feelings of fatigue such that when the perceived workload is high (vs. low), developers experience fewer feelings of fatigue from the use of agile SD practices.	Supported	• Automation • Code consistency • Feels beneficial • Speed of development	• Feels time-consuming
H4: Perceived workload positively moderates the relationship between the use of agile PM practices and developers' feelings of fatigue such that when the perceived workload is high (vs. low), developers experience more feelings of fatigue from the use of agile PM practices.	Supported	[no factors identified]	• Discomfort • Displeasure • Frustration • Futility • No progress • Self-control

The results confirmed most of our hypotheses. They show that the use of agile SD practices has a replenishing effect on developers' regulatory resources (leading to less fatigue) and that this effect becomes stronger when developers are experiencing higher workloads.

Moreover, our mediation analysis reveals an explanatory mechanism for why the use of agile SD practices is beneficial in the context of ISD work. Developers' feelings of fatigue as a mediator carries the positive effects of the use of agile SD practices over to their attitudes toward their current work, which ultimately results in lower turnover intentions, especially when the workload is high. Conversely, the use of agile PM practices has neither an energizing nor a depleting effect on developers. An explanation for this result might be that agile PM practices, especially the meetings inherent to these practices, may trigger both resource-draining and resource-enhancing processes. As hypothesized, developers may be exposed to many interruptions from their development work and subject to constant time pressure to deliver, requiring discipline, vigilance, and self-regulatory resources (e.g., Baumeister et al., 2007). However, frequent meetings also remind team members to interact closely and regularly with various stakeholders, which offers great potential to build a shared understanding and circumvent conflicts through mutually agreed upon targets (e.g., Hummel et al., 2015; Hummel et al., 2016; Maruping & Matook, 2020). This, in turn, can facilitate cognitive processing and create a positive atmosphere among developers, reducing demands on self-regulation (Muraven & Baumeister, 2000; Schmeichel et al., 2003; Tice et al., 2007). Hence, because resource-draining and resource-enhancing processes operate simultaneously, the effects may cancel each other out, making clear predictions with regard to the directions of the effect difficult (Benlian, 2022; Gross et al., 2011). However, we found a significant positive interaction between the use of agile PM practices and developers' perceived workload, such that in high-workload situations, developers experience increased depletion and fatigue from the use of agile PM practices. This finding highlights the importance of recognizing that for various reasons, such as subjective perceptions or only partially applied agile ISD practices, workloads may be experienced differently even in agile work environments, where a sustainable pace of development is targeted yet often unrealistic. Additionally, following the results of our moderated mediation analysis, we revealed that higher workloads also lead to an amplified positive effect of the use of agile PM practices on developers' turnover intentions via intensified feelings of fatigue.

The results from our qualitative study confirmed and expanded the results from the quantitative model testing, providing rich additional insights into agile ISD practices as potential antecedents of fatigue. In particular, the findings substantiate that while agile SD practices are associated with a resource-enhancing effect during periods of normal and high perceived workloads, agile PM practices exhibit a leveled effect when the workload is perceived to be normal and a resource-depleting effect when the workload is perceived to be high. The interview results concerning the use of agile PM practices during normal-workload periods, indicating a relatively equal level of individual costs and benefits, support our above-mentioned explanation for why we could not find an effect between these practices and fatigue in our quantitative study. In addition to confirmation, the findings expand the understanding of the underlying self-regulatory mechanisms of these effects. They provide detailed insights into the nature and causes of the relationship between the use of agile ISD practices and developer well-being and shed light onto the intended and likely unintended effects of introducing these practices, such as a higher speed of development and facilitated communication, on one side, and frustration and displeasure on the other.

7.1 Theoretical Contributions

Our results provide four important theoretical contributions to research on agile ISD, IT workforce, and ego depletion in work-related contexts. First and foremost, this study advances existing literature by providing a more balanced and comprehensive view on the implications of the use of agile ISD practices for individual developers, revealing resource-enhancing and resource-depleting mechanisms. Drawing on EDT, we theorized and empirically identified positive as well as negative consequences of using agile SD and PM practices, respectively, with several important theoretical implications. We have taken a first step toward bringing clarity to ambivalence in prior studies regarding the implications of using agile ISD practices for developers, such as the effects of increased communication intensity enhancing shared understanding yet requiring self-control (Ghobadi & Mathiassen, 2016; Hummel et al., 2013). In addition, our results complement the current knowledge on the individual-level effects of using agile ISD practices by extending the scope of research to potentially depleting demands that agile ISD practices, particularly agile PM practices, place on developers. Thus, we move beyond the predominant understanding of the use of agile ISD practices as an invariably positive phenomenon that enhances satisfaction and mitigates exhaustion (Tripp et al., 2016; Venkatesh et al., 2020). Our study offers an important counterpoint to a potential positivity bias in previous research on the consequences of the use of agile ISD practices. In addition, by showing that agile SD and PM practices not only differ from each other in their influence on individual developers' job perceptions but have opposing effects under specific conditions, we highlight the importance of taking a nuanced view on the use of specific types of agile ISD practices instead of aggregating them into a monolithic concept (Benlian, 2022; Tripp et al., 2016; Venkatesh et al., 2020).

Second, with the detailed insights from our qualitative study on the resource-enhancing and resource-depleting effects of using agile ISD practices, we answer the calls of Tripp et al. (2016) and Venkatesh et al. (2020) to extend research on the effects of the use of agile ISD practices on individual developers. In particular, we shed light on the psychological processes underlying the effects of using agile ISD practices on feelings of fatigue. On the one hand, our study showed that the use of agile SD practices decreases feelings of fatigue and thus enhances regulatory resources, especially in high-workload situations and that the use of agile PM practices consumes regulatory resources when the workload is high. On the other hand, our interview results added another perspective to the discussion on the psychological impact of the use of agile ISD practices and added rich explanations to the findings from the quantitative assessment. Agile SD practices lead to lower fatigue through higher automation, better product quality, and an enhanced feeling of certainty when coding. Concerning the use of agile PM practices, however, the resource-enhancing effect of constant feedback and shared understanding, on the one hand, and the depleting effect of needless meetings and little progress on the other, seem to cancel each other out. However, it is only when the workload is perceived to be high that the resource-depleting effects prevail such that fatigue is significantly affected by the use of agile PM practices. Taken together, our findings contribute to previous research by providing more nuanced insights into the psychological mechanisms underlying the individual-level consequences of the use of agile ISD practices (Benlian, 2022; Venkatesh et al., 2020).

