



## Information Systems Research

Publication details, including instructions for authors and subscription information:  
<http://pubsonline.informs.org>

### Team-Enacted Use vs. Developer-Needed Use of Agile Practices: How Perceptual (In-)Congruence and Team Feedback-Seeking Shape Developer Well-Being

Alexander Benlian, Marc Pinski, Martin Adam

To cite this article:

Alexander Benlian, Marc Pinski, Martin Adam (2025) Team-Enacted Use vs. Developer-Needed Use of Agile Practices: How Perceptual (In-)Congruence and Team Feedback-Seeking Shape Developer Well-Being. *Information Systems Research* 36(4):2253-2277. <https://doi.org/10.1287/isre.2023.0402>

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. You are free to download this work and share with others, but cannot change in any way or use commercially without permission, and you must attribute this work as “*Information Systems Research*. Copyright © 2025 The Author(s). <https://doi.org/10.1287/isre.2023.0402>, used under a Creative Commons Attribution License: <https://creativecommons.org/licenses/by-nc-nd/4.0/>.”

Copyright © 2025 The Author(s)

Please scroll down for article—it is on subsequent pages



With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes.

For more information on INFORMS, its publications, membership, or meetings visit <http://www.informs.org>

# Team-Enacted Use vs. Developer-Needed Use of Agile Practices: How Perceptual (In-)Congruence and Team Feedback-Seeking Shape Developer Well-Being

 Alexander Benlian,<sup>a,\*</sup> Marc Pinski,<sup>a</sup> Martin Adam<sup>b</sup>

<sup>a</sup>Chair for Information Systems & E-Services, Department of Business, Law and Economics, Technical University of Darmstadt, 64289 Darmstadt, Germany; <sup>b</sup>Chair for Smart Services, Faculty of Business and Economics, Georg August University Göttingen, 37073 Göttingen, Germany

\*Corresponding author

Contact: [benlian@ise.tu-darmstadt.de](mailto:benlian@ise.tu-darmstadt.de),  <https://orcid.org/0000-0002-7294-3097> (AB); [marc.pinski@gmail.com](mailto:marc.pinski@gmail.com),

 <https://orcid.org/0000-0003-2464-9026> (MP); [martin.adam@uni-goettingen.de](mailto:martin.adam@uni-goettingen.de),  <https://orcid.org/0000-0001-9369-7203> (MA)

Received: June 30, 2023

Revised: October 24, 2023; June 3, 2024; February 15, 2025

Accepted: March 17, 2025


Published Online in Articles in Advance: April 14, 2025

<https://doi.org/10.1287/isre.2023.0402>

Copyright: © 2025 The Author(s)

**Abstract.** Despite significant progress in understanding how the use of agile practices in teams affects crucial developer outcomes, we still know little about whether the team-enacted use of agile practices (e.g., daily stand-up meetings or pair programming) always aligns with the developer-needed use of such practices. In other words, does incongruence between the team-enacted versus individually needed use of agile practices matter for developer well-being? Additionally, how does developers' team feedback-seeking behavior shape how developers respond to congruence and incongruence? Drawing on person-environment fit theory, we shed light on developers' perceived (in-)congruence between the team-enacted versus individually needed use of agile practices and the resulting consequences for their well-being. In an experience-sampling study with 149 developers and 1,510 daily-level responses and by using multilevel polynomial regression and response surface methods, we show that perceived congruence (versus incongruence) consistently leads to higher levels of developer well-being. We also find that not all instances in which developers perceive high levels of team-enacted use of agile practices are equivalent. In fact, depending on the developer-needed use of agile practices in their daily work, enacting a high level of agile practices in teams can be just as detrimental as enacting a low level, revealing that the experience of agile practices use is more nuanced than previously recognized. Our findings also reveal an asymmetry; frequent (versus infrequent) team feedback-seeking developers experience amplified well-being losses from incongruence but not amplified well-being gains from congruence. In follow-up interviews with 32 developers from the experience-sampling study, we corroborate the main findings of our quantitative study and provide complementary insights into important boundary conditions of incongruence effects. Overall, these insights offer important implications for research on agile information systems development and for organizations seeking to align team-enacted agile practices with developers' individual needs and feedback-seeking behavior in a human-centered and sustainable way.

**History:** Rajiv Kohli, Senior Editor; Christy Cheung, Associate Editor.

 **Open Access Statement:** This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. You are free to download this work and share with others, but cannot change in any way or use commercially without permission, and you must attribute this work as "Information Systems Research. Copyright © 2025 The Author(s). <https://doi.org/10.1287/isre.2023.0402>, used under a Creative Commons Attribution License: <https://creativecommons.org/licenses/by-nc-nd/4.0/>."

**Funding:** The authors' work is funded by the German Research Foundation.

**Supplemental Material:** The online appendix is available at <https://doi.org/10.1287/isre.2023.0402>.

**Keywords:** agile practices • team-enacted use • developer needs • team feedback-seeking • well-being • person-environment fit theory • experience sampling • multilevel polynomial regression • response surface analysis • interviews

Rigorous processes are designed to standardize people to the organization, while agile processes are designed to capitalize on each individual and each team's unique strengths: One-size-fits-one—every process must be selected, tailored, and adapted to the individuals on a particular project team. (Cockburn and Highsmith 2001, p. 132)

## Introduction

Although agile practices, such as pair programming or daily stand-ups, have increasingly found their way into information systems development (ISD) projects over the last two decades, agile ISD projects still suffer from alarmingly high failure rates—with estimates ranging

from 50% to 96% (Milne 2022)—that may impede organizations' pursuit of digital transformation (Toutaoui et al. 2022, Carroll et al. 2023). One of the foremost human-related challenges faced by agile projects is that developers often experience an imbalance between the team-enacted use of agile practices and their own needed use of agile practices (Luong et al. 2021). These deviations can leave developers feeling overwhelmed by excessive meetings and coworking sessions on some days, while experiencing frustration because of the absence of suitable agile practices on others. Such imbalances can irritate and deplete developers, harming their well-being and hampering their work productivity. If left unchecked, they may even culminate in serious health consequences (e.g., work exhaustion and burn-out), undermine sustained team effectiveness, and derail agile ISD projects altogether (van Oorschot et al. 2018, Junker et al. 2024).

Existing individual-level research on agile ISD has typically put the spotlight on the extent of agile practices use, focusing on the enactment of these practices in teams and the resulting impact on developer outcomes. For example, Tripp et al. (2016) found that the use of agile project management and software development practices enacted in teams increased developers' job satisfaction via their perceptions of job characteristics (e.g., autonomy). Venkatesh et al. (2020) showed how the team-enacted use of agile practices helped developers alleviate work exhaustion by achieving clear and unambiguous role perceptions, especially if the developers had strong organizational skills. Windeler et al. (2017) found that technical ISD risk factors (e.g., project size) can heighten developers' stress levels in agile software teams by increasing role ambiguity and role conflict. More recent studies (e.g., Benlian 2022, Mueller and Benlian 2022) have revealed that using agile practices in teams can deplete developer well-being, with this effect being shaped by project characteristics (e.g., perceived workload) and developer traits (e.g., information technology mindfulness).

Despite these valuable contributions to the agile ISD literature, it is essential to highlight three prevailing assumptions in the literature on agile practices use that warrant deeper inquiry. First, existing conceptualizations of agile practices use implicitly assume that developer needs are perfectly reflected in the developer team's enacted use of agile practices, predicated on the belief that developers' direct involvement in choosing and tailoring agile methods results in the seamless integration of personal preferences into team-level practices (e.g., Venkatesh et al. 2020, Benlian 2022). However, the practical reality in agile project teams tells a different story, where the developer-needed use of agile practices can often (temporarily or persistently) diverge from the team-enacted use because of individual-, team-, and project-level factors, such as personal working styles,

team diversity, and dynamically shifting project priorities (e.g., Masood et al. 2022, Nazir et al. 2024). Such potential incongruences<sup>1</sup> are concealed in the current one-dimensional conceptualizations of agile practices use, hindering us from better understanding the internal conflicts that developers face when evaluating the team-enacted versus individually needed use of agile practices. Consequently, our ability to discern the unique effects of these tensions on crucial developer health and performance outcomes is compromised. As such, if developer needs are not explicitly accounted for in our study of agile practices use, we run the risk of ignoring the fundamental plea for a one-size-fits-one philosophy made by the two Agile Manifesto signatories in our opening quotation, and the potential adverse effects of agile practices use on developers may go unnoticed.

Second, the prevailing research on agile ISD operates under the presumption that the use of agile practices in teams and the developer-needed use of such practices are relatively static and do not change over short-term intervals (e.g., Tripp et al. 2016, Kude et al. 2019). However, the temporal complexity inherent in contemporary, large-scale agile work environments, characterized by continuous development, event-based scheduling, and rapid cycle times (e.g., Meckenstock 2024, O'Connor et al. 2024), challenges this notion. Such dynamism can lead to daily fluctuations in the team-enacted and developer-needed use of agile practices, with consequential short-term effects on developer well-being and performance. Recognizing and capturing these proximal effects may present an opportunity to advance our understanding of agile practices' temporal dynamics and the merits of well-timed interventions.

Third, scholars and practitioners have long recognized proactive feedback-seeking as a cornerstone of effective teamwork, especially in agile settings (e.g., Ashford and Cummings 1983, Junker et al. 2022). Team feedback-seeking, defined as the intentional pursuit of information from team members about one's performance, adherence to expectations, and team rapport, fosters learning, team affiliation, and a sense of control (Ashford and Black 1996, Callister et al. 1999). Although fundamental agile principles and literature on agile ISD suggest that team feedback seeking should naturally complement agile practices (e.g., Junker et al. 2022, Tripp and Sambamurthy 2023), this assumption overlooks its role in shaping perceptions of and reactions to (in-)congruence. Developers who frequently seek team feedback may rely more heavily on team input and become more attuned to misalignments between team-enacted and individually needed agile practices, making even minor discrepancies more salient and taxing. Conversely, developers who rarely seek feedback tend to operate with greater independence and self-sufficiency, making them less vulnerable to incongruence effects. Viewing team feedback-seeking as an

indicator of incongruence sensitivity presents an opportunity to identify it as a key boundary condition that explains why developers vary in their susceptibility to incongruence, deepening theoretical and practical insights into agile teamwork dynamics.

In light of these challenges to our currently held assumptions, our study seeks to understand the effects of (in-)congruence between the team-enacted and individually needed use of agile practices on developer well-being and how developers' team feedback-seeking behavior shapes these relationships. We focus on developer well-being as an outcome as it aptly captures the immediate day-to-day effects of the (in-)congruence between what individuals need and what they actually use in their environment (Edwards et al. 1998, Diener et al. 1999), with wider implications for work performance and job satisfaction (Wright and Cropanzano 2000). Accordingly, we ask the following questions.

*How do daily congruence and incongruence in the team-enacted versus individually needed use of agile practices affect developer well-being? Additionally, how pronounced is the influence of this congruence and incongruence among frequent (versus infrequent) team feedback-seeking developers?*

In our pursuit to address these research questions, we develop a dynamic theory of agile practices use rooted in the person-environment (P-E) fit theory that provides a unique lens through which to discern the (in-)congruence between the team-enacted and individually needed use of agile practices. In an experience-sampling (ES) study with 149 agile developers and 1,510 daily-level responses, we employ multilevel polynomial regression and response surface methods to uncover the distinct impacts of congruence and incongruence on developer well-being, and, as our supplementary analyses show, on broader work and job outcomes. We also reveal that frequent (versus infrequent) feedback-seeking developers incur greater well-being losses when the team-enacted use of agile practices diverges from developer-needed use. Follow-up interviews with developers of the experience-sampling study provide supporting evidence for our theorizing and complementary insights into important boundary conditions.

We seek to advance the field by offering several key contributions. First, drawing on key notions of the P-E fit perspective, we offer a refined theoretical conceptualization that reveals the importance of explicitly accounting for developer needs in the study of agile practices use to recognize potentially consequential incongruence effects. Moving beyond the existing one-dimensional view of agile practices use (e.g., Venkatesh et al. 2020, Benlian 2022), our study thus provides a novel perspective for examining the (in-)congruence between the team-enacted and individually needed use of agile practices. Second, departing from the prevailing assumption of short-term stability in the team-enacted and developer-needed use of agile practices (e.g., Tripp

et al. 2016, Kude et al. 2019), we introduce a novel micro-temporal lens that unveils how daily fluctuations in the (in-)congruence of the team-enacted versus individually needed use of agile practices can have immediate effects on developer well-being, ultimately influencing satisfaction and performance outcomes. Third, our study challenges the prevalent belief in the literature that feedback-seeking and agile practices are always complementary (e.g., Junker et al. 2022, Tripp and Sambamurthy 2023), showing instead that high (versus low) levels of team feedback-seeking can hurt developer well-being in situations of incongruence. Supplementary exploratory and corroborating qualitative insights from our follow-up interviews further reveal that the adverse effects of incongruence are particularly pronounced in large-scale agile projects, underscoring the relevance of our findings in contemporary, complex project environments. Finally, as our findings highlight the critical role of aligning developer needs to improve well-being and performance in agile ISD projects, the practical implications of this study emphasize the adoption of strategies to proactively prevent, mitigate, or resolve perceptual incongruences, enabling a more sustainable and human-centered approach to agile practices.

## Theory and Hypothesis Development

We begin by reviewing the relevant literature on the implications of agile practices use for developers. Next, we provide a brief overview of P-E fit theory, the foundation for our dynamic theory on how daily (in-)congruence between team-enacted and individually needed agile practices affects developer well-being.