Third, our moderation and moderated mediation analyses demonstrated that perceived workload intensifies the effect of both agile SD and agile PM practices on feelings of fatigue and, ultimately, turnover intention. Whereas agile SD practices had an increasingly energizing effect when the workload increased, agile PM practices were significantly more exhausting. Our qualitative analysis reinforces these quantitative results. To our knowledge, we are the first to systematically consider and confirm perceived workload as an important individual-level moderator in the context of agile ISD research (Fortmann-Müller, 2018). Perceived workload has the power to boost both resource-replenishing and resource-depleting effects in developers. We see two implications of this result. We complement research on the role of workload in agile ISD settings in the sense that varying levels of perceived workload are not only a consequence of the use of agile ISD practices (Tuomivaara et al., 2017) but also provide an important boundary condition and differentiator between specific types of agile ISD practices, which has been previously overlooked (Benlian, 2022). Moreover, by theorizing workload as a subjective perception in the form of a higher or lower intensity of engagement in one or more projects, we shift the existing conversation on workload effects in agile ISD from a project management (Tuomivaara et al., 2017; Vidgen & Wang, 2009) to a developer perspective, revealing that individual perceptions of workload can make a difference in how we assess the effects of the use of agile ISD practices.

Finally, our study extends research on the work-based implications of ego depletion following the call of Johnson et al. (2018). Our findings that the use of agile SD practices helps enhance regulatory resources in software developers add to the discussion of at-work factors that minimize fatigue (Uy et al., 2017). We looked at the influence of specific IT job design features, namely agile SD practices, as a mechanism to reduce work-related fatigue and energize developers during work. Our qualitative study identifies the automation of processes and behavior, product quality, and feelings of certainty as major drivers of the resource-enhancing effect of the use of agile SD practices. While these effects on regulatory resources are in line with prior research (e.g., Milkman, 2012; Muraven & Baumeister, 2000; Webb & Sheeran, 2003), we have taken a first step toward showing that they also hold and even potentiate in situations of high perceived workload. In addition to the effects *on* ego depletion, our results also show that turnover intentions in agile ISD contexts are affected *by* the level of self-regulatory resource consumption. Using agile SD practices lowers the intention to change jobs by reducing fatigue in developers. Hence, we reveal that turnover intentions in agile ISD contexts can be reduced not only by higher levels of satisfaction (Setor & Joseph, 2019) but also by self-regulatory effects and lower levels of fatigue.

7.2 Practical Contributions

Software developers are a scarce resource and their turnover is costly. Thus, retaining IT professionals is a key issue for IT organizations (Pflügler et al., 2018). Our findings suggest several opportunities to reduce the potential of fatigue and turnover intentions in software developers and make agile ISD projects more sustainable. Organizations should direct their attention toward providing training that creates awareness of potential resource-related consequences of agile ISD practices. In particular, trainers should sensitize developers and project managers to the resource-depleting implications of agile PM practices, show them what can be done about it, and encourage them to seek help and support, specifically in high workload situations. Collaboratively, project managers and developers may carefully consider whether certain agile PM practices can be suspended or reduced for some time based on the stage of the project in order to free up valuable time for coding. In line with the qualitative study results, we specifically propose to reevaluate the frequency of daily standups and retrospectives as well as the agile SD practice of refactoring because these are often perceived as particularly disruptive and annoying when workloads are high. At the same time, the training should emphasize the resource-enhancing effects of agile SD practices, which can provide compensation and reduce feelings of fatigue through automation, quality, and the speed of development even under heavy workloads. Based on our results, we also recommend that organizations adopt suitable tools and procedures that can increase automation, improve code consistency, and provide immediate feedback for developers. In addition, organizations might think about introducing interventions such as regular microbreaks or optimized project staffing processes to counteract potential deficiencies in agile ISD projects and minimize work-related fatigue.

Moreover, we show that the application of agile ISD practices not only benefits productivity and software product quality (Balijepally et al., 2009; Maruping et al., 2009a) but also has diverse psychological implications—positive and negative—for developers' well-being. Practitioners can use our results as additional arguments in order to convince skeptical executives (Rigby et al., 2016) to implement agile SD methods throughout software development functions and give agile ISD teams the autonomy to decide on the best way to do their work.

Lastly, self-regulation is an inner resource that is common across different domains, e.g., at work and at home (Baumeister et al., 1998; Courtright et al., 2016; Reina et al., 2017). The resource-enhancing and resource-depleting effects of the use of agile ISD practices may therefore not only spill over to activities at work outside of agile projects but also to family life (Benlian, 2020). Implementing agile SD practices and carefully balancing agile PM practices offer a means toward improving work-life balance. Because a balanced relationship between family and work responsibilities is increasingly important to employees (Kaarst-Brown et al., 2019), our findings can support organizations in strengthening their position in competing for talent.

7.3 Limitations and Future Research

Despite the theoretical and practical contributions of this research, our study has four major limitations, which could open up a series of interesting research directions. First, we used feelings of fatigue as a measure for the resource-depleting effects using agile ISD practices. However, we did not include an explicit construct to survey resource-enhancing effects in our research model. Instead, we assumed that a reduction in fatigue would lead to an enhancement of developers' regulatory resources. While this approach is in accordance with prior research (e.g., Gross et al., 2011), using constructs such as engagement, dedication, or vigor could be a fruitful avenue for future research to explicitly measure resource-enhancing effects (e.g., Diestel et al., 2015).

Second, we theorized and modeled ego depletion as being expressed by enhanced levels of fatigue, which, according to groundwork in this research field, is a particularly suitable proxy because the depletion of regulatory resources resembles the process leading to muscle fatigue. Hence, ego depletion not only occurs concurrently with the triggering action but leads to a period of scarcity until the resources are built back up again—a state that is expressed by intensified feelings of fatigue (Muraven & Baumeister, 2000; Muraven et al., 1998). However, other studies suggest that ego depletion is not entirely analogous to muscular fatigue but that it is a separate construct, based on the exhaustion of an inner energy (Vohs et al., 2011). Future work could embrace this distinctive, more specific view and examine the individual-level consequences related to the use of agile ISD practices with regard to other outcomes of self-regulatory resource depletion—for example, aggression or decision-making comprehensiveness (Reina et al., 2017; Vohs et al., 2011). In addition, future studies might measure ego depletion using measures beyond those of fatigue, exhaustion, and burnout (e.g., Chan & Wan, 2012; Courtright et al., 2016; Uy et al., 2017) such as Bertrams et al.'s (2011) scale, which assesses the psychological state of ego depletion.

Third, we recognize the potential risk caused by considering only the four most used practices of agile SD and PM methods and the use of shortened scales for our latent variables. We did these things to ensure an acceptable survey length, as recommended by the panel company and the pretest participants, in order to obtain as many complete and valid answers as possible. However, we concede that using fewer items may distort measures and findings, even though our validity and reliability analyses indicate the satisfactory quality of our measurement model. Therefore, encourage future researchers investigating the individual-level effects of using agile ISD practices to utilize complete preexisting scales as well as a more comprehensive list of practices.

Finally, we examined the consequences of using agile ISD practices at an individual level only. While this focus provided rich insights into the psychological processes of software developers working in agile ISD teams, we invite future research to consider multiple levels when studying agile ISD practices. Team or project characteristics, such as team distribution, project complexity, and leadership practices (Windeler et al., 2017), could generate further insights into the boundaries of resource-enhancing or resource-depleting effects regarding the use of agile ISD practices. Similarly, studying team performance as a valuable team-level outcome of the use of ISD practices (Kude et al., 2019; Pee et al., 2010) could be complemented by taking individual-level ego depletion into account. In addition to multilevel examinations, researchers could engage in dyadic studies and compare the implications of using agile ISD practices from different stakeholder perspectives, e.g., developers and product managers (Benlian & Haffke, 2016; Yakovleva et al., 2010).

Beyond addressing the shortcomings of our study, we recommend that further research take a more balanced perspective when examining the implications of using agile ISD practices for developers' well-being and job outcomes. We propose research on the unintended effects of the use of agile ISD practices to improve the understanding of the dark sides of agile ISD and how to mitigate them. Scholars might examine which individual and team-wide interventions are necessary and helpful to reduce resource depletion caused by using agile ISD practices. For example, they could refer to research on microbreaks or project staffing processes (e.g., Maurer, 2010; Zacher et al., 2014) to see whether these things can enhance the sustainability of agile ISD projects.

Moreover, complementing research on single agile ISD practices (e.g., Kude et al., 2019), future research could examine which individual agile SD or PM practices are more, or less, depleting in order to gain a more nuanced understanding of the main drivers behind the effects that we found in our study. In particular, our nonsignificant results concerning the depleting effects of agile PM practices suggest that, overall, they may be not as demanding as hypothesized and individual practices might cancel each other out. An examination on a more granular level of analysis might be a fruitful avenue for future research seeking to identify more and less beneficial practices and usage conditions for developers working with agile PM practices.

We additionally propose that scholars acknowledge the significant impact of the interaction between the use of agile ISD practices and perceived workload on individual developers in agile ISD teams. For example, we found that higher perceived workloads intensify feelings of energy when using agile SD practices. Therefore, further research might dig deeper into the role of these practices as job design features because they could potentially serve as buffers in high-workload situations. This is especially interesting against the backdrop that work overload and time pressure have been proposed as depleting employees' regulatory resources (Diestel & Schmidt, 2012; Prem et al., 2016).

Finally, the underlying mechanisms we identified in the qualitative study, such as frustration and futility, may serve as a starting point to study the unintended consequences of the use of agile ISD practices for important health-related aspects such as burnout and depression. Such endeavors would go above and beyond the focus on fatigue and turnover to better punctuate the health implications and individual-level sustainability of agile ISD work processes.

8 Conclusion

Our research was aimed at providing a nuanced and balanced view on the effects of using agile ISD practices on developers' feelings of fatigue and turnover intentions based on EDT. We theorized that the use of agile SD practices has an energizing (i.e., resource-enhancing) impact on individual developers' regulatory resources and hence decreases their level of fatigue and their turnover intentions, while the use of agile PM practices depletes developers' regulatory resources, enhancing their level of fatigue and intentions to change jobs. In addition, we proposed that these effects are amplified by higher levels of perceived workload. Our empirical studies largely supported the hypotheses in our model. The use of agile SD practices reduces turnover intentions via the mediating effect of fatigue, especially in high-workload situations. The results concerning the use of agile PM practices indicate a depletion effect during high-workload periods only. We contribute to a deeper understanding of the positive and negative individual-level implications of the use of agile ISD practices and demonstrate the importance of considering individual developers' perceived workloads when evaluating their effectiveness. Practitioners can use our findings to help them take effective actions to retain key IT employees and when tailoring agile ISD practices to organizational settings.

References

Ahuja, M. K., Chudoba, K. M., Kacmar, C. J., McKnight, D. H., & George, J. F. (2007). IT road warriors: Balancing work-family conflict, job autonomy, and work overload to mitigate turnover intentions. *MIS Quarterly, 31*(1), 1-17.

Armstrong, D. J., Brooks, N. G., & Riemenschneider, C. K. (2015). Exhaustion from information system career experience: Implications for turn-away intention. *MIS Quarterly, 39*(3), 713-728.

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

Ashcraft, C., McLain, B., & Eger, E. (2016). *Women in tech: The facts*. National Center for Women & Information Technology.

Aspinwall, L. G. (1998). Rethinking the role of positive affect in self-regulation. *Motivation and Emotion, 22*(1), 1-32.

Baham, C., & Hirschheim, R. (2022). Issues, challenges, and a proposed theoretical core of agile software development research. *Information Systems Journal*, *32*, 103-129.

Balijepally, V., Mahapatra, R., & Nerur, S. P. (2006). Assessing personality profiles of software developers in agile development teams. *Communications of the Association for Information Systems*, *18*, 55-75.

Balijepally, V., Mahapatra, R., Nerur, S. P., & Price, K. H. (2009). Are two heads better than one for software development? The productivity paradox of pair programming. *MIS Quarterly*, *33*(1), 91-118.

Balijepally, V., Nerur, S. P., & Mahapatra, R. (2015). Task mental model and software developers' performance: An experimental investigation. *Communications of the Association for Information Systems*, *36*, 53-76.

Barlow, J. B., 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*, 25-44.

Baumeister, R. F., Bratslavsky, E., Muraven, M., & Tice, D. M. (1998). Ego depletion: Is the active self a limited resource? *Journal of Personality and Social Psychology, 74*(5), 1252-1265.

Baumeister, R. F., Gailliot, M., DeWall, C. N., & Oaten, M. (2006). Self-regulation and personality: How interventions increase regulatory success, and how depletion moderates the effects of traits on behavior. *Journal of personality*, *74*(6), 1773-1802.

Baumeister, R. F., & Vohs, K. D. (2007). Self-regulation, ego depletion, and motivation. *Social and Personality Psychology Compass*, *1*(1), 115-128.

Baumeister, R. F., Vohs, K. D., & Tice, D. M. (2007). The strength model of self-control. *Current Directions in Psychological Science*, *16*(6), 351-355.

Beal, D. J., Weiss, H. M., Barros, E., & MacDermid, S. M. (2005). An episodic process model of affective influences on performance. *Journal of Applied Psychology*, *90*(6), 1054-1068.

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

Becker, J.-M., Klein, K., & Wetzels, M. (2012). Hierarchical latent variable models in PLS-SEM: Guidelines for using reflective-formative type models. *Long Range Planning*, *45*(5-6), 359-394.

Benlian, A. (2020). A daily field investigation of technology-driven spillovers from work to home. *MIS Quarterly*, *44*(3), 1259-1300.

Benlian, A. (2022). Sprint zeal or sprint fatigue? The benefits and burdens of agile ISD practices use for developer well-being. *Information Systems Research*, *33*(2) 557-578.

Benlian, A., & Haffke, I. (2016). Does mutuality matter? Examining the bilateral nature and effects of CEO-CIO mutual understanding. *The Journal of Strategic Information Systems*, *25*(2), 104-126.

Bertrams, A., Unger, A., & Dickhäuser, O. (2011). Momentan verfügbare Selbstkontrollkraft–Vorstellung eines Messinstruments und erste Befunde aus pädagogisch-psychologischen Kontexten. *Zeitschrift für Pädagogische Psychologie*, *25*(3), 185-196.