### Agile Practices Use and Its Effects on Developers

Since the advent of the new millennium, agile ISD methods, such as Scrum, extreme programming (XP), lean development, and Kanban, have evolved in response to growing frustration with the historically low success rates of software projects (Fitzgerald et al. 2006, Ågerfalk et al. 2009). Rooted in the values and principles of the Agile Manifesto, these methods prioritize people and face-to-face communication, high-quality working software, active customer involvement, and adaptability to change (Beck et al. 2001, Mueller et al. 2025). Agile teams typically operate in short cycles, iteratively delivering functional software while adjusting to the evolving project landscape (Vidgen and Wang 2009). Despite sharing a common philosophical foundation, each agile method has distinct origins and foci, such as XP's emphasis on software development and Scrum's project management orientation. In line with earlier research (Sarker et al. 2018), we refer to agile techniques, such as daily stand-ups, pair programming, and collective code ownership, as "practices." In contrast, "methods"

encompass various clearly defined, interconnected sets of practices used by developers, including Scrum, XP, and Kanban. In light of this distinction, our research adopts an aggregate perspective on agile practices, considering their combined influence on developers' experiences and behaviors across various methods. In other words, we conceptualize the use of agile practices as the application of a range of agile project management, software development, and process improvement techniques within agile teams (Tripp et al. 2016). These practices are typically enacted at the team level, reflecting the interdependence of tasks among developers, evident in practices such as pair programming, daily stand-ups, and collective code ownership.

Early information systems (IS) research on agile ISD methods focused on the effectiveness of the use of agile practices vis-à-vis traditional, plan-based ISD approaches, including how agile practices were tailored and customized to the specific context of agile teams before they were put into use (e.g., Fitzgerald et al. 2006, Wang et al. 2012). With the increasing adoption and intensified use of agile practices, scholarly focus has gradually shifted to understanding their effects on team and project outcomes. For example, Maruping et al. (2019) explored their impact on software project quality, whereas Ramesh et al. (2012) examined the development of contextual ambidexterity in agile distributed teams. Although the adoption of agile practices has continued to proliferate in recent years, with 71% of organizations reporting using them in their software development life cycle (digital.ai 2023), firms have become disillusioned by the inconsistent results that agile teams still deliver in reality, questioning the purported benefits of agile practices in terms of speed, quality, and customer satisfaction (Masood et al. 2022). Among the main reasons, the mixed results are attributed to human-related challenges, including lack of motivation, poor communication, and neglect of individual needs (Conboy et al. 2011, Meckenstock 2024). Indeed, poor alignment between the team-enacted and developer-needed use of agile practices can produce developer stress and impair team performance (Singh and Strobel 2022).

More recent IS studies have thus increasingly gravitated toward the implications of agile practices use for developer outcomes, including job satisfaction, work exhaustion, and developer stress, thereby underscoring the critical role of developers in agile project success. Tripp et al. (2016) provided evidence that the adoption of agile practices in teams positively influence developers' job satisfaction by shaping their perceptions of job characteristics. In a contrasting vein, Windeler et al. (2017) highlighted the potential drawbacks of agile team environments, demonstrating how role ambiguity and conflict can exacerbate stress levels among agile developers, thereby hindering team performance. Building on

this, Venkatesh et al. (2020) demonstrated the importance of developers' organizational skills, essential for stakeholder alignment, in reducing these role-related stressors and subsequently, decreasing work exhaustion. Extending this line of inquiry, recent studies by Benlian (2022) and Mueller and Benlian (2022) have revealed that the daily (positive or negative) appraisals of agile practices and the perceived workload can significantly influence whether using these practices in teams is beneficial or burdensome for developers.

Although these studies have offered valuable insights, the prevalent conceptualizations of agile practices use have primarily adopted a one-dimensional perspective centered on the extent of developers' use of agile practices in their teams (e.g., Tripp et al. 2016, Venkatesh et al. 2020), implicitly assuming that it is sufficient to consider what is "enacted" in a team environment. However, this prevailing focus on agile practices tends to sideline a crucial dimension: the individual needs of developers. Developers typically derive meaning by comparing how well their team's use of agile practices aligns with their individual needs around using such practices in their daily tasks. Recognizing the importance of this comparative evaluation is crucial as it emphasizes potential incongruences and highlights the profound implications that these mismatches can have for personal and professional outcomes (Fischer et al. 2023).

Although the prevailing one-dimensional view of agile practices use implies that individual developer needs are perfectly factored into the team-enacted use of agile practices, the practical realities often diverge from this idealized depiction because of individual-, team-, and project-level factors (Masood et al. 2022, Nazir et al. 2024).<sup>2</sup> For example, developers' need for uninterrupted focus time for refactoring code may clash with team-enacted practices of frequent check-ins, whereas differences in skill levels and seniority within teams can create inconsistencies in interpreting agile practices. Additionally, shifting project priorities may pressure teams to emphasize immediate delivery over the reflective practices crucial for developers' growth. Recognizing these sources of incongruence, individual developer needs may coincide with the team-enacted use of agile practices at some points in time but diverge significantly at others, reflecting a dynamic interplay between personal needs and actual team practice. In this vein, although daily stand-ups may meet a developer's needs one day, they may be perceived as excessive on another. Considering such variance in how the daily enactment of agile practices in teams aligns with (congruence) or diverges from (incongruence) an individual developer's needs regarding the use of such practices opens up new avenues for theorizing the effects of agile practices use on developer outcomes. Understanding how the intricate dynamics of congruence and incongruence can shape developer well-being and work

outcomes also plays a pivotal role in anticipating the broader impact of agile practices on sustained team effectiveness and the success of larger transformation programs as well as in crafting targeted intervention initiatives.

In summary, we argue that focusing solely on the enactment of agile practices within teams provides an incomplete picture. Overlooking individual developer needs poses an increasingly pressing problem, particularly in light of the acknowledged human-related challenges in agile ISD projects (Conboy et al. 2011, Luong et al. 2021). To genuinely embrace a “one-size-fits-one” approach in agile ISD, it is thus imperative to understand how both the team-enacted *and* developer-needed use of agile practices influence developer well-being. Our study seeks to theoretically integrate this dual emphasis by examining agile practices use through a P-E fit perspective.

### **A Person-Environment Fit Perspective on the Team-Enacted vs. Developer-Needed Use of Agile Practices**

The core tenet of the P-E fit theory, grounded in the reasoning that attitudes and behaviors result not from the person or environment in isolation but rather from the relationship between the two (Edwards et al. 1998), suggests that optimal functioning and positive outcomes (such as well-being, job satisfaction, and performance) occur when there is a high degree of compatibility or correspondence between an individual and their environment. This compatibility can be in terms of the skills, needs, values, or personality traits of the individual aligning with the demands, resources, practices, or culture of the environment (such as a team or organization). This dual emphasis on the person and the environment is characteristic of the interactional perspective in psychology, which indicates that behavior, attitudes, and well-being are determined jointly by the person and the environment (Bandura 1989).

Within the P-E fit paradigm, the alignment between an individual’s needs and the organizational practices enacted in his or her environment (e.g., work setting) is a key determinant of outcomes (Kristof 1996). Needs, developed through experiences and social interactions, are the individual’s desired practice levels that contribute to achieving higher-order goals, such as well-being and personal growth (Edwards and Parry 1993, Kristof-Brown et al. 2005). In contrast, the practices enacted in the individuals’ collective environment (e.g., in teams) and used by these individuals in their daily work constitute crucial elements that interact with these needs. The fundamental psychological process underlying P-E fit theory involves the individual’s cognitive assessment of how the practices enacted in their environment compare with individuals’ needed use of such practices (Edwards et al. 2006), such as developers assessing the

fit between the team-enacted use of agile practices and their own needs around the use of such agile practices. This comparison and the resulting congruence or incongruence play a significant role in shaping an individual’s well-being and behavior. Previous P-E fit literature typically refers to the term “congruence” for a condition in which both amounts are perceived to match or fit, whereas “incongruence” indicates any level of divergence between both amounts, ranging from slight differences to major mismatches (Edwards 1994, 2002). Taken together, the P-E fit theory serves as a useful theoretical lens through which we can precisely discern and articulate how the enactment of agile practices within team environments meets or diverges from the developer-needed use of such practices.

Applying the P-E fit framework to our research context, with agile practices as the focal resource, we refer to the “environment” as developers’ teams in which agile practices (e.g., daily stand-ups, pair programming, and collective code ownership) are collectively enacted on an ongoing basis and to the “person” as the developers working in these teams. The collective enactment of agile practices is largely a product of collaborative work in self-organizing teams, where the extent of usage by developers is shaped by team-level agreements and established routines (Baham and Hirschheim 2022). However, developers’ use of agile practices may deviate from their individual needs because of personal, team, or project factors (Masood et al. 2022, Nazir et al. 2024). Drawing from this notion and in line with the P-E fit perspective, developers compare the extent of the team-enacted use of agile practices with their specific needs in daily work.<sup>3</sup> This nexus between the environmental aspect (team-enacted use) and the personal aspect (developer-needed use) co-constitutes states of congruence and incongruence, with important well-being implications. In P-E fit theory, well-being, defined as individuals’ cognitive and emotional evaluations of their lives (Diener et al. 1999), represents a well-suited outcome for gauging the short-term effects of congruence and incongruence between the team-enacted and individually needed use of agile practices (Edwards et al. 1998). Because well-being is particularly sensitive to discrepancies between actual and desired states, it serves as a proximal indicator of stress and strain in response to these conflicts (Edwards and Shipp 2007) and of potential domino effects on work performance and job satisfaction (Wright and Cropanzano 2000).

As a conceptual framework to guide our theorizing on the effects of (in-)congruence in the team-enacted versus individually needed use of agile practices on developer well-being, and in line with previous models of congruence (Joyce et al. 1982, Fry and Smith 1987), we propose a two-by-two matrix (see Figure 1); it juxtaposes the two dimensions of P-E fit and delineates the extent of the team-enacted use versus the developer-needed

**Figure 1.** (In-)Congruence Between the Team-Enacted and Developer-Needed Use of Agile Practices

		Team-Enacted Use of Agile Practices	
		Low	High
Developer-Needed Use of Agile Practices	High	<b>High Needs – Low Use Incongruence (Deficiency)</b> Developer experiences a <b>deficiency</b> of agile practices <i>(Quadrant 1)</i>	<b>High Needs – High Use Congruence</b> Developer experiences that his/her needs are met at high levels <i>(Quadrant 2)</i>
	Low	<b>Low Needs – Low Use Congruence</b> Developer experiences that his/her needs are met at low levels <i>(Quadrant 3)</i>	<b>Low Needs – High Use Incongruence (Excess)</b> Developer experiences an <b>excess</b> of agile practices <i>(Quadrant 4)</i>

use of agile practices (hereafter, “agile practices used versus needed”) to differentiate congruence (i.e., at high and low levels) (quadrants 2 and 3 in Figure 1) from incongruence (i.e., where the extent of agile practices used in teams falls short of or exceeds the developer-needed use of agile practices) (quadrants 1 and 4 in Figure 1). Below, we theorize about the differences between congruence and incongruence in their daily effects on developer well-being before hypothesizing whether and how the two types of incongruence (deficiency versus excess) vary in terms of their impact on developer well-being.

### Daily Effects of Congruence vs. Incongruence

According to P-E fit theory, individuals in a workplace regularly evaluate how organizational practices are enacted in their environment, comparing this with their personal needs for those practices (Edwards and Harrison 1993). When there is alignment between personal needs and enacted practices, individuals tend to experience better well-being and exhibit a more positive attitude. With congruence, in other words, individuals should find that their work environment supports or complements their personal needs, enhancing their motivation and reducing psychological stress and strain (Cable and Edwards 2004). Conversely, incongruence between personal needs and organizational practices can result in increased stress and anxiety as individuals struggle to adapt to an environment that is not sufficiently attuned to their personal needs. Drawing upon these arguments, we expect that congruence between the extent of agile practices used versus needed will result in higher levels of developer well-being than incongruence.

When developers experience that their individual needs are aligned with the team-enacted use of agile practices (see quadrants 2 and 3 in Figure 1), they feel that congruence and harmony exist. Much like a soccer

player who feels empowered when the tactics and practices of their team resonate with their individual needs, developers should experience a sense of understanding and empowerment. For example, if developers are on the same page as their team in terms of the extent to which they use daily stand-ups or pair programming sessions, they should feel comfortable and aligned. Such congruence between the team-enacted and developer-needed use of agile practices likely bolsters their confidence and self-efficacy. Similarly, congruence may also help developers mitigate frustration and stress as they feel more competent and valued in their daily work, which likely enhances their well-being. Previous research has also shown that employees experience increased well-being when team-enacted work practices closely match their individual needs (Edwards and Shipp 2007, Rietze and Zacher 2022).

In contrast, when developers find that their needs are out of step with the team-enacted use of agile practices (see quadrants 1 and 4 in Figure 1), they may perceive this incongruence as frustrating and stressful. Staying with the above analogy, just as soccer players can feel irritated when their team’s tactics and moves do (at least temporarily) not align with their individual preferences, developers may experience similar frustrations or feel overwhelmed when team-enacted agile practices clash with their personal needs. Developers may be annoyed and stressed if they feel that the time (e.g., dailies, sprints, and time-boxing) and place (e.g., colocation, open office space, and face-to-face meetings) requirements enacted through agile practices are not in line with their individual needs for using these practices at a given moment (Wiesche 2021, Kude et al. 2023). This misalignment may, for example, be further compounded by tensions between individual and collective code ownership as developers often view their code as a deeply personal extension of their identity (Nazir et al. 2024). Such incongruence can create inner conflict as

developers balance meeting team-enacted requirements against their own situational needs, potentially undermining their sense of autonomy and well-being. Meta-analytic evidence also demonstrates that a mismatch between the organizational practices enacted at employees' workplaces and their individual needs for these practices—manifesting as either an excess or a deficit—can lead to detrimental effects on both their health and performance (Kristof-Brown et al. 2005).

Taken together, we posit that congruence in the extent of agile practices used versus needed leads to higher levels of developer well-being compared with incongruence. Conversely, we propose that developer well-being will be lower when the extent of agile practices used versus needed diverges rather than matches.

**Hypothesis 1.** *Daily developer well-being will be higher when the extent of team-enacted use and developer-needed use of agile practices are congruent compared with when they are incongruent.*

Having contrasted the effects of congruence versus incongruence, we now consider the two cases of incongruence, namely deficiency (quadrant 1 in Figure 1) and excess (quadrant 4 in Figure 1), and their potentially diverging effects on developer well-being. Is it more harmful to developers' well-being when they experience that the team-enacted use of agile practices is insufficient and falls below their needed use of such practices or when it exceeds their needs? We expect that a deficiency will be worse than an excess. We contend that when developers perceive a shortfall in the team-enacted use of agile practices compared with their individually needed use, this mismatch can have a more pronounced negative impact on their well-being than in opposite situations. According to the P-E fit perspective, individuals are usually more sensitive to a lack of critical resources in their work environment than to a surplus, underscoring the asymmetric impact of resource scarcity versus abundance on employee responses (Edwards 1994). Indeed, prior research has predominantly shown that individuals experience greater psychological strain and emotional distress when faced with deficiencies rather than excesses. Deficiencies evoke feelings of deprivation and abandonment, which usually outweigh the feelings of abundance and overload associated with excess situations (e.g., Jansen and Kristof-Brown 2005).

Correspondingly, we argue that developers facing a shortfall in the team-enacted use of agile practices are likely to experience more detrimental effects as they might feel directionless or constrained in their ability to accomplish tasks effectively. For example, a lack of or insufficient structure and timely guidance provided through daily stand-up meetings or pair programming sessions can leave developers feeling lost and adrift. Deficient levels of team-enacted agile practices may

thus contribute to a sense of helplessness and exacerbate concerns that developers' teams are disconnected from or even ignore their specific needs. In contrast, developers confronted with an excess of team-enacted agile practices may grapple with feelings of being overwhelmed or of micromanagement when facing an onslaught of meetings, standards, and code releases in their daily tasks. Indeed, research indicates that an overuse of organizational practices in teams, even those with positive intentions, can lead to fatigue, imposing additional cognitive and emotional burdens on employees (Vogel et al. 2020). However, despite the potential drawbacks of excessive team-enacted agile practices, developers may maintain their confidence in handling their responsibilities as they would still benefit from the available guidance and work structure that agile practices provide for task completion. An abundance of team-enacted agile practices may also signal a commitment to supporting developers, albeit to an excessive degree, which can foster a sense of security and belonging, offsetting some negative effects on well-being.

For these reasons, we posit that exceeding developers' needs should cut less deeply into developers' emotions and well-being than falling short in this regard. We thus propose an asymmetric incongruence effect wherein incongruence between agile practices used versus needed is less detrimental to developer well-being when the team-enacted use of agile practices exceeds developers' needs than when it falls short of their needs.

**Hypothesis 2.** *Daily developer well-being will be lower when the extent of the team-enacted use of agile practices falls short of developers' needs for these practices compared with when it exceeds their needs.*

### The Moderating Role of Developers' Team Feedback-Seeking Behavior

In line with the P-E fit perspective, individuals' perception of (mis-)fit between their personal needs and the resources in their environment does not occur in a vacuum; rather, individual differences can amplify or buffer the effects of congruence or incongruence (Edwards et al. 1998). In other words, personal characteristics may magnify or attenuate the consequences of larger gaps in incongruence compared with smaller ones. In particular, proactive feedback practices, such as developers' team feedback-seeking behavior, have been shown to play a vital role in fostering team performance and software quality (e.g., Junker et al. 2022, Tripp and Sambamurthy 2023). Defined as an individual's goal-oriented pursuit of information from team members about their performance, alignment with expectations, and team rapport, team feedback-seeking fosters learning, team affiliation, and a sense of control (Ashford and Black 1996, Callister et al. 1999). Feedback-seeking resonates with agile methodologies' foundational principles

of continuous adaptation and rapid feedback cycles, underscoring the proactive role that developers play in agile ISD (e.g., Williams and Cockburn 2003, Schwaber and Sutherland 2020).

Prior research indicates that feedback-seeking should be a natural complement to agile practices (e.g., Ribeiro and Alves 2021, Junker et al. 2022, Tripp and Sambamurthy 2023), suggesting that frequent team feedback-seeking developers should draw beneficial outcomes from the use of agile practices. However, by largely assuming a consistent complementarity between agile practices and team feedback-seeking, the existing literature may overemphasize the positive outcomes of this interaction. Our study challenges this assumption by examining whether the (in-)congruence effects of agile practices on developer well-being vary based on team feedback-seeking, offering new insights into the boundary conditions of agile practices use (e.g., Venkatesh et al. 2020, Benlian 2022).

At the core of our theorizing is the idea that developers' team feedback-seeking differentially shapes the (in-)congruence effects of the team-enacted versus developer-needed use of agile practices on well-being. We argue that team feedback-seeking serves as an indicator of incongruence sensitivity, influencing how attuned developers are to misalignments between team practices and their needs. Frequent feedback seekers tend to rely more on team input to ensure alignment with collective norms. As a result, even minor deviations from their needs may be perceived as disruptive and problematic, intensifying the negative well-being impact of incongruence. For example, consider a developer who values individual code ownership, seeking control and prolonged refinement of "their" code, but whose team adopts a collective code ownership model emphasizing shared responsibility and frequent code reviews (Nazir et al. 2024). This misalignment may trigger inner conflict and frustration, particularly for frequent feedback seekers, whose heightened sensitivity amplifies the strain of incongruence. In contrast, infrequent feedback seekers may be more resilient to such misalignments as they are more self-sufficient and less dependent on external alignment cues.

On the other hand, it may seem intuitive that frequent (versus infrequent) team feedback-seeking developers would experience enhanced well-being when team-enacted agile practices align with their personal needs as this congruence reinforces their sense of coherence with the team's expectations, minimizing potential sources of uncertainty or conflict. However, we argue that developers, regardless of frequency in team feedback-seeking, tend to be less sensitive to congruence than to incongruence because such alignment confirms rather than challenges their sense of fit within the team. When developers feel secure in their alignment with team practices, there is no internal tension to process,

making a heightened emotional response unlikely. This absence of dissonance reduces the intensity of any positive impact on well-being as congruence is seen as an expected and natural state rather than a positive deviation.

Our rationale for the differential impact of team feedback-seeking in shaping congruence and incongruence effects is grounded in the "bad is stronger than good" principle of Baumeister et al. (2001). This principle, widely observed across various life domains, suggests that individuals tend to react more strongly to negative experiences than to positive ones. This heightened sensitivity to negative events is rooted in an innate instinct for self-preservation, prompting individuals to be more vigilant in the face of potential threats. In the context of agile development, frequent team feedback-seeking developers may view incongruence as a "bad" experience that disrupts their sense of alignment within the team, evoking heightened emotional responses and strain. In contrast, situations of congruence—where the team-enacted and developer-needed use of agile practices align—may be perceived as "good" but stable and therefore, not requiring the same level of vigilance or emotional reaction. For frequent team feedback seekers, congruence may simply confirm their expectations, yielding a sense of alignment without a substantial boost in well-being. Taken together, we therefore propose that frequent (versus infrequent) team feedback-seeking developers should not benefit from a well-being premium under congruence, inasmuch as they bear a well-being loss under incongruence.

**Hypothesis 3a.** *The effects of incongruence on developer well-being are more adverse for frequent (versus infrequent) feedback-seeking developers.*

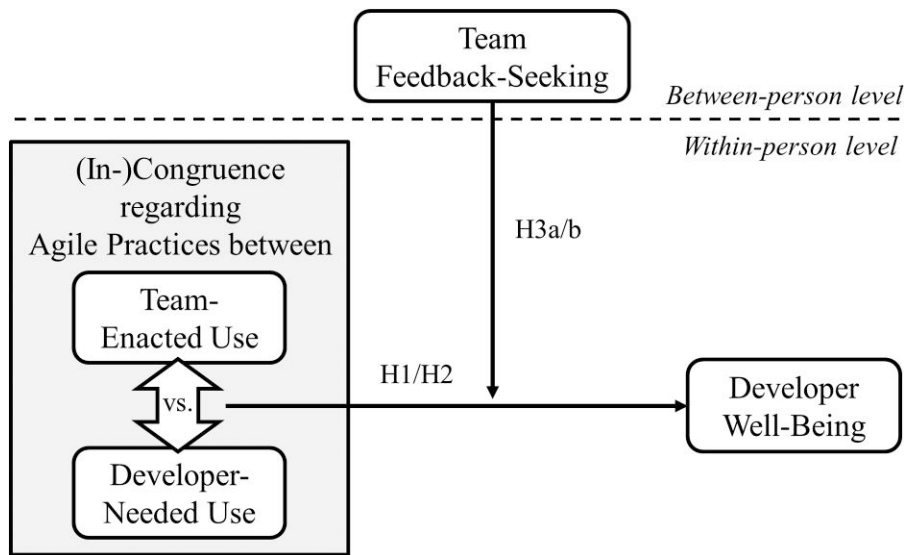
**Hypothesis 3b.** *The effects of congruence on developer well-being are not more beneficial for frequent (versus infrequent) feedback-seeking developers.*

## Methodology

### Participants

We collected daily survey data from 149 agile software development professionals to assess our research model as shown in Figure 2. We specifically targeted agile software developers within teams who adhered to the core roles outlined in the Scrum Guide (Schwaber and Sutherland 2020), namely product owners, Scrum masters, and developers. In these agile teams, the concept of a "formal team leader" is not present. Instead, these teams are designed to be self-managing and self-organizing, particularly in relation to the use of agile practices. Therefore, our target group of agile developers should be part of teams where decision-making and the enactment of agile practices were a collective responsibility. According to this sampling frame, we recruited

Figure 2. Research Model



Note. H, hypothesis.

participants via our industry partners in the software development sector, utilizing a snowballing technique in which our partners advertised our study in their agile software development units. Snowballing is a widely used data collection technique in ES studies, and it is particularly effective in daily data collection, where participants face a high response burden (Gabriel et al. 2019). After 798 potential respondents expressed interest in our study, we checked whether they fit our target group and what tasks they assumed in the team. Moreover, to ensure that our respondents had sufficient background and experience, we asked them to report on their organizational tenure and agile experience. In line with previous agile ISD research (Tripp et al. 2016, Benlian 2022), the respondents were required to have at least one year of experience in agile development and to routinely apply agile practices.<sup>4</sup> Based on the responses to our filter questions, we screened out 446 potential respondents (e.g., developers in hybrid teams using plan-driven practices together with agile development practices were excluded) and invited the remaining 352 respondents to participate in a daily study about their experiences with agile ISD practices. The invitation email included a link containing a description of the purpose and requirements of the voluntary study and informed consent. We also assured them of the confidentiality and anonymity of the responses to all surveys. The participants were incentivized with a raffle, in which each completed survey resulted in an entry toward 1 of 20 \$50 prizes.

Individuals who consented to and qualified for our study were then forwarded to a baseline survey containing measures of the between-person variables and demographics. Following the baseline survey, the

participants enrolled in the study were sent two surveys a day for two consecutive workweeks (10 days in total). A total of 274 participants signed up to participate in the study, of which 149 provided complete data for the baseline survey and for at least 7 days (of a maximum of 10 days) of the matched daily surveys. The completed surveys resulted in a final response rate of 19%, which is comparable with other ES studies (Gabriel et al. 2019). The developers had an average age of 37.9 years, and 58.5% were male. Their average organizational tenure was 7.3 years, and their median experience with agile ISD practices was “1.5 to 3 years.” We conducted *t*-tests to compare the demographic data of the developers included in the final data analysis with those who did not participate in the study. We found no significant differences in any demographics captured in this study. The mean size of the team in which the developers were working was 8.6 members. Forty-two percent indicated that their team was embedded in large-scale agile project environments, with at least 50 people or six teams guided by large-scale agile practices (e.g., SAFe or Nexus), whereas 58% responded that they worked in small- to midscale project settings (Dikert et al. 2016). Similar to previous agile ISD research (e.g., Tripp et al. 2016), in our study, the developers’ responsibilities included software development, project management, quality assurance, and testing. Eighty-two percent of respondents stated that decisions on the use of agile practices are made collectively by the project team (e.g., in retrospectives), whereas 13% indicated that individual developers predominately decide on their own, and 5% pointed to a formal team leader as the primary decision maker. This largely aligns with recommendations in the agile project management literature that teams

should be self-organizing and have collective ownership for the choice and use of agile practices (Hoda and Murugesan 2016, Schwaber and Sutherland 2020).

### Experience-Sampling Procedures and Measures

To align our theory about the daily effects of the (in-)congruence between the team-enacted versus individually needed use of agile practices with an appropriate methodology, we conducted an ES study over two weeks, consisting of two major parts. In the first part, the developers completed a baseline survey that assessed demographics and between-person variables (i.e., team feedback-seeking and between-person controls). In the second part of the study, which began one week after the completion of the first part, data were collected using an ES methodology. This methodology employed repeated assessments of the same respondents, with a focus on measuring variables that show notable short-term within-person variation that is typically missed by cross-sectional surveys (Cram et al. 2024). Additionally, the ES methodology allowed us to gain access to developers in their natural project settings and to measure constructs where they were experienced, enhancing ecological validity and minimizing recall biases (Bolger et al. 2003). As a result, this multiple-measurement design served to create a more accurate and realistic picture of developers' perceived (in-)congruence between the team-enacted versus individually needed use of agile practices than would be possible with a single retrospective survey.

We asked the developers to complete two short surveys each workday over two weeks. Following Reis and Wheeler (1991, p. 287), who stated that a “2-week record-keeping period is assumed to represent a stable and generalizable estimate of social life,” we chose 10 days for the daily surveys. We used a fixed-time schedule for daily data collection, which aligns with previous ES studies (Benlian 2022). Each afternoon during the study, the developers received a link to the first daily survey at 2:00 p.m., asking for the extent of the agile ISD practices used in teams and needed by developers on that given day. The link to the second daily survey was sent at 6:00 p.m. to ask developers to report on the extent of their well-being, work performance, and job satisfaction. The time lags between the daily afternoon and evening surveys allowed us to establish temporal separation and causal precedence between the predictor and outcome variables, alleviating common method bias and causality concerns in testing the hypotheses (Gabriel et al. 2019). According to the established processes of daily-level ES studies (Bolger et al. 2003), we referred to all items in the daily surveys as strictly daily.