Boehm, B. (2002). Get ready for agile methods, with care. *Computer*, *35*(1), 64-69.

Bono, J. E., Glomb, T. M., Shen, W., Kim, E., & Koch, A. J. (2013). Building positive resources: Effects of positive events and positive reflection on work stress and health. *Academy of Management Journal*, *56*(6), 1601-1627.

Chan, K. W., & Wan, E. W. (2012). How can stressed employees deliver better customer service? The underlying self-regulation depletion mechanism. *Journal of Marketing*, *76*, 119-137.

Chin, W. W. (1998). The partial least squares approach to structural equation modeling. In G. A.

Marcoulides (Ed.), *Modern methods for business research* (pp. 295-336). Lawrence Erlbaum Associates.

Conboy, K., & Fitzgerald, B. (2010). Method and developer characteristics for effective agile method tailoring: A study of XP expert opinion. *ACM Transactions on Software Engineering and Methodology, 20*(1), 1-30.

Courtright, S. H., Gardner, R. G., Smith, T. A., McCormick, B. W., & Colbert, A. E. (2016). My family made me do it: A cross-domain, self-regulatory perspective on antecedents to abusive supervision. *Academy of Management Journal, 59*(5), 1630-1652.

Coyle, S., Conboy, K., & Acton, T. (2015). *An exploration of the relationship between contribution behaviours and the decision making process in agile teams.* Paper presented at the 36th International Conference on Information Systems (ICIS), Fort Worth, TX.

Cropanzano, R., Rupp, D. E., & Byrne, Z. S. (2003). The relationship of emotional exhaustion to work attitudes, job performance, and organizational citizenship behaviors. *Journal of Applied Psychology, 88*(1), 160-169.

De Croon, E. M., Sluiter, J. K., Blonk, R. W., Broersen, J. P., & Frings-Dresen, M. H. (2004). Stressful work, psychological job strain, and turnover: a 2-year prospective cohort study of truck drivers. *Journal of Applied Psychology, 89*(3), 442-454.

Diestel, S., Rivkin, W., & Schmidt, K.-H. (2015). Sleep quality and self-control capacity as protective resources in the daily emotional labor process: Results from two diary studies. *Journal of Applied Psychology, 100*(3), 809-827.

Diestel, S., & Schmidt, K. H. (2012). Lagged mediator effects of self-control demands on psychological strain and absenteeism. *Journal of Occupational and Organizational Psychology, 85*(4), 556-578.

Drury, M., Conboy, K., & Power, K. (2012). Obstacles to decision making in agile software development teams. *Journal of systems and software, 85*(6), 1239-1254.

Duvall, P. M., Matyas, S., & Glover, A. (2007). *Continuous integration: Improving software quality and reducing risk*. Addison-Wesley.

Fitzgerald, B., Hartnett, G., & Conboy, K. (2006). Customising agile methods to software practices at Intel Shannon. *European Journal of Information Systems, 15*(2), 200-213.

Fitzgerald, B., Russo, N. L., & O'Kane, T. (2003). Software development method tailoring at Motorola. *Communications of the ACM, 46*(4), 64-70.

Fornell, C., & Larcker, D. F. (1981). Evaluating structural equation models with unobservable variables and measurement error. *Journal of marketing research, 18*, 39-50.

Fortmann-Müller, L. (2018). *Energizing or depleting? Understanding the effects of agile methodologies on individual software developers' resources.* Paper presented at the 39th International Conference on Information Systems.

Fowler, M., & Beck, K. (1999). *Refactoring: improving the design of existing code*. Addison-Wesley Professional.

Fowler, M., & Highsmith, J. (2001). The Agile Manifesto. *Software Development, 9*(8), 28-35.

Gefen, D., & Straub, D. (2005). A practical guide to factorial validity using PLS-Graph: Tutorial and annotated example. *Communications of the Association for Information Systems, 16*, 91-109.

Geister, S., Konradt, U., & Hertel, G. (2006). Effects of process feedback on motivation, satisfaction, and performance in virtual teams. *Small group research, 37*(5), 459-489.

Ghobadi, S., & Mathiassen, L. (2016). Perceived barriers to effective knowledge sharing in agile software teams. *Information Systems Journal, 26*(2), 95-125.

Gross, S., Semmer, N. K., Meier, L. L., Kälin, W., Jacobshagen, N., & Tschan, F. (2011). The effect of positive events at work on after-work fatigue: They matter most in face of adversity. *Journal of Applied Psychology, 96*(3), 654-664.

Hagger, M. S., Wood, C., Stiff, C., & Chatzisarantis, N. L. (2010). Ego depletion and the strength model of self-control: A meta-analysis. *Psychological Bulletin, 136*(4), 495-525.

Hair, J. F., Hult, G. T. M., Ringle, C. M., Sarstedt, M., Richter, N. F., & Hauff, S. (2017). *Partial Least Squares Strukturgleichungsmodellierung: Eine anwendungsorientierte Einführung*. Vahlen.

Hayes, A. F. (2015). An index and test of linear moderated mediation. *Multivariate behavioral research, 50*(1), 1-22.

Hazzan, O., & Dubinsky, Y. (2003). *Bridging cognitive and social chasms in software development using extreme programming.* Paper presented at the International Conference on Extreme Programming and Agile Processes in Software Engineering.

Hekkala, R., Stein, M.-K., Rossi, M., & Smolander, K. (2017). *Challenges in transitioning to an agile way of working*. Paper presented at the 50th Hawaii International Conference on System Sciences.

Hoda, R., Salleh, N., Grundy, J., & Tee, H. M. (2017). Systematic literature reviews in agile software development: A tertiary study. *Information and software technology, 85*, 60-70.

Huarng, A. S. (2001). Burnout syndrome among information system professionals. *Information Systems Management, 18*(2), 15-20.

Hummel, M., Rosenkranz, C., & Holten, R. (2013). The role of communication in agile systems development. *Business & Information Systems Engineering, 5*, 343-355.

Hummel, M., Rosenkranz, C., & Holten, R. (2015). The role of social agile practices for direct and indirect communication in information systems development teams. *Communications of the Association for Information Systems, 36*, 273-300.

Hummel, M., Rosenkranz, C., & Holten, R. (2016). *The role of shared understanding in distributed Scrum development: An empirical analysis*. Paper presented at the 24th European Conference on Information Systems.

Isen, A. M., & Reeve, J. (2005). The influence of positive affect on intrinsic and extrinsic motivation: Facilitating enjoyment of play, responsible work behavior, and self-control. *Motivation and Emotion, 29*(4), 295-323.

Jackson, S. E., Schwab, R. L., & Schuler, R. S. (1986). Toward an understanding of the burnout phenomenon. *Journal of Applied Psychology, 71*(4), 630-640.

Ji, Y., Mookerjee, V. S., & Sethi, S. P. (2005). Optimal software development: A control theoretic approach. *Information Systems Research, 16*(3), 292-306.

Jick, T. D. (1979). Mixing qualitative and quantitative methods: Triangulation in action. *Administrative Science Quarterly, 24*(4), 602-611.