For all constructs, we adopted measurement items from validated instruments and adapted them to the daily level, where necessary. Online Appendix A presents the survey items of all constructs, their psychometric

properties, and the results of a multilevel confirmatory factor analysis demonstrating convergent and discriminant validity. We controlled for developers' work engagement and work exhaustion as within-person controls as these factors have been shown to be proximate causes of developers' daily well-being (Benlian 2022). We also controlled for prior-day well-being to check for autoregressive effects and provide evidence for our hypothesized causal direction (Beal 2015). For between-person controls, following recent agile ISD studies (e.g., Kude et al. 2019), we included age, gender, organizational tenure, and agile experience in the entry survey. In addition, we accounted for important work task attributes, namely task complexity, task innovativeness, and task demand volatility (Windeler et al. 2017, Mikkelsen 2021). These between- and within-person controls allowed us to test the incremental validity of our predictions over these factors.

### Analytical Strategy

**Multilevel Analysis.** Accounting for the hierarchical data structure (i.e., days nested within individual developers), we performed multilevel analyses using the “lme4” and “lmerTest” packages in R version 4.4.3 (Nestler et al. 2019) to test the proposed effects. Following best practices in modeling ES data (Beal 2015), we used random slopes to model the level-1 within-person relationships between the team-enacted and developer-needed uses of agile practices with developer well-being. Feedback-seeking was modeled as a between-person level-2 construct. We handled missing data using the robust full information maximum likelihood estimation, which is a preferred method in multilevel models for dealing with missing data (Enders 2010). According to the recommendations of Hofmann and Gavin (1998), we group-mean centered the level-1 predictors and grand-mean centered team feedback-seeking. By removing between-person variance, group-mean centering provides a clearer estimate of the within-person relationship. This ensures that level-1 variable relationships are not influenced by individual differences, such as personality traits, response tendencies, or demographics (Gabriel et al. 2019). As mentioned earlier, we also followed the best-practice recommendations from Beal (2015) to control for lagged criteria (i.e., prior-day developer well-being), allowing us to account for a possible autocorrelation in the well-being scores and to diminish concerns of reverse causality.

### Polynomial Regression Analysis and Response Surface Modeling.

Because our predictions focused on the (in-)congruence between agile practices used versus needed, we estimated a multilevel polynomial regression model (using the “RSA” package in R 4.4.3) (Nestler et al. 2019), in which daily developer well-being was regressed on daily agile practices needed (*DAN*), daily

agile practices used ( $DAU$ ), the product term between daily agile practices needed and used ( $DAN \times DAU$ ), daily agile practices needed squared ( $DAN^2$ ), and daily agile practices used squared ( $DAU^2$ ). Polynomial regression is a more robust approach than difference scores to test the effects of (in-)congruence in that it permits a higher degree of precision (Edwards 2002, Klein et al. 2009).

Complementary to polynomial regression, we employed three-dimensional response surface modeling to analyze the relationship between  $DAU$ ,  $DAN$ , and developer well-being (Lankton et al. 2016). This approach, which plots  $DAU$  and  $DAN$  on two perpendicular horizontal axes with developer well-being on the vertical axis (see Figures 3 and 4), allows for the elucidation of complex patterns in the data<sup>5</sup> (Edwards 2001). To substantiate our hypotheses testing, we thus calculated and assessed the statistical significance of various response surface features following methodologies used in prior research (Edwards 2002; Benlian 2013, 2014). These features included the stationary point, where the surface slope is zero in all directions, and the slopes and curvatures along the lines of congruence (where  $DAU$  equals  $DAN$ ) and incongruence (where  $DAU$  and  $DAN$  differ) as well as the first and second principal axes that run perpendicular to each other and intersect at the stationary point, indicating the directions of minimized and maximized downward curvature on concave surfaces. We also analyzed the rotation of the response surface off the line of congruence to determine which type of incongruence (i.e.,  $DAU > DAN$  or  $DAU < DAN$ ) had more or less impact on the outcome variable (i.e., developer well-being). To assess the statistical significance of these response surface features, we adopted a

hierarchical bootstrapping procedure for testing linear combinations of regression coefficients, accounting for the multilevel nature of our data.<sup>6</sup> We also followed the recommendation that level-1 predictor variables' group means, their squared terms, and their interactions be included as level-2 predictors of the intercept term to control for any dependencies that remained from using group-mean centering (Preacher et al. 2016). The substantive conclusions of our results remain consistent when these additional controls are excluded.

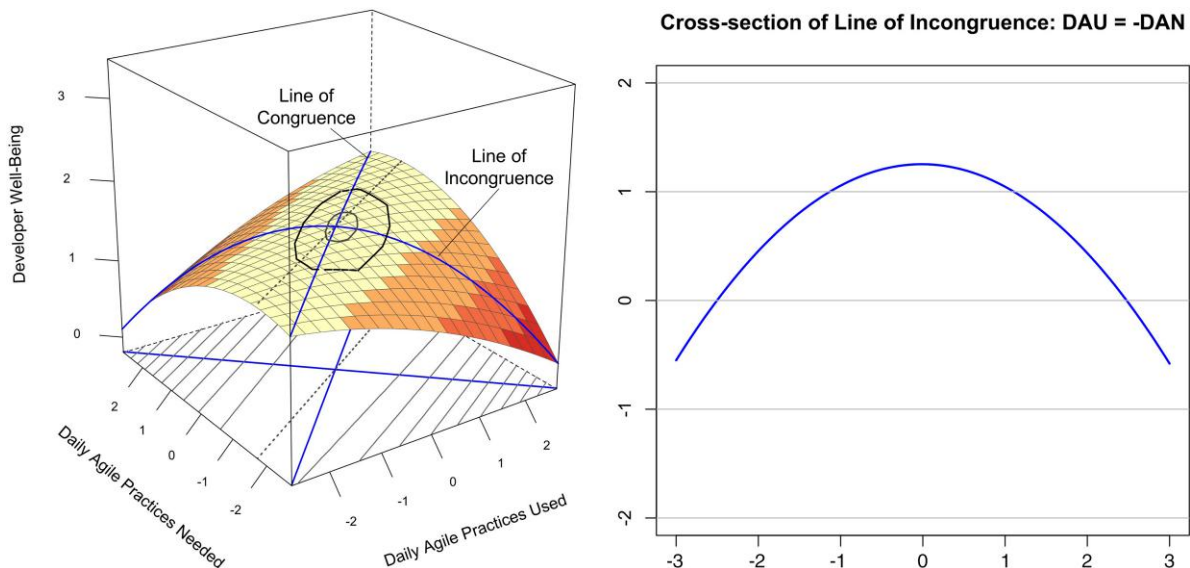
## Results

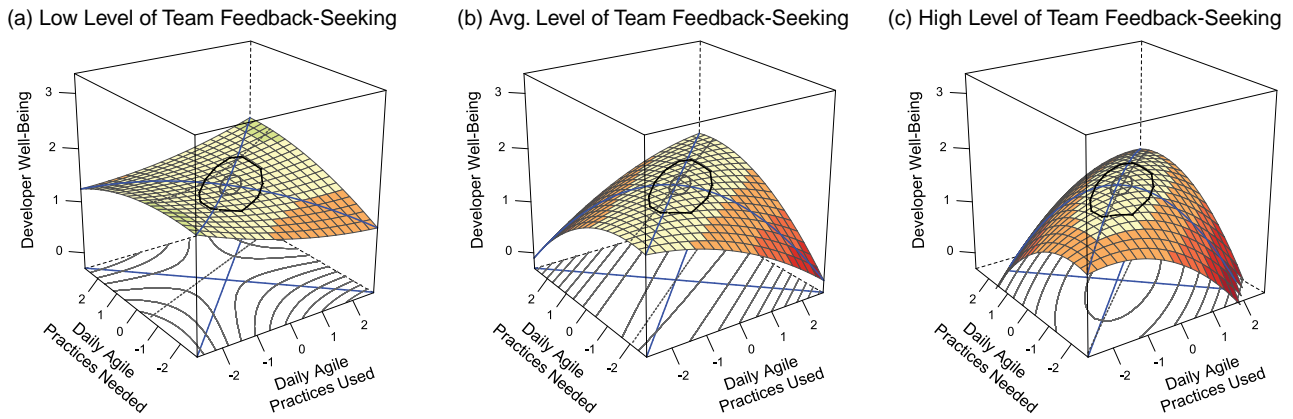
### Descriptive Statistics and Within-Person Variation in Daily Variables

Table 1 presents the descriptive statistics, reliability coefficients, and correlations for the focal variables of our study. Before testing our hypotheses, to examine whether a multilevel analysis was appropriate, we partitioned each level-1 variable into its within- and between-person variance components. As seen from the analysis of unconditional multilevel models in Table 3 in Online Appendix A, the percentage of within-person variance of level-1 variables ranged from 50.6% to 60.0%. Thus, daily fluctuations in the level-1 variable scores caused a sizable proportion of the total variance.

**Tests of Hypotheses.** The results from the multilevel polynomial regression and response surface analyses are reported in Tables 2 and 3, whereas Figures 3 and 4 show the three-dimensional response surface plots<sup>7</sup> that help illustrate our findings. Before testing our hypotheses, we first ascertained that a polynomial regression analysis was warranted. A model including higher-order

**Figure 3.** (Color online) Response Surface Plot from a Multilevel Polynomial Regression Analysis



**Figure 4.** (Color online) Response Surface Plots for Low, Average, and High Levels of Team Feedback-Seeking

Notes. (a) Low level of team feedback-seeking. (b) Average level of team feedback-seeking. (c) High level of team feedback-seeking.

(i.e., quadratic and interaction) terms (Model 2 in Table 2) for  $DAN$  and  $DAU$  accounted for significant incremental variance in developer well-being compared with a model that included only linear terms (Model 1 in Table 2). An  $F$ -test showed that the variance explained by the polynomial model was significantly higher ( $\Delta$ pseudo- $R^2 = 0.03$ ,  $p < 0.001$ ) than the variance explained by the linear model, rejecting the linear model in favor of the polynomial model (Edwards 1994). Moreover, we could identify quadratic and interaction terms in the polynomial regression equations for  $DAN$  and  $DAU$  that significantly affected developer well-being. Consequently, polynomial regression analysis adequately represented the intricate complexity of the investigated three-dimensional relationships between  $DAN$ ,  $DAU$ , and developer well-being.

**Testing of Hypothesis 1.** Hypothesis 1 predicted that developer well-being would be higher when  $DAN$  and  $DAU$  were congruent than when both components were incongruent. Figure 3 depicts the response surface plot for this polynomial regression where the congruence line ( $DAU = DAN$ ) corresponds to the line on the floor of the graph that starts in the bottom corner and runs to the top corner and the incongruence line ( $DAU = -DAN$ ) corresponds to the line on the floor of the graph that begins in the left corner and runs to the right corner. The surface in Figure 3 indicates an inverted U-shaped curve along the incongruence line, demonstrating that developer well-being is higher when the team-enacted ( $DAU$ ) and developer-needed ( $DAN$ ) uses of agile practices are congruent, whereas any deviations from the congruence line are associated with lower developer well-being. To support the congruence effect predicted by this hypothesis, the curvature along the incongruence line must be negative (i.e., an inverted U-shape) (Edwards and Cable 2009, Venkatesh and Goyal 2010). That is, developer well-being must increase as one

moves from situations of deficiency (i.e.,  $DAN > DAU$ ) toward situations of fit, be maximized in situations of fit (i.e.,  $DAU = DAN$ ), and decrease as one moves from situations of fit to situations of excess (i.e.,  $DAN < DAU$ ). In other words, developer well-being decreases when the team-enacted versus developer-needed uses of agile practices diverge from congruent levels to either deficient or excess levels. As shown in Model 2 in Table 2, the curvature along the incongruence line was negative and significant (curvature =  $-0.217$ ,  $p < 0.05$ ), indicating that the level of developer well-being increased when moving from deficient to congruent levels of  $DAN$  and  $DAU$  and decreased when moving from congruent to excess levels of  $DAN$  and  $DAU$ . This is also reflected in the decrease in developer well-being in both directions from the line of congruence as depicted in the cross-section of the line of incongruence in Figure 3. Notably, developer well-being does not decline quickly with minor incongruence; within the range of approximately  $-1$  to  $1$ , the effects on well-being are rather minimal. However, once incongruence exceeds this threshold, well-being begins to decline more steeply. To provide further support for our hypothesized congruence effect, we examined the second feature for testing a congruence effect (see also Online Appendix A), the slope of the response surface ridge (i.e., the ridge of the response surface depicted as a dashed line in Figure 3). That is, the ridge should run along the congruence line ( $DAU = DAN$ ) such that developer well-being is maximized at the point of congruence at each and every level of team-enacted ( $DAU$ ) and developer-needed ( $DAN$ ) use of agile practices (Edwards and Cable 2009). This condition is achieved when the ridge of the response surface is located along the congruence line and thus, has a slope ( $p_{11}$ ) of one and an intercept ( $p_{10}$ ) of zero (Edwards and Parry 1993). Our tests revealed that the response surface ridge had a slope that was not significantly different from 1.0

**Table 1.** Descriptives and Correlations

Variable	Descriptives		Correlation										
	Mean	SD	1	2	3	4	5	6	7	8	9	10	11
<b>Within-person variables</b>													
Daily agile practices needed (1)	3.47	0.49	<b>0.91</b>	0.751***	0.238***	0.235***	-0.049	0.211***	0.113***	0.025	-0.157***	0.236***	0.068**
Daily agile practices used (2)	3.35	0.42	0.838***	<b>0.87</b>	0.221***	0.229***	0.024	0.211***	0.075**	0.060	-0.071**	0.258***	0.082**
Developer well-being (3)	4.42	0.46	0.420***	0.404***	<b>0.89</b>	0.667***	-0.462***	0.185***	0.117***	0.144**	-0.250***	0.187***	0.049
Work engagement (4)	3.64	0.56	0.475***	0.470***	0.706***	<b>0.86</b>	-0.228***	0.130***	0.105***	0.099***	-0.165***	0.248***	-0.011
Work exhaustion (5)	2.40	0.78	-0.068	0.046	-0.469***	-0.250***	<b>0.94</b>	-0.05***	-0.118***	-0.142***	0.247***	-0.035	0.068**
<b>Between-person variables</b>													
Team Feedback-seeking (6)	4.12	0.57	0.298***	0.314***	0.268**	0.207*	-0.007	<b>0.75</b>	0.079**	0.027	-0.046	0.273***	0.217***
Agile experience (7)	2.85	1.07	0.165*	0.117	0.176*	0.180*	-0.186*	0.073	—	0.124***	-0.201***	0.187***	-0.023
Tenure (8)	7.29	6.87	0.093	0.076	0.202*	0.164	-0.199*	0.020	0.102	—	-0.084**	0.072**	-0.180***
Task complexity (9)	2.77	0.95	-0.224*	-0.111	-0.348***	-0.255***	0.343***	-0.041	-0.212*	-0.080	<b>0.82</b>	-0.036	0.295***
Task innovativeness (10)	3.85	0.72	0.327***	0.375***	0.269***	0.387***	-0.069	0.278**	0.160	0.050	-0.036	<b>0.80</b>	-0.004
Task demand volatility (11)	3.62	1.17	0.102	0.134	0.071	-0.009	0.104	0.242**	0.006	-0.163	0.303***	0.004	<b>0.73</b>

Notes.  $n = 1,510$  observations.  $N = 149$  participants. Correlations below the diagonal represent between-individual correlations ( $n = 149$ ). Correlations above the diagonal represent within-individual correlations ( $n = 1,510$ ). Cronbach's alpha coefficients are bold on the diagonal. SD, standard deviation.

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

( $p_{11} = 0.62, p > 0.10$ ) and an intercept ( $p_{10}$ ) that was not significantly different from zero ( $p_{10} = 0.12, p > 0.10$ ) (Saravanan et al. 2020). Overall, our results thus provide support for Hypothesis 1.

**Testing of Hypothesis 2.** Hypothesis 2 predicted an asymmetrical incongruence effect such that daily developer well-being would be higher when there was a low level of *DAN* and a high level of *DAU* (i.e., excess) compared with when there was a high level of *DAN* and a low level of *DAU* (i.e., deficiency). For this hypothesis to be supported, the slope along the incongruence line (i.e.,  $DAU = -DAN$ ) must be negative such that an excess level of *DAU* results in a higher level of developer well-being than an equivalently deficient level of *DAU* (Edwards 2002). In other words, we tested whether a linear decrease in the values of developer well-being emerged when moving from excess to deficient levels of *DAU*. As shown in Model 2 in Table 2, the slope along the incongruence line was negative but insignificant (slope =  $-0.002, p > 0.1$ ), demonstrating that an excess level of *DAU* did not result in higher levels of well-being than an equivalently deficient level of *DAU*. In other words, this indicates that the effects on developer well-being are the same whether  $DAN < DAU$  or  $DAN > DAU$ . The response surface plot and its cross-section in Figure 3 also illustrate that developers experienced similarly lower levels of well-being on days with both excessive and deficient levels of *DAU* as shown by the symmetrical decline in well-being on either side of the congruence line. We, therefore, failed to find support for Hypothesis 2.

**Testing of Hypothesis 3.** In Hypothesis 3a, we expected frequent (versus infrequent) team feedback seekers to experience stronger well-being losses from incongruence, whereas Hypothesis 3b proposed no additional well-being gains for frequent team feedback seekers from congruence. Following the procedures in previous IS studies using moderated polynomial regression analysis (e.g., Maruping et al. 2019, Chau et al. 2020), we first tested whether the change in pseudo- $R^2$  from the polynomial regression analysis excluding the moderator (i.e., team feedback-seeking) to the one including the moderator was significant. The moderated terms of team feedback-seeking in Model 3 in Table 2 yielded a significant increase ( $\Delta\text{pseudo-}R^2 = 0.02, p < 0.001$ ) versus the basic quadratic model (Model 2 in Table 2), ascertaining that a moderating effect was worthy of investigation.<sup>8</sup> In addition, the curvature of the line of incongruence in the moderated polynomial regression analysis (Model 3 in Table 2) was negative (i.e., concave) and significant (curvature =  $-0.220, p < 0.05$ ).

To further investigate the moderating effect of team feedback-seeking, we plotted the response surfaces (see Figure 4) and tested their slopes along the lines of

**Table 2.** Multilevel Polynomial Regression Results

Variables	Model 1: Developer well-being		Model 2: Developer well-being		Model 3: Developer well-being	
	$\gamma$	<i>s.e.</i>	$\gamma$	<i>s.e.</i>	$\gamma$	<i>s.e.</i>
Intercept	2.958***	0.220	1.671	0.911	1.515	0.909
L1 (within-person variables)						
DAN (Daily Agile Practices Needed)	<b>-0.002</b>	0.025	<b>0.006</b>	0.026	<b>0.010</b>	0.026
DAU (Daily Agile Practices Used)	<b>0.026</b>	0.025	<b>0.009</b>	0.029	<b>0.005</b>	0.029
DAN <sup>2</sup>			<b>-0.040</b>	0.034	<b>-0.041</b>	0.033
DAU <sup>2</sup>			<b>-0.075*</b>	0.038	<b>-0.071*</b>	0.039
DAN × DAU			<b>0.101</b>	0.051	<b>0.108</b>	0.051
L2 (between-person variables)						
Team feedback-seeking (TFS)	0.060*	0.024	0.044	0.024	0.055*	0.025
DAN-L2M			0.578	0.675	0.679	0.672
DAU-L2M			0.117	0.808	0.093	0.805
DAN-L2M <sup>2</sup>			0.107	0.131	0.112	0.127
DAU-L2M <sup>2</sup>			0.221	0.274	0.247	0.271
DAN-L2M × DAU-L2M			-0.411	0.368	-0.453	0.361
Controls (L1 and L2)						
Work engagement (L1)	0.385***	0.013	0.382***	0.013	0.382***	0.013
Work exhaustion (L1)	-0.191***	0.011	-0.192***	0.011	-0.192***	0.011
Developer well-being on prior day (L1)	0.081**	0.025	0.079**	0.025	0.081**	0.025
Age (L2)	0.004	0.003	0.004	0.003	0.004	0.003
Gender (L2)	0.026	0.039	0.018	0.039	0.018	0.039
Agile experience (L2)	-0.018	0.023	-0.013	0.023	-0.014	0.022
Tenure in organization (L2)	0.002	0.004	0.001	0.004	0.000	0.004
Task complexity (L2)	-0.069**	0.025	-0.065*	0.026	-0.065*	0.025
Task innovativeness (L2)	0.000	0.032	-0.011	0.033	-0.008	0.033
Task demand volatility (L2)	0.054*	0.021	0.046*	0.021	0.046*	0.021
Moderation						
DAN × TFS					<b>0.033</b>	0.024
DAU × TFS					<b>-0.031</b>	0.027
DAN <sup>2</sup> × TFS					<b>-0.059*</b>	0.034
DAU <sup>2</sup> × TFS					<b>-0.043*</b>	0.037
DAN × DAU × TFS					<b>0.058*</b>	0.047
Response surface parameters						
Line of Congruence—Slope			<b>0.015</b>	0.023	<b>0.015</b>	0.023
Line of Congruence—Curvature			<b>-0.015</b>	0.029	<b>-0.004</b>	0.029
Line of Incongruence—Slope			<b>-0.002</b>	0.050	<b>0.005</b>	0.050
Line of Incongruence—Curvature			<b>-0.217*</b>	0.098	<b>-0.220*</b>	0.098
Log likelihood	-606.0		-598.5		-595.3	
AIC	1,254.0		1,285.0		1,288.5	
BIC	1,365.7		1,519.1		1,549.2	
Pseudo-R <sup>2</sup>	0.56		0.59		0.61	

Notes. *n* = 1,510 observations. *N* = 149 participants. AIC, Akaike information criterion; BIC, Bayesian information criterion; TFS, team feedback-seeking; L, level; L2M, level-2 mean; *s.e.*, standard error. Focal coefficients are bold in the table.

\**p* < 0.05, \*\**p* < 0.01, \*\*\**p* < 0.001.

incongruence and congruence (see Table 3) with team feedback-seeking at one standard deviation (SD) below and above the mean (Maruping et al. 2019, Chau et al. 2020). The response surfaces show that the lines of incongruence become increasingly concave (i.e., the

inverted U shape becomes stronger) when moving from a low level of team feedback-seeking (-1 SD: curvature = -0.059, *p* > 0.05) (Figure 4(a)) to an average level of team feedback-seeking ( $\pm 0$  SD: curvature = -0.220, *p* < 0.05) (Figure 4(b)) to a high level of team feedback-

**Table 3.** Results of the Tests of Slopes and Curvatures Along Lines of Congruence and Incongruence

Level of team feedback-seeking	Surface along the line of congruence ( <i>X</i> = <i>Y</i> )		Surface along the line of incongruence ( <i>X</i> = - <i>Y</i> )	
	Slope	Curvature	Slope	Curvature
Low (infrequent)	0.012	0.041	-0.060	-0.059
Average	0.015	-0.004	0.005	-0.220*
High (frequent)	0.017	-0.048	0.070	-0.381**

\**p* < 0.05; \*\**p* < 0.01.

seeking (+1 SD: curvature =  $-0.381$ ,  $p < 0.01$ ) (Figure 4(c)), indicating that incongruence in *DAU* versus *DAN* becomes more consequential for developer well-being for frequent (versus infrequent) team feedback-seeking developers. In other words, when developers seek team feedback more often, higher incongruence leads to increased impairment of developer well-being and thus, has more adverse health effects. For our hypothesis to be supported, the curvature of the response surface along the line of incongruence should be more negative among frequent (versus infrequent) team feedback-seeking developers. This was confirmed when testing the statistical difference between the curvatures of the line of incongruence at one SD above (i.e.,  $-0.381$ ) and at one SD below the mean (i.e.,  $-0.059$ ) as they were significantly different from one another ( $p < 0.01$ ). This is also graphically reflected in the way that the response surface structure shifts from an almost plain-like pattern in Figure 4(a) (when the level of team feedback-seeking is low) to a dome-shaped surface in Figure 4(c) (when the level of team feedback-seeking is high), which shows how differentially vulnerable developers are to incongruence based on their team feedback-seeking behavior. Thus, Hypothesis 3a could be supported.

In Hypothesis 3b, we posited that the effects of congruence on developer well-being are not more beneficial for frequent (versus infrequent) team feedback-seeking developers. Our analysis revealed that both the slope ( $0.015$ ,  $p > 0.05$ ) and the curvature ( $-0.004$ ,  $p > 0.05$ ) of the line of congruence in the moderated polynomial regression analysis (see Table 2) were not significant. As the results in Table 3 further show, the slopes and curvatures of the line of congruence were also not significant for different levels of team feedback-seeking. The response surfaces in Figure 4 demonstrate that the line of congruence is convex for low levels of team feedback-seeking ( $-1$  SD: curvature =  $0.041$ ,  $p > 0.05$ ) (Figure 4(a)), almost linear for average levels of team feedback-seeking ( $\pm 0$  SD: curvature =  $-0.004$ ,  $p > 0.05$ ) (Figure 4(b)), and concave for high levels of team feedback-seeking (+1 SD: curvature =  $-0.048$ ,  $p > 0.05$ ) (Figure 4(c)). Statistical tests also revealed no significant differences in the curvatures (and slopes) of the lines of congruence between frequent (versus infrequent) team feedback-seeking developers (all  $p > 0.05$ ), showing that congruence between *DAU* and *DAN* does not have more beneficial effects on developer well-being for those with high (versus low) levels of team feedback-seeking behavior. Hypothesis 3b was, therefore, supported.

### Supplementary Analyses Based on Experience-Sampling Study

To further assess the relevance and robustness of our findings from the ES study, we employed a series of supplemental analyses. First, one recommended practice for exploring the results from polynomial regression and response surface analysis is to delve deeper into the (in-)

congruence effects on specific indicators of an outcome variable (Edwards et al. 1998). For example, Edwards and Harrison (1993) examined job dissatisfaction and depression as indicators of strain to better understand the stress-related effects of (mis-)fits in environment and personal factors. Following this approach, we analyzed the (in-)congruence effects on cognitive and affective well-being individually—two subconstructs that are often combined into an aggregate well-being measure. Whereas cognitive well-being emphasizes mental evaluations of life satisfaction and the assessment of future prospects, affective well-being encompasses positive and negative feelings, like stress, frustration, or enthusiasm, in response to daily experiences (Diener et al. 1999). Our analyses (see Online Appendix B) reveal that although the overall implications regarding the (in-)congruence and moderating effects (Hypotheses 1–3) remain consistent with our main findings, situations of incongruence activated cognitive-reflective processes in agile developers more strongly than emotional responses, offering a more nuanced perspective on the mechanisms behind these effects. Second, we conducted a subsample analysis to examine how our findings differ between agile developers working in small- to mid-sized and large-scale agile project environments (see Online Appendix C). Our findings show that the impact of incongruence on developer well-being intensifies in large-scale agile environments, underscoring that these effects extend beyond smaller projects to complex, contemporary agile settings. Third, we assessed the validity and wider implications of our results by testing the downstream effects of developer well-being on job satisfaction and work performance. Although work performance measures individuals' task-related work productivity, job satisfaction is a general job-related outcome variable, both of which have important implications for an individual's work life. The results of our analyses (see Online Appendix D) revealed that daily developer well-being positively affected daily job satisfaction ( $\gamma = 0.734$ ,  $p < 0.001$ ) and daily work performance ( $\gamma = 0.444$ ,  $p < 0.001$ ), demonstrating the pervasive carry-over effects of well-being on a given day. Finally, we tested a model predicting the direct effects of (in-)congruence between *DAN* and *DAU* on daily job satisfaction and daily work performance. However, our analyses did not yield significant results for the two work outcome variables (both  $p > 0.1$ ). This finding indicates that a model suggesting reverse causality, in which (in-)congruence affects job satisfaction and work performance first followed by its influence on well-being, does not align with the patterns observed in our data.