Johnson, R. E., Muraven, M., Donaldson, T. L., & Lin, S.-H. J. (2018). Self-control in work organizations. In D. L. Ferris, R. E. Johnson, & C. Sedikides (Eds.), *The self at work: Fundamental theory and research* (pp. 129-144). Routledge.

Joia, L. A., & Mangia, U. (2017). Career transition antecedents in the information technology area. *Information Systems Journal, 27*(1), 31-57.

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.

Kaarst-Brown, M., Quesenberry, J., Niederman, F., & Weitzel, T. (2019). Special issue editorial: New approaches to optimizing the digital workplace. *MIS Quarterly Executive, 17*(4).

Keutel, M., Michalik, B., & Richter, J. (2014). Towards mindful case study research in IS: A critical analysis of the past ten years. *European Journal of Information Systems, 23*(3), 256-272.

Kock, N. (2015). Common method bias in PLS-SEM: A full collinearity assessment approach. *International Journal of e-Collaboration, 11*(4), 1-10.

Korunka, C., Hoonakker, P., & Carayon, P. (2008). Quality of working life and turnover intention in information technology work. *Human factors and ergonomics in manufacturing & service industries, 18*(4), 409-423.

Kude, T., Mithas, S., Schmidt, C. T., & Heinzl, A. (2019). How pair programming influences team performance: The role of backup behavior, shared mental models, and task novelty. *Information Systems Research, 30*(4), 1145-1163.

Lanaj, K., Johnson, R. E., & Wang, M. (2016). When lending a hand depletes the will: The daily costs and benefits of helping. *Journal of Applied Psychology, 101*(8), 1097–1110.

Laux, I., & Kranz, J. (2019). *Coexisting plan-driven and agile methods: How tensions emerge and are resolved*. Paper presented at the 40th International Conference on Information Systems.

Leatz, A. C., & Stolar, W. M. (1993). Career success/personal stress: How to stay healthy in a high-stress environment. McGraw-Hill.

Lee, Z. W., Cheung, C. M., & Chan, T. K. (2016). *Technology-mediated self-regulation: An implication for preventing online gaming addiction*. Paper presented at the 37th International Conference on Information Systems.

Leiter, M. P., Laschinger, H. K. S., Day, A., & Oore, D. G. (2011). The impact of civility interventions on employee social behavior, distress, and attitudes. *Journal of Applied Psychology, 96*(6), 1258-1274.

Lin, S.-H. J., & Johnson, R. E. (2015). A suggestion to improve a day keeps your depletion away: Examining promotive and prohibitive voice behaviors within a regulatory focus and ego depletion framework. *Journal of Applied Psychology, 100*(5), 1381-1397.

Maruping, L. M., & Matook, S. (2020). The multiplex nature of the customer representative role in agile information systems development. *MIS Quarterly, 44*(3), 1411-1437.

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.

Maruping, L. M., Zhang, X., & Venkatesh, V. (2009). Role of collective ownership and coding standards in coordinating expertise in software project teams. *European Journal of Information Systems, 18*(4), 355-371.

Maurer, I. (2010). How to build trust in inter-organizational projects: The impact of project staffing and project rewards on the formation of trust, knowledge acquisition and product innovation. *International journal of project management, 28*(7), 629-637.

McAvoy, J., & Butler, T. (2009). The role of project management in ineffective decision making within agile software development projects. *European Journal of Information Systems, 18*(4), 372-383.

McHugh, O., Conboy, K., & Lang, M. (2011). Using agile practices to influence motivation within IT project teams. *Scandinavian Journal of Information Systems, 23*(2), 85-110.

Milkman, K. L. (2012). Unsure what the future will bring? You may overindulge: Uncertainty increases the appeal of wants over shoulds. *Organizational Behavior and Human Decision Processes, 119*(2), 163-176.

Mingers, J. (2001). Combining IS research methods: Towards a pluralist methodology. *Information Systems Research, 12*(3), 240-259.

Moore, J. E. (2000a). One road to turnover: An examination of work exhaustion in technology professionals. *MIS Quarterly, 24*(1), 141-168.

Moore, J. E. (2000b). Why is this happening? A causal attribution approach to work exhaustion consequences. *Academy of Management Review, 25*(2), 335-349.

Moquin, R., Riemenschneider, C. K., & Wakefield, R. (2019). Psychological contract and turnover intention in the information technology profession. *Information Systems Management, 36*(2), 111-125.

Mor Barak, M. E., Nissly, J. A., & Levin, A. (2001). Antecedents to retention and turnover among child welfare, social work, and other human service employees: What can we learn from past research? A review and metanalysis. *Social service review, 75*(4), 625-661.

Muraven, M., & Baumeister, R. F. (2000). Self-regulation and depletion of limited resources: Does self-control resemble a muscle? *Psychological Bulletin, 126*(2), 247-259.

Muraven, M., Gagné, M., & Rosman, H. (2008). Helpful self-control: Autonomy support, vitality, and depletion. *Journal of Experimental Social Psychology, 44*(3), 573-585.

Muraven, M., & Slessareva, E. (2003). Mechanisms of self-control failure: Motivation and limited resources. *Personality and social psychology bulletin, 29*(7), 894-906.

Muraven, M., Tice, D. M., & Baumeister, R. F. (1998). Self-control as a limited resource: Regulatory depletion patterns. *Journal of personality and social psychology, 74*(3), 774-789.

Myers, M. D., & Newman, M. (2007). The qualitative interview in IS research: Examining the craft. *Information and Organization, 17*(1), 2-26.

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

Palinkas, L. A., Horwitz, S. M., Green, C. A., Wisdom, J. P., Duan, N., & Hoagwood, K. (2015). Purposeful sampling for qualitative data collection and analysis in mixed method implementation research. *Administration and Policy in Mental Health and Mental Health Services Research, 42*(5), 533-544.

Pee, L. G., Kankanhalli, A., & Kim, H.-W. (2010). Knowledge sharing in information systems development: a social interdependence perspective. *Journal of the Association for Information Systems, 11*(10), 550-575.

Pflügler, C., Wiesche, M., Becker, N., & Krcmar, H. (2018). Strategies for retaining key IT professionals. *MIS Quarterly Executive, 17*(4), 297-314.

Pikkarainen, M., Wang, X., & Conboy, K. (2007). *Agile practices in use from an innovation assimilation perspective*. Paper presented at the

28th International Conference of Information Systems.

Podsakoff, P. M., MacKenzie, S. B., Lee, J.-Y., & Podsakoff, N. P. (2003). Common method biases in behavioral research: A critical review of the literature and recommended remedies. *Journal of Applied Psychology, 88*(5), 879-903.

Preacher, K. J., & Hayes, A. F. (2008). Asymptotic and resampling strategies for assessing and comparing indirect effects in multiple mediator models. *Behavior Research Methods, 40*(3), 879-891.

Prem, R., Kubicek, B., Diestel, S., & Korunka, C. (2016). Regulatory job stressors and their within-person relationships with ego depletion: The roles of state anxiety, self-control effort, and job autonomy. *Journal of Vocational Behavior, 92*, 22-32.