### Supplementary Qualitative Insights from Follow-up Interviews with Experience-Sampling Study Participants

We followed up with ES study participants in qualitative interviews for three main purposes: (1) to deepen

our understanding of the incongruence effects examined in our research model and corroborate the main findings from our quantitative study, (2) to provide complementary insights into the reasons for and perceived nature of incongruence, and (3) to further explore the boundary conditions (i.e., team feedback-seeking and project scale) identified in our quantitative analyses. We anticipated that developers' responses in the interviews would be especially insightful as they could provide deeper reflections on the key questions addressed in our research, building on their prior responses in the ES study and thereby, enhancing completeness of understanding (Venkatesh et al. 2013). Of the 149 agile developers who participated in the ES study, 32 agreed to a follow-up interview. With an average age of 42.8 years and 72% male, they had on average 9.3 years of software development experience; 31% of the interviewees worked in large-scale agile project settings, and 69% worked in small- to mid-scale agile project settings. Online Appendix E provides more detailed information on data collection, participant demographics, agile project settings, and the semistructured interview questions.

After we collected and compiled the interview responses, we analyzed the qualitative data based on a structural, thematic coding approach, which refers to assigning labels to segments of data based on specific topics or themes that reflect the structure of the research or interview questions (Braun and Clarke 2021, Saldaña 2021). We specifically explored questions that expanded and corroborated our research model, focusing on (1) incongruence antecedents (i.e., reasons for incongruence emergence), (2) the perceived nature (i.e., the phenomenology) of incongruence, (3) the (cognitive and affective) well-being effects of incongruence, and (4) relevant boundary conditions of these effects, with particular attention to (a) team feedback-seeking behavior and (b) project scale. At the end of each interview, we also probed for strategies and methods that agile teams and developers used to prevent or mitigate incongruence. We applied inductive coding to generate complementary insights regarding (1) and (2), whereas we used deductive coding to deepen our understanding of (3) and (4) based on the existing results from our confirmatory and exploratory analyses in the ES study (Miles et al. 2014, Saldaña 2021). The coding was conducted independently by two of the three authors, whereas any divergent responses were noted. After the initial structural coding, we held regular meetings to compare the coding and discuss any differences until reaching a consensus on the final coding scheme (see Online Appendix E for our coding scheme and exemplary quotes). In the next section, we present the main insights and inferences from our qualitative research in relation to the findings from the quantitative ES study.

**Incongruence Antecedents: Why Incongruence Emerges.** A qualitative method provides an opportunity to delve more deeply into a phenomenon of

interest. Given our interest in incongruence between the team-enacted and developer-needed uses of agile practices, we specifically probed for its emergence or “incongruence antecedents.” Using inductive coding, we categorized these antecedents into three levels: individual, team, and project. At the individual level, the most prominent factor contributing to incongruence was the divergence in personal priorities and work styles—a source of variation that seems inevitable. One interviewee reflected, “In a project like EduApp, the team prioritized rapid iterations, while I needed more time for testing and refactoring. This conflict led to incomplete feedback cycles, causing frustration on my end when features weren't tested for technical stability. Balancing speed and quality remains a challenge in agile environments” (I27).

At the team level, we found that diversity in experience and seniority often contributes to incongruence. In our interviews, we observed that less-experienced team members tend to prefer more structured agile practices, whereas seasoned developers favor flexibility. One participant explained, “Team diversity in experience can affect how agile practices are applied as less-experienced members might need more structured feedback and seasoned ones may prefer flexibility, which can lead to a disconnect in how intensively we engage in practices like continuous delivery or reviews” (I4). Some developers also highlighted inherent power imbalances within teams and insular team dynamics, which created friction around synchronizing the use of agile practices across members.

The interviews also revealed that project-level dynamics are essential factors in the emergence of incongruence. Shifting project priorities, particularly when deadlines are tight, often forces teams to compromise on agile practices use. One developer shared, “Project priorities and deadlines often shift rapidly [in our project], and so causing the team to focus more on delivery than on thorough code reviews or retrospectives, of which I rely on for continuous improvement” (I26). As project demands fluctuate, developers may find that the practices that they rely on for personal or professional growth, such as code reviews or retros, are deprioritized, leading to a sense of incongruence between their needs and the team's focus. In summary, our findings suggest that incongruence between the team-enacted and developer-needed use of agile practices can emerge from a complex set of individual-, team-, and project-related factors, with mismatches in personal working styles, team diversity, and shifting project priorities often driving this incongruence.

**Incongruence Perceptions: What Incongruence Feels Like.** Through our qualitative interviews, we also sought to deepen our understanding of how developers perceive incongruence: specifically, how it feels and

how it manifests in their daily routines. Our inductive coding revealed two primary types of incongruence perceptions. The first, described by developers as a form of temporary drift, occurs when their individual needs shift between syncs, leading to a momentary disconnect between personal workflows and the team's agile practices. As one interviewee explained, "Incongruence for me is like a pendulum swing. Between our sync-ups, I sometimes feel disconnected from the rest of the team's way of working because my own development needs might change. But with regular touchpoints ... like our sprint planning or reviews ... we realign quickly. It's more of a temporary drift rather than a lasting misalignment" (I23). This view highlights the transient nature of incongruence, suggesting that frequent team checkpoints serve as critical mechanisms to realign individual and team practices.

In contrast, other developers perceived incongruence as a deeper, more persistent miscommunication within the team. For instance, one participant (I3) likened incongruence to "using different vocabularies for the same process," where divergence between the team-enacted and developer-needed use of agile practices felt like a communication barrier. Another developer noted, "I might need more ceremonies, like daily stand-ups and reviews, to keep things transparent and fluid, but the team doesn't always stick to them. It feels like the language of agile we're supposed to be using isn't being spoken properly, so there's a lot of friction" (I19). Overall, our interviews suggest that perceived incongruence appears in diverse forms—ranging from recurring temporary misalignments that surface between syncs to deeper, systemic communication gaps, each carrying the potential to impact developer well-being.

**Incongruence Effects: How Developer Well-Being Is Affected.** For a more robust understanding of the incongruence effects, we sought to corroborate and confirm the inferences from the quantitative ES study using insights from our qualitative interviews. Grounded in the definition of subjective well-being (Diener et al. 1999, 2009) and supported by our exploratory findings (see Online Appendix B), we employed a deductive coding approach to differentiate between incongruence effects on cognitive and affective developer well-being. Specifically, we classified quotes under cognitive well-being when they reflected developers' mental evaluations of "incongruent" work situations, including references to clarity, focus, decision-making, and cognitive strain. In contrast, we assigned quotes to affective well-being when they captured developers' emotional responses to incongruence, such as frustration, stress, anxiety, or exhaustion.

Regarding cognitive well-being, we found that incongruence between the team-enacted and developer-needed uses of agile practices had negative impacts.

Developers reported confusion when practices, such as daily stand-ups or sprint backlog grooming, did not align with their personal productivity requirements. One participant described the cognitive toll that this disconnect takes: "Honestly, it gets really confusing when the team sticks rigidly to practices like daily stand-ups and backlog grooming that consume so much of my time. Like, I get lost in all these meetings, and I leave thinking, 'Wait, what exactly am I supposed to focus on?' It's mentally draining, and sometimes it feels like I'm just spinning my wheels" (I27). This sense of mental fatigue and uncertainty highlights how misaligned agile practices can impair focus and erode cognitive well-being.

In a similar vein, our findings show that incongruence has detrimental effects on affective well-being, with developers expressing feelings of resentment and emotional exhaustion. For example, one developer described her experience with rigid stand-up times, saying "In our e-commerce project, we have daily stand-ups at 9 a.m. sharp, but I'm often pulling late nights fixing bugs on the checkout system. I've mentioned in the retro that it would help to have stand-ups later, but the team prefers mornings. It's exhausting, and honestly, it's starting to make me feel unhappy ... and I'm just going through the motions at this point" (I6). This quote illustrates how incongruence can lead to frustration and disengagement when individual needs are overlooked. Together, these qualitative insights reinforce our quantitative ES findings, showing how incongruence can harm both cognitive and emotional well-being, thereby enriching our understanding of its nuanced implications for developer health.

**Incongruence Effects: Team Feedback-Seeking and Project Scale as Boundary Conditions.** Our interviews also allowed us to delve deeper into the boundary conditions of the incongruence effects that we identified in the confirmatory and exploratory analyses of our ES study: that is, developers' team feedback-seeking and project scale. Once again, we used deductive coding to distinguish incongruence effects between frequent (versus infrequent) team feedback-seeking developers and between developers in large-scale (versus small- to mid-scale) project settings.

In terms of developers' team feedback-seeking, our findings suggested that developers who frequently seek team feedback are particularly vulnerable to the adverse effects of incongruence between the team-enacted and developer-needed use of agile practices as they rely more heavily on regular feedback to track progress and ensure alignment with team expectations. For instance, one developer noted, "Team feedback is huge for me ... it's how I learn and stay on course. So, when the team's practices aren't in sync with what I need ... like when we skip retros or the reviews are spotty ... I

feel like thrown off. It's disorienting and makes me anxious" (I14).

Conversely, we observed that less frequent team feedback-seeking developers exhibit greater resilience to these incongruences. They often prioritize independence and are less reliant on continuous team input. As one developer stated, "I don't really need constant feedback from my team to stay on track. As long as I know the overall goals and have the leeway to work my way, I'm fine. If the team's way of doing things doesn't match up perfectly with my style, it's not a big deal because I'm used to working independently" (I22). This independence seems to enable them to maintain their flow and sense of accomplishment, even when team practices diverge from their needs. In sum, these insights suggest that frequent team feedback-seeking developers are more sensitive to incongruence because of their greater reliance on team feedback, whereas infrequent team feedback seekers are more self-sufficient in their work styles.

In terms of project scale, we found in our qualitative interviews that incongruence between the team-enacted and developer-needed use of agile practices manifests more prominently in large-scale project settings. One developer in a large-scale agile project highlighted how such incongruence induces significant stress, stating, "In my project, there's this constant feeling that everything you do is under a microscope. The team's practices have less leeway because there are so many moving parts when you do large-scale agile. But if those practices don't work for me, it doesn't just slow me down ... it creates extra stress because even minor delays have ripple effects on our continuous delivery pipeline and feel like a failure on a much larger scale. It affects my focus and well-being in a big way" (I15). This added layer of structural complexity in large-scale projects (e.g., additional program-level events, synchronized cadence, and increased interdependencies) seems to magnify the effects of incongruence, weighing more heavily on developers.

In contrast, developers in smaller, less complex projects reported greater ease in dealing with such misalignments. One developer stated, "It doesn't really bother me if we have a practice that feels unnecessary. It's not like in big projects where it would cause stress. Here, if something's off, we just bring it up casually, and it usually gets sorted without much fuss" (I30). This greater adaptability in smaller projects likely enables developers to address incongruence informally, making them less prone to frustration and dissatisfaction. Together, these insights underscore that incongruence effects are particularly pronounced for developers in large-scale projects, where team interdependencies and synchronization pressures tend to constrain the ability to align team agile practices with individual developer needs.

For illustrative purposes, we crafted two contrasting, stylized vignettes to synthesize the main aspects of our quantitative and qualitative study findings into a coherent and relatable picture (see Online Appendix F).

## Discussion

Much of the existing agile ISD literature has taken a narrow view, treating agile practices use in teams as a unidimensional construct without considering developers' individual needs. It has also overlooked how developers' team feedback-seeking behavior—an indicator of incongruence sensitivity—can potentially shape their experience of agile practices use. This oversight risks perpetuating unsustainable frameworks that neglect the human element in agile ISD, prompting us to examine how the (in-)congruence between the team-enacted *and* individually needed use of agile practices impact developer well-being and how these effects are affected by developers' team feedback-seeking.

Our findings demonstrate that congruence in the daily team-enacted versus individually needed use of agile practices consistently leads to higher levels of developer well-being than incongruence. Based on our P-E fit theorizing, we also found that not all instances in which developers experience high levels of team-enacted use are equivalent. In fact, depending on developers' daily needs, enacting high levels (i.e., excess conditions) can be just as detrimental to developer well-being as enacting low levels (i.e., deficiency conditions), which was inconsistent with our prediction that deficiency would be worse than excess. We surmise that an excess of team-enacted agile practices use may undermine developers' autonomy, inasmuch as a deficiency may increase their behavioral uncertainty. The added burden of navigating excessive agile practices use, particularly in work environments with high volatility, uncertainty, complexity, and ambiguity, may take a similar toll on well-being as working in settings with too little structure and guidance.

Our study also revealed that developers' team feedback-seeking behavior shapes their sensitivity to incongruence. Notably, we show that frequent (versus infrequent) team feedback-seeking developers suffer more when their needs diverge from the team-enacted use of agile practices, experiencing greater well-being losses in situations of incongruence. In contrast, they do not experience amplified well-being gains in situations of congruence. Our qualitative insights from follow-up interviews corroborated the main findings of the quantitative ES study and suggested that detrimental incongruence effects are particularly prevalent in large-scale agile project environments. Overall, our research provides novel answers to how and why developers' needs and team feedback-seeking behaviors should be central to advancing research on the

individual-level effects of agile practices use as we discuss next.

### Implications for Theory and Research

By incorporating developer needs and team feedback-seeking into the investigation of individual-level consequences of agile practices use, our study challenges currently held assumptions, resulting in valuable adaptations that expand our understanding in several important ways (see Table 4).