Ramasubbu, N., Bharadwaj, A., & Tayi, G. K. (2015). Software process diversity: conceptualization, measurement, and analysis of impact on project performance. *MIS Quarterly, 39*(4), 787-807.

Reina, C. S., Peterson, S. J., & Zhang, Z. (2017). Adverse effects of CEO family-to-work conflict on firm performance. *Organization Science, 28*(2), 228-243.

Rigby, D. K., Sutherland, J., & Noble, A. (2018). Agile at scale. *Harvard Business Review, 96*(3), 88-96.

Rigby, D. K., Sutherland, J., & Takeuchi, H. (2016). Embracing agile. *Harvard Business Review, 94*(5), 40-50.

Ringle, C. M., Wende, S., & Becker, J.-M. (2015). *SmartPLS 3*. SmartPLS GmbH. http://www.smartpls.com.

Rutner, P. S., Hardgrave, B. C., & McKnight, D. H. (2008). Emotional dissonance and the information technology professional. *MIS Quarterly, 32*(3), 635-652.

Sarker, S., Xiao, X., Beaulieu, T., & Lee, A. S. (2018). Learning from first-generation qualitative approaches in the IS discipline: An evolutionary view and some implications for authors and evaluators (PART 1/2). *Journal of the Association for Information Systems, 19*(8), 752-774.

Sarstedt, M., Hair Jr, J. F., Nitzl, C., Ringle, C. M., & Howard, M. C. (2020). Beyond a tandem analysis of SEM and PROCESS: Use of PLS-SEM for mediation analyses! *International Journal of Market Research*, *62*(3), 288-299.

Schlauderer, S., Overhage, S., & Fehrenbach, B. (2015). *Widely used but also highly valued? acceptance factors and their perceptions in water-Scrum-fall projects.* Paper presented at the 36th International Conference on Information Systems.

Schmeichel, B. J., Vohs, K. D., & Baumeister, R. F. (2003). Intellectual performance and ego depletion: role of the self in logical reasoning and other information processing. *Journal of Personality and Social Psychology*, *85*(1), 33.

Schmidt, K.-H., Hupke, M., & Diestel, S. (2012). Does dispositional capacity for self-control attenuate the relation between self-control demands at work and indicators of job strain? *Work & Stress*, *26*(1), 21-38.

Schwaber, K. (2004). *Agile project management with Scrum*. Microsoft Press.

Schwaber, K., & Beedle, M. (2002). *Agile software development with Scrum*. Prentice Hall.

Setor, T., & Joseph, D. (2019). *When agile means staying: the relationship between agile development usage and individual IT professional outcomes*. Paper presented at SIGMIS-CPR '19.

Soror, A. A., Hammer, B. I., Steelman, Z. R., Davis, F. D., & Limayem, M. M. (2015). Good habits gone bad: Explaining negative consequences associated with the use of mobile phones from a dual-systems perspective. *Information Systems Journal*, *25*(4), 403-427.

Taris, T. W., Schreurs, P. J., & van Iersel-van Silfhout, I. J. (2001). Job stress, job strain, and psychological withdrawal among Dutch university staff: Towards a dual-process model for the effects of occupational stress. *Work & Stress*, *15*(4), 283-296.

Tessem, B. (2014). Individual empowerment of agile and non-agile software developers in small teams. *Information and Software Technology, 56*(8), 873-889.

Tice, D. M., Baumeister, R. F., Shmueli, D., & Muraven, M. (2007). Restoring the self: Positive affect helps improve self-regulation following ego depletion. *Journal of Experimental Social Psychology*, *43*(3), 379-384.

Tripp, J. F., Riemenschneider, C. K., & 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.

Tuomivaara, S., Lindholm, H., & Känsälä, M. (2017). Short-term physiological strain and recovery among employees working with agile and lean methods in software and embedded ICT systems. *International Journal of Human-Computer Interaction, 33*(11), 857-867.

Uy, M. A., Lin, K. J., & Ilies, R. (2017). Is it better to give or receive? The role of help in buffering the depleting effects of surface acting. *Academy of Management Journal, 60*(4), 1442-1461.

van Yperen, N. W., & Hagedoorn, M. (2003). Do high job demands increase intrinsic motivation or fatigue or both? The role of job control and job social support. *Academy of Management Journal, 46*(3), 339-348.

Venkatesh, V., Brown, S. A., & Bala, H. (2013). Bridging the qualitative-quantitative divide: Guidelines for conducting mixed methods research in information systems. *MIS Quarterly, 37*(1), 21-54.

Venkatesh, V., Sykes, T., Chan, F. K., Thong, J. Y., & Hu, P. J. (2019). Children's internet addiction, Family-to-work conflict, and job outcomes: a study of parent-child dyads. *MIS Quarterly, 43*(3), 903-927.

Venkatesh, V., Thong, J. Y., Chan, F. K., Hoehle, H., & Spohrer, K. (2020). How agile software development methods reduce work exhaustion: Insights on role perceptions and organizational skills. *Information Systems Journal, 30*(4), 733-761.

VersionOne. (2019). *13th annual state of agile report*. VersionOne.

Vidgen, R., & Wang, X. (2009). Coevolving systems and the organization of agile software development. *Information Systems Research, 20*(3), 355-376.

Vinekar, V., Slinkman, C. W., & Nerur, S. (2006). Can agile and traditional systems development approaches coexist? An ambidextrous view. *Information Systems Management, 23*(3), 31-42.

Vohs, K. D., Glass, B. D., Maddox, W. T., & Markman, A. B. (2011). Ego depletion is not just fatigue: Evidence from a total sleep deprivation experiment. *Social Psychological and Personality Science, 2*(2), 166-173.

Walsham, G. (1995). Interpretive case studies in IS research: Nature and method. *European Journal of Information Systems, 4*(2), 74-81.

Watson, D., Clark, L. A., & Tellegen, A. (1988). Development and validation of brief measures of positive and negative affect: the PANAS scales. *Journal of Personality and Social Psychology, 54*(6), 1063-1070.

Webb, T. L., & Sheeran, P. (2003). Can implementation intentions help to overcome ego-depletion? *Journal of Experimental Social Psychology, 39*(3), 279-286.

Wiener, M., Cram, W., & Benlian, A. (in press). Algorithmic control and gig workers: A legitimacy perspective of Uber drivers. *European Journal of Information Systems*.

Windeler, J. B., Maruping, L. M., & Venkatesh, V. (2017). Technical systems development risk factors: The role of empowering leadership in lowering developers' stress. *Information Systems Research, 28*(4), 775-796.

Wright, T. A., & Cropanzano, R. (1998). Emotional exhaustion as a predictor of job performance and voluntary turnover. *Journal of Applied Psychology, 83*(3), 486-493.

Yakovleva, M., Reilly, R. R., & Werko, R. (2010). Why do we trust? Moving beyond individual to dyadic perceptions. *Journal of Applied Psychology, 95*(1), 79-91.

Zacher, H., Brailsford, H. A., & Parker, S. L. (2014). Micro-breaks matter: A diary study on the effects of energy management strategies on occupational well-being. *Journal of Vocational Behavior, 85*(3), 287-297.

Zohar, D., Tzischinski, O., & Epstein, R. (2003). Effects of energy availability on immediate and delayed emotional reactions to work events. *Journal of Applied Psychology, 88*(6), 1082-1093.

Appendix A: Definitions

Table A1. Primarily Applied Agile ISD Practices

Practices		Definition
Agile PM practices	Iterative delivery - Iteration planning - Iteration reviews	The process of planning and delivering in an incremental manner. Specifically, the concept that delivery in small chunks provides the team with the ability to generate code and immediately receive feedback from the environment after each iteration. (Tripp et al., 2016)
	Daily standup	A (usually) daily meeting in which all project members meet while standing to encourage brevity. In Scrum, the meeting involves asking and answering: What did I accomplish yesterday? What will I do today? What obstacles are impeding my progress? (Schwaber & Beedle, 2002)
	Retrospectives	Meeting held at the end of the iteration in which the team critically reflects on the last iteration and identifies/implements continuous improvement opportunities (Schwaber & Beedle, 2002).
	Short iterations	In general, the iteration length is fixed and should not exceed 30 days (Schwaber, 2004).
Agile SD practices	Unit testing	Using dedicated test code that one can run (usually automatically) to test the effects of changes to the system. The team performs this testing before team members are allowed to check in code, which allows developers to be sure that they have not broken anything in the system (Beck, 2000).
	Continuous integration	The process of systematically and regularly building and deploying the code to a test server (Duvall, Matyas, & Glover, 2007).
	Coding standards	A set of established norms regarding code-naming and consistency (Beck, 2000).
	Refactoring	A commitment by the team to use practices that lead to removing redundancy, eliminating unused functionality, and refreshing obsolete designs (Fowler & Beck, 1999).

Appendix B: Survey

Table B1. Descriptive Statistics of Survey Respondents

Category	%
Gender	
Female	34.3
Male	65.7
Age	
18-30	21.3
31-40	37.7
41-55	33.3
> 55	7.7
Experience with agile ISD practices	
< 1 year	25.1
1-2 years	24.6
3-5 years	32.4
> 5 years	17.9
Number of agile projects	
One single agile project	36.7
Several agile projects	63.3

Table B2. Measurement Items

Construct	Indicator	Items
The use of agile ISD practices (Tripp et al., 2016) 7-point Likert scale ("strongly disagree" to "strongly agree") and "I don't know" as the eighth option.		
Iterative delivery - Iteration planning - Iteration reviews	ID1	At the beginning of each work cycle, the team and business owners agree on what will be delivered during the work cycle.
	ID2	The team gives input as to how much work can be completed in a work cycle.
	ID3	At the end of each work cycle, the project is assessed against the goals of this work cycle.
Daily standup	DS1	Our team has a short meeting every workday to discuss what is going on with the project.
	DS2	Our team discusses issues together daily.
Retrospectives	RE1	On a regular basis, the team reflects on previous work and looks for ways to improve team performance.
	RE2	At the end of each work cycle, the team asks itself "what went well" and "what could be improved."
Short iterations	SI1	The length of one work cycle is as short as possible but as long as necessary.
	SI2	The length of work cycles is fixed throughout the project.
Unit testing	UT1	We have a separate set of "test" code that is written specifically to test the "real" code.
	UT2	Every programmer in our team is responsible for writing unit tests for the code he or she writes.
Continuous integration	CI1	Members of our team integrate code changes as soon as possible.
	CI2	The team has a process that automatically rebuilds the software several times a day.
Coding standards	CS1	The naming and structure of our code is consistent.
	CS2	Our team uses standards for consistent code formatting.
Refactoring	RF1	Whenever we see the need, we improve the design of the code we have written previously.
	RF2	Every member of the team attempts to improve the structure of the code when making a change.
Ego depletion effect (Chan & Wan, 2012; van Yperen & Hagedoorn, 2003) 7-point Likert scale ("never," "almost never," "rarely," "sometimes," "rather often," "nearly all the time," "daily")		
Feelings of fatigue	FF1	I feel tired.
	FF2	I feel really fatigued at the end of a working day.
	FF3	I feel like my "batteries" are "dead."
	FF4	During the last stage of a working day, I feel too exhausted to perform well.
	FF5	I find it difficult to relax at the end of a working day.

Turnover intention (Leiter et al., 2011) Seven-point Likert scale ("strongly disagree" to "strongly agree")		
Turnover intention	TI1	I plan on leaving my company within the next year.
	TI2	I want to remain in my job.
Perceived workload (Moore, 2000a) Seven-point Likert scale ("strongly disagree" to "strongly agree")		
Perceived workload	PW1	I feel that the number of requests, problems, or complaints I deal with is more than expected.
	PW2	I feel busy or rushed.
Negative affect (Watson et al., 1988) Seven-point Likert scale ("not at all," "a little," "to some degree," "moderately," "more than moderate," "quite a lot," "extremely")		
Negative affect	NA1	During the last few weeks/ during the project, to what extent did you feel distressed?
	NA2	During the last few weeks/ during the project, to what extent did you feel upset?
	NA3	During the last few weeks/ during the project, to what extent did you feel nervous?