First, prevailing conceptualizations of agile practices use implicitly assume that developer needs fully manifest in the team-enacted use of agile practices (e.g.,

Venkatesh et al. 2020, Benlian 2022). This assumption is rooted in the idea that developers’ active participation in selecting and adapting agile practices ensures a complete correspondence between individual preferences and team-enacted practices. However, the practical reality in agile software teams (e.g., Masood et al. 2022, Nazir et al. 2024) and the findings from our follow-up interviews show that incongruence can arise because of mismatches in personal working styles, team diversity, and dynamically changing project priorities. Drawing on the P-E fit theory, we thus propose a more balanced theory that places equal emphasis on understanding the team-enacted *and* developer-needed use of agile

**Table 4.** Challenged Assumptions and Theoretical Contributions of This Study

Currently held assumption	Adapted assumption	Theoretical contribution of this study
Existing conceptualizations of agile practices use implicitly assume an inherent alignment between the team-enacted and developer-needed use of agile practices predicated on the belief that developer needs are consistently factored into team agile practices. Thus, it is sufficient to focus on the team-enacted use of agile practices alone (e.g., Venkatesh et al. 2020, Benlian 2022).	The team-enacted use of agile practices may not always perfectly reflect individual developer needs because of individual-, team-, and project-level factors, leading to incongruence. Acknowledging such incongruence has meaningful implications for developer well-being, with consequential impacts on job satisfaction and work performance.	This study offers a fresh theoretical conceptualization that places equal emphasis on understanding the team-enacted <i>and</i> individually needed use of agile practices. This more balanced notion uncovers the hidden concepts of congruence and incongruence, providing an important counterpoint to the prevailing unidimensional view of (team-enacted) agile practices use. By unpacking these conceptual dimensions, our paper offers novel perspectives for examining the (mis-)alignment between the team-enacted and individually needed use of agile practices, thereby addressing important human-centric challenges inherent in agile development processes.
The team-enacted use of agile practices and corresponding developer needs are relatively static and hardly change over short-term intervals (e.g., Tripp et al. 2016, Kude et al. 2019).	The team-enacted and developer-needed use of agile practices can exhibit considerable fluctuations over shorter time frames, with immediate impacts on developers’ health and work performance.	Our study contributes a novel microtemporal perspective, showing how day-to-day variations in the (in-)congruence between the team-enacted and individually needed use of agile practices can have <i>short-term effects</i> on developer well-being, with consequential impacts on job satisfaction and work performance.
Frequent (vs. infrequent) team feedback-seeking developers consistently achieve superior outcomes when using agile practices in teams, assuming that team feedback-seeking invariably works in synergy with the use of agile practices (e.g., Ribeiro and Alves 2021, Junker et al. 2022, Tripp and Sambamurthy 2023).	High levels of team feedback-seeking do not uniformly lead to better outcomes in conjunction with using agile practices. Frequent team feedback-seeking developers may face greater well-being losses in situations of incongruence when the team-enacted use of agile practices diverges from their individually needed use of such practices.	This research deepens our theoretical understanding of agile practices by revealing a more intricate relationship between feedback-seeking and agile practices use than previously recognized. Developers’ team feedback-seeking behavior can be viewed as an indicator of incongruence sensitivity, akin to a tuning fork, that amplifies developers’ awareness of misalignments between the team-enacted and individually needed use of agile practices. This heightened sensitivity can translate into greater susceptibility to adverse well-being effects in the face of such misalignments, thereby adding a critical dimension to our understanding of the interplay between feedback-seeking and agile practices use.

practices. Placing exclusive emphasis on the team-enacted use dimension obscures the importance of the individual needs of developers who derive meaning for their work and well-being from an intrapersonal comparative assessment between both components. This assessment provides a nuanced vocabulary for discerning the differences between congruence and incongruence effects on developer well-being. The value of such a distinction becomes most apparent in situations of incongruence (versus congruence); developers incur increased well-being costs not only in cases of deficiency but also in cases of excess—an insight that would have eluded previous unidimensional conceptualizations. Paying attention to this distinction is crucial for identifying circumstances in which developers' impaired well-being is caused by unmet needs or a debilitating overabundance of agile practices. Our study thus contributes to the agile ISD literature by providing an important counterpoint to the prevailing unidimensional view of agile practices use, offering new ways of understanding how the interplay between the team-enacted and individually needed use of agile practices relates to developer health and performance. More broadly, our study speaks to the larger conversation about how to better integrate developer needs into agile practices, making agile ISD a more sustainable and less "one-size-fits-all" experience over time.

Second, we introduce a novel microtemporal perspective to agile ISD research. Barring a few noteworthy exceptions (Benlian 2022, O'Connor et al. 2024), the dominant view in prior research has treated the use of agile practices largely as a static and unvarying phenomenon (e.g., Tripp et al. 2016, Kude et al. 2019). Our theory complements emerging dynamic perspectives by demonstrating how perceived (in-)congruence in the team-enacted versus individually needed use of agile practices can fluctuate daily within developers and how these fluctuations can be consequential for variations in developer well-being. In particular, even in cases where the use of agile practices in teams remains relatively steady, the daily shifts in developer needs can produce tangible variations in their well-being experiences. Our conceptualization of the daily interplay between the team-enacted and developer-needed use of agile practices offers a nuanced examination of immediate temporal effects and advances our understanding of agile practices' short-term dynamics, underscoring the usefulness of well-timed interventions.

Third, although past research and practical insights highlight feedback-seeking as essential to cultivating agile principles and practices (e.g., Williams and Cockburn 2003, Schwaber and Sutherland 2020), the existing literature may present an overly positive picture of their relationship by assuming that agile practices and feedback-seeking are always complementary (e.g., Ribeiro and Alves 2021, Junker et al. 2022, Tripp and Sambamurthy 2023).

Our study challenges the assumption that team feedback-seeking consistently enhances developer well-being by revealing its asymmetric effects in the context of (in-)congruence. Frequent (versus infrequent) team feedback seekers experience greater well-being losses when the team-enacted use of agile practices misaligns with their individual needs, yet they do not reap additional well-being gains from congruence. This suggests that team feedback-seeking acts as an indicator of incongruence sensitivity, heightening developers' awareness and emotional responses to misalignments. Much like a tuning fork that sensitizes perceptions to discordant notes, frequent feedback seekers become more attuned to situations of incongruence, making them more vulnerable when their sense of alignment is compromised. Our interview findings corroborated this notion, revealing that developers who rely more heavily on team feedback experience greater frustration and reduced confidence when faced with incongruence, whereas infrequent feedback seekers, prioritizing independence and self-sufficiency, remain more resilient to such misalignments. Overall, our research adds an often overlooked yet critical layer to understanding the intricate relationship between feedback-seeking and agile practices while also offering new insights into the boundary conditions that shape the effectiveness of agile practices use.

### Practical Implications

Our findings have practical implications for developers and agile teams, emphasizing the pivotal role of integrating developer needs and proactive feedback mechanisms into agile practices to shape developer well-being and subsequent work outcomes. A central implication of this study highlights the delicate balance between team-wide decisions on agile practices use and individual developer needs for such practices, acknowledging that prioritizing the former often results in overlooking the latter. This balancing act requires teams and individuals to continuously navigate how best to prevent, mitigate, or resolve perceptual incongruences, ensuring that both collective goals and personal well-being needs are addressed. By actively managing these discrepancies, teams can better support developers and reduce potential well-being costs. In our follow-up interviews with agile developers (see Table 14 in Online Appendix E), we identified three main strategies to address potential incongruence, including (1) increasing flexibility in agile practices, (2) fostering open communication and shared ownership, and (3) ensuring "agile-minded" team composition.

First, increasing flexibility in agile practices emerged as a significant strategy for bridging gaps between team-enacted practices and individual developer needs. Our qualitative findings suggest that adapting agile practices to be more flexible, both in structure and in frequency, allows developers to engage in ways that align with their

personal work styles. Examples from our interviews include introducing “No Meeting Fridays” (I16), which provided uninterrupted focus time, and adopting a 4 + 1 sprint structure, where every fifth week focuses on code refactoring and quality improvements—a “cooldown week” that accommodates the need for a balanced workload (I27). Additional options for flexibility include mid-sprint retrospectives for more real-time adjustments (I1) and offering developers more autonomy, such as opting in or out of pair programming based on task complexity (I7). Additionally, creating individualized “agile contracts” allows developers to specify their preferred level of involvement and feedback in various agile practices, enhancing their sense of agency and alignment with team practices (I4).

Second, fostering open communication and shared ownership was identified as another key factor in reducing perceptual incongruence. Our interviewees highlighted the value of blocking dedicated segments during retrospectives to regularly discuss team members’ workflow and feedback preferences, a practice that encourages ongoing dialogue and reinforces openness and respect for others’ working styles (I12). Teams can strengthen collective commitment by involving developers more in initial goal-setting sessions, where everyone participates in defining team norms on agile practices use. In one example, a team held a workshop at the beginning of a banking feature project to align everyone on sprint planning, stand-ups, and retrospectives, which resulted in greater alignment (I21). Promoting error tolerance and reinforcing shared accountability also strengthen psychological safety, ensuring that developers feel heard and included in aligning team practices with individual needs (I28).

Third, an “agile-minded” team composition can help prevent incongruence by building a team that embraces agile principles and continuous feedback. Our interviews revealed the importance of prioritizing cultural fit and openness to learning the agile way when bringing new members on board. This involves selecting individuals who are receptive to feedback and adaptive to agile principles rather than solely focusing on technical skills (I15). One participant noted that “the people that did not believe in the agile philosophy eventually worked their way out of their job as they became more of a liability than an asset” (I15). Choosing team members who understand the importance of respect and mutual support, who “can look for burnout,” and who can provide “feedback carefully but instantly” is crucial for creating a cohesive team (I20). This proactive approach to team composition helps mitigate cultural friction that can disrupt workflows, ultimately ensuring that agile practices are enacted in a manner that aligns with team and individual goals.

At a minimum, our findings urge agile teams to acknowledge the risks of overusing or underutilizing agile practices relative to individual developer needs

while also accounting for individual differences in feedback-seeking tendencies—especially in large-scale projects where maintaining alignment is particularly challenging. By adopting strategies, such as incorporating flexible practices, prioritizing open communication, and composing agile-minded teams, organizations can build a resilient environment where incongruence is effectively identified, discussed, and addressed. Such efforts can reduce negative impacts on developer well-being while fostering sustained team effectiveness over time.

### Limitations and Future Research Directions

As with any field research, our study has limitations that should be considered when interpreting the results. One limitation is our inability to examine agile practices at a finer level because of our theoretical emphasis on the overall perspective of daily agile practices use. This limits our understanding of how more discrete agile methods (e.g., XP) and practices (e.g., pair programming) contribute to (in-)congruence effects on developer well-being. Future research could delve into these finer-grained methods and explore the conceptual distinctions in social and cognitive agile practices (Hennel and Rosenkranz 2021) or software development and project management practices (Tripp et al. 2016, Mueller and Benlian 2022) to enrich our findings. Similarly, to uphold commensurability with the use dimension, our study focused on developers’ needs for agile practices at the expense of more fundamental need categories, such as competence or social relatedness. Investigating the (mis-)alignment of these basic needs with the team-enacted use of agile practices could provide deeper insights into developers’ psychological responses.

A second limitation is the potential for reverse causality, despite the temporal ordering of our theoretical constructs according to the P-E fit theory. For example, developers experiencing lower levels of well-being may need and opt to use fewer agile practices, and those with frequent team feedback-seeking behaviors may actively work to close the incongruence gap between used and needed agile practices over time. We addressed potential reverse causality by employing lagged measurements and testing a model where developer well-being was assumed to influence the team-enacted and developer-needed use of agile practices. This analysis revealed no significant effects, providing stronger evidence for our theorized causal direction (Beal 2015). However, fully excluding rival explanations would require experimental or interventional approaches. We encourage future research to adopt such approaches to explore reverse causality and other temporal dynamics involving agile practices, feedback-seeking, and developer outcomes. For example, future research may investigate how proactive feedback

practices may change perceptual incongruence over longer periods of time.

A third limitation of our study is the reliance on a single data source for all variables. Although developers are ideally positioned to self-report their needs and well-being, given the inherently personal and psychological nature of these constructs (Gabriel et al. 2019), this approach may lead to concerns about common method variance (CMV). To reduce such concerns, we took several steps. First, we group-mean centered our predictor variables, effectively controlling for common source effects, such as recall bias, social desirability, and trait affectivity (Hofmann and Gavin 1998). Second, by employing time lags between the daily afternoon and evening surveys, we made sure that the measures of our focal variables were separated across time (Podsakoff et al. 2003). Third, our study's identification of significant crosslevel interactions serves to reduce CMV concerns (Lai et al. 2013).

These limitations notwithstanding, our research opens several directions for future research. First, we deliberately focused on the key (in-)congruence relationships between the team-enacted and individually needed use of agile practices rather than more broadly exploring the complex web of relationships within the entire (in-)congruence space. Future research can leverage polynomial regression and response surface modeling to delve deeper into more intricate research questions. For example, how does escalating incongruence for both excess and deficiency affect developer outcomes? Or, does it matter for developer well-being whether the congruence between the team-enacted and individually needed use of agile practices occurs at high or low levels of agile practices use (see quadrants 2 and 3 in Figure 1)? We also recognize the likelihood of different underlying mechanisms driving congruence and incongruence effects. Congruence may lead to reduced cognitive and emotional responses (e.g., decreased arousal), whereas incongruence (i.e., excess and deficiency) may trigger heightened cognitive and emotional reactions, including mental fatigue and frustration as we witnessed in our follow-up interviews. Hence, future studies could investigate distinct mediation pathways for congruence and incongruence. Second, although our study centered on (in-)congruence regarding agile practices, there is an opportunity to extend this perspective to the nature of work tasks. Although agile methods guide how developers approach their work, they do not explicitly account for specific task characteristics, such as complexity, volatility, or significance. Future research could explore how (in-)congruence between task characteristics and developer needs affects their work task performance, thereby deepening our understanding of P-E fit dynamics in agile work settings.

Finally, although we have taken initial steps toward understanding how large-scale agile project settings can affect developer well-being and work performance,

there is significant room for further research. As organizations increasingly implement large-scale agile and high-speed postagile methods (e.g., Dennehy and Conway 2018, Carroll et al. 2023), there is a growing need for multilevel and longitudinal studies. Future research could contribute to the nascent stream of research on the normalization of these methods by exploring how teams and developers make sense of integration challenges and unique temporal mechanisms, such as rhythmic synchronization, cadence, and flow (Blagoev et al. 2024, O'Connor et al. 2024). Such investigations could yield valuable insights into novel temporal coordination patterns and routine dynamics of scaled (post-)agile projects in modern organizations.