Appendix C: Model Validation and Common Method Bias Testing

Table C1. Cross-Loadings

	Construct	Item	1	2	3	4	5	6	7	8	9	10	11	12
1	Daily standups	**DS1**	**0.90**	0.49	0.48	0.49	0.45	0.50	0.35	0.40	-0.11	-0.27	-0.14	-0.14
		DS2	**0.92**	0.55	0.59	0.55	0.42	0.55	0.40	0.43	-0.17	-0.38	-0.25	-0.22
2	Retrospectives	**RE1**	0.58	**0.89**	0.63	0.50	0.42	0.57	0.47	0.53	-0.11	-0.31	-0.11	-0.11
		RE2	0.42	**0.86**	0.55	0.50	0.35	0.47	0.41	0.51	-0.15	-0.24	-0.21	-0.18
3	Iterative delivery	**ID1**	0.52	0.57	**0.88**	0.51	0.47	0.53	0.54	0.45	-0.27	-0.39	-0.27	-0.33
		ID2	0.47	0.55	**0.85**	0.49	0.43	0.53	0.50	0.54	-0.08	-0.21	-0.23	-0.16
		ID3	0.53	0.63	**0.86**	0.52	0.46	0.47	0.44	0.53	-0.14	-0.30	-0.22	-0.19
4	Short iterations	**SI1**	0.50	0.44	0.45	**0.83**	0.41	0.49	0.34	0.38	-0.06	-0.19	-0.11	-0.15
		SI2	0.46	0.52	0.54	**0.86**	0.48	0.54	0.43	0.53	-0.18	-0.28	-0.12	-0.21
5	Unit tests	**UT1**	0.41	0.39	0.43	0.48	**0.87**	0.45	0.47	0.39	-0.14	-0.21	-0.13	-0.23
		UT2	0.43	0.38	0.49	0.44	**0.89**	0.49	0.49	0.49	-0.19	-0.27	-0.13	-0.18
6	Continuous integration	**CI1**	0.50	0.49	0.56	0.47	0.43	**0.86**	0.53	0.55	-0.25	-0.43	-0.23	-0.35
		CI2	0.47	0.52	0.44	0.57	0.47	**0.83**	0.41	0.51	-0.18	-0.22	-0.19	-0.19
7	Coding standards	**CS1**	0.38	0.46	0.54	0.44	0.48	0.52	**0.92**	0.56	-0.19	-0.39	-0.13	-0.32
		CS2	0.38	0.47	0.51	0.41	0.52	0.51	**0.92**	0.60	-0.12	-0.38	-0.12	-0.23
8	Refactoring	**RF1**	0.40	0.52	0.52	0.44	0.45	0.57	0.61	**0.88**	-0.21	-0.32	-0.19	-0.25
		RF2	0.40	0.51	0.50	0.51	0.43	0.53	0.48	**0.85**	-0.26	-0.34	-0.26	-0.20
9	Feelings of fatigue	**FF1**	-0.11	-0.14	-0.17	-0.15	-0.19	-0.22	-0.18	-0.31	**0.82**	0.40	0.49	0.59
		FF2	-0.09	-0.08	-0.13	-0.02	-0.12	-0.18	-0.08	-0.19	**0.79**	0.31	0.40	0.46
		FF3	-0.13	-0.15	-0.14	-0.10	-0.17	-0.20	-0.14	-0.24	**0.86**	0.40	0.47	0.58
		FF4	-0.17	-0.11	-0.20	-0.14	-0.16	-0.21	-0.18	-0.20	**0.86**	0.43	0.50	0.58
		FF5	-0.13	-0.11	-0.14	-0.15	-0.13	-0.23	-0.12	-0.16	**0.82**	0.45	0.49	0.61
10	Turnover intention	**TI1**	-0.32	-0.23	-0.30	-0.25	-0.22	-0.33	-0.33	-0.31	0.50	**0.93**	0.40	0.55
		TI2	-0.35	-0.37	-0.35	-0.28	-0.29	-0.39	-0.44	-0.40	0.40	**0.92**	0.27	0.45
11	Perceived workload	**PW1**	-0.14	-0.06	-0.17	-0.05	-0.02	-0.15	-0.05	-0.12	0.38	0.27	**0.81**	0.33
		PW2	-0.23	-0.22	-0.29	-0.17	-0.20	-0.26	-0.16	-0.29	0.58	0.35	**0.92**	0.46
12	Negative affect	**NA1**	-0.15	-0.13	-0.21	-0.15	-0.15	-0.23	-0.21	-0.19	0.61	0.51	0.40	**0.91**
		NA2	-0.14	-0.15	-0.26	-0.13	-0.20	-0.27	-0.31	-0.25	0.65	0.45	0.44	**0.92**
		NA3	-0.25	-0.17	-0.24	-0.30	-0.29	-0.37	-0.31	-0.26	0.60	0.48	0.41	**0.87**

Table C2. Variance Inflation Factors (VIFs)

Construct	VIF
Turnover intention	
Use of agile PM practices	2.554
Use of agile SD practices	2.619
Feelings of fatigue	2.234
Negative affect	2.065
Perceived workload	1.545
Feelings of fatigue	
Use of agile PM practices	2.527
Use of agile SD practices	2.666
Negative affect	1.629

Perceived workload	1.329
Turnover intention	1.618
Perceived workload	
Use of agile PM practices	2.494
Use of agile SD practices	2.654
Feelings of fatigue	1.963
Negative affect	2.139
Turnover intention	1.656
Use of agile PM practices	
Use of agile SD practices	1.294
Feelings of fatigue	2.307
Negative affect	2.147
Perceived workload	1.542
Turnover intention	1.625
Use of agile SD practices	
Use of agile PM practices	1.233
Feelings of fatigue	2.319
Negative affect	2.107
Perceived workload	1.587
Turnover intention	1.597
Negative affect	
Use of agile PM practices	2.540
Use of agile SD practices	2.636
Feelings of fatigue	1.722
Perceived workload	1.524
Turnover intention	1.553

Appendix D: Interviews

Table D1. Descriptive Statistics of Interviewed Software Developers

No.	Industry	Experience with agile ISD practices (in years)	Age	Agile PM practices: Daily standups	Agile PM practices: Retrospectives	Agile PM practices: Iterative delivery	Agile PM practices: Short iterations	Agile SD practices: Unit testing	Agile SD practices: Cont. integration	Agile SD practices: Coding standards	Agile SD practices: Refactoring
1	Banking	3	29	U	U	U	U	U	P	P	P
2	Banking	3	53	U	U	U	U	N	N	U	N
3	IT service provider	9	39	U	U	U	U	N	P	U	U
4	IT Service provider	2	33	U	U	U	U	U	N	U	U
5	Agriculture/ engineering	5	36	N	P	U	U	U	P	P	P
6	Agriculture/ engineering	7	46	N	P	U	U	P	U	N	U
7	Agriculture/ engineering	7	30	U	U	U	U	P	U	U	U
8	IT consulting	1	24	U	U	P	U	N	N	P	P
9	Agriculture/ engineering	2	31	U	P	P	U	N	N	U	U
10	IT consulting	9	30	U	P	U	U	P	U	P	U
11	IT service provider	3	32	U	U	U	U	U	P	U	U
12	IT service provider (catering)	4	31	U	U	U	U	U	U	U	U
13	IT service provider (catering)	2	31	U	U	U	U	U	U	P	P
14	Web engineering	5	32	U	U	U	U	U	U	U	U
15	Software engineering	2	29	U	U	U	U	U	U	U	U

Note: U = used, N = not used, P = partly used

About the Authors

Lea Mueller is a PhD candidate in management information systems at Technische Universität Darmstadt, Germany. Before beginning her PhD, she worked for several years as a consultant and IT project manager. She currently examines agile software development, agile project management, and digital transformation. Her research has been presented at the International Conference on Information Systems and published in journals such as *HMD Praxis der Wirtschaftsinformatik*.

Alexander Benlian is a full professor of management information systems at Technische Universität Darmstadt, Germany. He earned his PhD at Ludwig-Maximilians-University of Munich and also served as a senior consultant with McKinsey & Company. His research focuses on the digital transformation of work and people, algorithmic management, IT entrepreneurship, and human-computer interaction, with over 200 academic publications in these areas. His work has appeared in *MIS Quarterly, Information Systems Research, Journal of Management Information Systems, Journal of the Association for Information Systems, Entrepreneurship Theory and Practice, Journal of Business Venturing, MIS Quarterly Executive*, and several other journals. He is a senior editor of *European Journal of Information Systems* and serves on the editorial boards of *Business Information Systems & Engineering* and *Electronic Markets*.