## Conclusion

Prevailing views in agile ISD research have often idealized the use of agile practices in teams, portraying it as inherently synchronized with individual developer needs and largely impervious to short-term fluctuations. However, our research reveals a more nuanced reality; (in-)congruence between the team-enacted and individually needed use of agile practices significantly influences developer well-being and performance, and these effects can vary daily. Additionally, we show that developers' team feedback-seeking behavior, often assumed to go hand in hand with agile practices use, can inadvertently harm their well-being in situations of incongruence. By emphasizing the critical role of developers' individual needs, our study advances a more nuanced, human-centered understanding of agile practices use, advocating for agile frameworks that equally sustain developer well-being and team performance in practice.

## Endnotes

<sup>1</sup> In line with the person-environment (P-E) fit theory, incongruence (or misfit) refers to any level of divergence, ranging from slight differences to major mismatches (Edwards 1994, 2002).

<sup>2</sup> In our follow-up interviews with developers of this study, we also identified various individual-, team-, and project-level factors contributing to the emergence of incongruence (see the Results section and Online Appendix E).

<sup>3</sup> We follow previous conceptualizations in IS research (Venkatesh et al. 2020, Benlian 2022) that focused their investigations on "the extent of" agile practice use, capturing the degree to which different types of agile practices (e.g., software development or project management practices) are enacted in their teams. Consistent with prior P-E fit theorizing, the need dimension is commensurate to the use dimension in its reference to agile practices.

<sup>4</sup> We, for example, checked via our baseline survey how many agile practices (e.g., daily stand-ups, pair programming, small releases, and test-driven development) developers continuously used in their teamwork (Baham and Hirschheim 2022). As we recruited participants via our university's industry partners, we could also verify on a test basis that the developers used these practices routinely in an agile way.

<sup>5</sup> For example, polynomial regression and response surface methods allow for the examination of both congruence and incongruence patterns, whereby for incongruence, any divergences—from slight differences to major mismatches—can be captured in assessing their impact on outcomes (Edwards 2002).

<sup>6</sup> In Online Appendix A, we provide supplementary information on testing (in-)congruence effects in polynomial regression and response surface analysis and on how the hierarchical bootstrapping procedure was applied.

<sup>7</sup> More specifically, the response surface in Figure 3 is based on the regression results of Model 2 in Table 2, whereas the regression results of Model 3 in Table 2 were used to plot the response surfaces in Figure 4.

<sup>8</sup> The change in the pseudo- $R^2$  value of the interaction model indicated that 2% of the total variance in developer well-being was attributed to the inclusion of the moderating effect of feedback seeking. Small incremental variance explained is common for (polynomial) crosslevel interaction models (e.g., Liao et al. 2019).

## References

- Ågerfalk PJ, Fitzgerald B, Slaughter SA (2009) Flexible and distributed information systems development: State of the art and research challenges. *Inform. Systems Res.* 20(3):317–328.
- Ashford SJ, Black JS (1996) Proactivity during organizational entry: The role of desire for control. *J. Appl. Psych.* 81(2):199–214.
- Ashford SJ, Cummings LL (1983) Feedback as an individual resource: Personal strategies of creating information. *Organ. Behav. Human Performance* 32(3):370–398.
- Baham C, Hirschheim R (2022) Issues, challenges, and a proposed theoretical core of agile software development research. *Inform. Systems J.* 32(1):103–129.
- Bandura A (1989) Human agency in social cognitive theory. *Amer. Psychologist* 44(9):1175–1184.
- Baumeister RF, Bratslavsky E, Finkenauer C, Vohs KD (2001) Bad is stronger than good. *Rev. General Psych.* 5(4):323–370.
- Beal DJ (2015) ESM 2.0: State of the art and future potential of experience sampling methods in organizational research. *Annual Rev. Organ. Psych. Organ. Behav.* 2(1):383–407.
- Beck K, Beedle M, Van Bennekum A, Cockburn A, Cunningham W, Fowler M, Grenning J, et al. (2001) Manifesto for agile software development. Accessed March 15, 2025, <https://agilemanifesto.org/>.
- Benlian A (2013) Effect mechanisms of perceptual congruence between information systems professionals and users on satisfaction with service. *J. Management Inform. Systems* 29(4):63–96.
- Benlian A (2014) Are we aligned ... enough? The effects of perceptual congruence between service teams and their leaders on team performance. *J. Service Res.* 17(2):212–228.
- Benlian A (2022) Sprint zeal or sprint fatigue? The benefits and burdens of agile ISD practices use for developer well-being. *Inform. Systems Res.* 33(2):557–578.
- Blagoev B, Hernes T, Kunisch S, Schultz M (2024) Time as a research lens: A conceptual review and research agenda. *J. Management* 50(6):2152–2196.
- Bolger N, Davis A, Rafaeli E (2003) Diary methods: Capturing life as it is lived. *Annual Rev. Psych.* 54(1):579–616.
- Braun V, Clarke V (2021) *Thematic Analysis: A Practical Guide* (SAGE, London).
- Cable DM, Edwards JR (2004) Complementary and supplementary fit: A theoretical and empirical integration. *J. Appl. Psych.* 89(5):822–834.
- Callister RR, Kramer MW, Turban DB (1999) Feedback seeking following career transitions. *Acad. Management J.* 42(4):429–438.
- Carroll N, Conboy K, Wang X (2023) From transformation to normalisation: An exploratory study of a large-scale agile transformation. *J. Inform. Tech.* 38(3):267–303.
- Chau DCK, Ngai EWT, Gerow JE, Thatcher JB (2020) The effects of business-IT strategic alignment and its governance on firm performance: A moderated polynomial regression analysis. *MIS Quart.* 44(4):1679–1703.
- Cockburn A, Highsmith J (2001) Agile software development, the people factor. *Computer* 34(11):131–133.
- Conboy K, Coyle S, Wang X, Pikkarainen M (2011) People over process: Key challenges in agile development. *IEEE Software* 28(4):48–57.
- Cram WA, D’arcy J, Benlian A (2024) Time will tell: A case for an idiographic approach for behavioral cybersecurity research. *MIS Quart.* 48(1):95–136.
- Dennehy D, Conboy K (2018) Identifying challenges and a research agenda for flow in software project management. *Project Management J.* 49(6):103–118.
- Diener E, Oishi S, Lucas RE (2009) Subjective well-being: The science of happiness and life satisfaction. Lopez S, Snyder C, eds. *Oxford Handbook of Positive Psychology* (Oxford University Press, Oxford, UK), 187–192.
- Diener E, Suh EM, Lucas RE, Smith HL (1999) Subjective well-being: Three decades of progress. *Psych. Bull.* 125(2):276–302.
- digital.ai (2023) 17th State of agile report. Accessed March 15, 2025, <https://digital.ai/resource-center/analyst-reports/state-of-agile-report/>.
- Dikert K, Paasivaara M, Lassenius C (2016) Challenges and success factors for large-scale agile transformations: A systematic literature review. *J. Systems Software* 119:87–108.
- Edwards JR (1994) The study of congruence in organizational behavior research: Critique and a proposed alternative. *Organ. Behav. Human Decision Processes* 58(1):51–100.
- Edwards JR (2001) Ten difference score myths. *Organ. Res. Methods* 4(3):265–287.
- Edwards JR (2002) Alternatives to difference scores: Polynomial regression analysis and response surface methodology. Drasgow F, Schmitt N, eds. *Measuring and Analyzing Behavior in Organizations: Advances in Measurement and Data Analysis* (Jossey-Bass/Wiley, San Francisco), 350–400.
- Edwards JR, Cable DM (2009) The value of value congruence. *J. Appl. Psych.* 94(3):654–677.
- Edwards JR, Harrison RV (1993) Job demands and worker health: Three-dimensional reexamination of the relationship between person–environment fit and strain. *J. Appl. Psych.* 78(4):628–648.
- Edwards JR, Parry ME (1993) On the use of polynomial regression equations as an alternative to difference scores in organizational research. *Acad. Management J.* 36(6):1577–1613.
- Edwards JR, Shipp AJ (2007) The relationship between person–environment fit and outcomes: An integrative theoretical framework. Ostroff C, Judge TA, ed. *Perspectives on Organizational Fit* (Jossey-Bass, San Francisco), 209–258.
- Edwards JR, Caplan RD, Van Harrison R (1998) Person–environment fit theory: Conceptual foundations, empirical evidence, and directions for future research. Cooper CL, ed. *Theories of Organizational Stress* (Oxford University Press, Oxford, UK), 28–67.
- Edwards JR, Cable DM, Williamson IO, Lambert LS, Shipp AJ (2006) The phenomenology of fit: Linking the person and environment to the subjective experience of person–environment fit. *J. Appl. Psych.* 91(4):802–827.
- Enders CK (2010) *Applied Missing Data Analysis* (Guilford Press, New York).
- Fischer M, Heinz M, Schlereth C, Rosenkranz C (2023) “You can’t always get what you want”: Examining employees’ preferences and job satisfaction in agile transformations. ECIS 2023 Research Paper No. 393, ECIS, Kristiansand, Norway.
- Fitzgerald B, Hartnett G, Conboy K (2006) Customising agile methods to software practices at Intel Shannon. *Eur. J. Inform. Systems* 15(2):200–213.

- Fry LW, Smith DA (1987) Congruence, contingency, and theory building. *Acad. Management Rev.* 12(1):117–132.
- Gabriel AS, Podsakoff NP, Beal DJ, Scott BA, Sonntag S, Trougakos JP, Butts MM (2019) Experience sampling methods: A discussion of critical trends and considerations for scholarly advancement. *Organ. Res. Methods* 22(4):969–1006.
- Hennel P, Rosenkranz C (2021) Investigating the “socio” in socio-technical development: The case for psychological safety in agile information systems development. *Project Management J.* 52(1):11–30.
- Hoda R, Murugesan LK (2016) Multi-level agile project management challenges: A self-organizing team perspective. *J. Systems Software* 117:245–257.
- Hofmann DA, Gavin MB (1998) Centering decisions in hierarchical linear models: Implications for research in organizations. *J. Management* 24(5):623–641.
- Jansen KJ, Kristof-Brown AL (2005) Marching to the beat of a different drummer: Examining the impact of pacing congruence. *Organ. Behav. Human Decision Processes* 97(2):93–105.
- Joyce W, Slocum JW, Von Glinow MA (1982) Person-situation interaction: Competing models of fit. *J. Organ. Behav.* 3(4):265–280.
- Junker TL, Bakker AB, Derks D (2024) Toward a theory of team resource mobilization: A systematic review and model of sustained agile team effectiveness. *Human Resource Management Rev.* 35(1):101043.
- Junker TL, Bakker AB, Gorgievski MJ, Derks D (2022) Agile work practices and employee proactivity: A multilevel study. *Human Relations* 75(12):2189–2217.
- Klein G, Jiang JJ, Cheney P (2009) Resolving difference score issues in information systems research. *MIS Quart.* 33(4):811–826.
- Kristof AL (1996) Person-organization fit: An integrative review of its conceptualizations, measurement, and implications. *Personnel Psych.* 49(1):1–49.
- Kristof-Brown AL, Zimmerman RD, Johnson EC (2005) Consequences of individuals’ fit at work: A meta-analysis of person–job, person–organization, person–group, and person–supervisor fit. *Personnel Psych.* 58(2):281–342.
- Kude T, Foerderer J, Mithas S, Heinzl A (2023) How deadline orientation and architectural modularity influence software quality and job satisfaction. *J. Oper. Management* 69(6):941–964.
- Kude T, Mithas S, Schmidt CT, Heinzl A (2019) How pair programming influences team performance: The role of backup behavior, shared mental models, and task novelty. *Inform. Systems Res.* 30(4):1145–1163.
- Lai X, Li F, Leung K (2013) A Monte Carlo study of the effects of common method variance on significance testing and parameter bias in hierarchical linear modeling. *Organ. Res. Methods* 16(2):243–269.
- Lankton NK, Mcknight DH, Wright RT, Thatcher JB (2016) Research note—Using expectation disconfirmation theory and polynomial modeling to understand trust in technology. *Inform. Systems Res.* 27(1):197–213.
- Liao Z, Liu W, Li X, Song Z (2019) Give and take: An episodic perspective on leader-member exchange. *J. Appl. Psych.* 104(1):34–51.
- Luong TT, Sivarajah U, Weerakkody V (2021) Do agile managed information systems projects fail due to a lack of emotional intelligence? *Inform. Systems Frontiers* 23(2):415–433.
- Maruping LM, Daniel SL, Cataldo M (2019) Developer centrality and the impact of value congruence and incongruence on commitment and code contribution activity in open source software communities. *MIS Quart.* 43(3):951–976.
- Masood Z, Hoda R, Blincoc K (2022) Real world scrum a grounded theory of variations in practice. *IEEE Trans. Software Engrg.* 48(5):1579–1591.
- Meckenstock J-N (2024) Shedding light on the dark side—A systematic literature review of the issues in agile software development methodology use. *J. Systems Software* 211:111966.
- Mikkelsen MF (2021) Perceived project complexity: A survey among practitioners of project management. *Internat. J. Managing Projects Bus.* 14(3):680–698.
- Miles MB, Huberman AM, Saldaña J (2014) *Qualitative Data Analysis: A Methods Sourcebook* (SAGE Publications, Thousand Oaks, CA).
- Milne A (2022) 10 reasons why agile transformations fail and how yours can succeed. *net solutions* (August 8), <https://www.netsolutions.com/insights/how-to-prevent-agile-transformation-failure/>.
- Mueller L, Benlian A (2022) Too drained from being agile? The self-regulatory effects of agile ISD practices use and their consequences for turnover intention. *J. Assoc. Inform. Systems* 23(6):1420–1455.
- Mueller L, Albrecht G, Toutaoui J, Benlian A, Cram WA (2025) Navigating role identity tensions—IT project managers’ identity work in agile information systems development. *Eur. J. Inform. Systems* 34(2):383–406.
- Nazir S, Collignon SE, Surendra NC (2024) Understanding collective ownership in agile development: Turbo charging the process. *Inform. Management* 61(6):104004.
- Nestler S, Humberg S, Schönbrodt FD (2019) Response surface analysis with multilevel data: Illustration for the case of congruence hypotheses. *Psych. Methods* 24(3):291–308.
- O’Connor M, Conboy K, Dennehy D, Carroll N (2024) Temporal complexity in information systems development flow: Challenges and recommendations. *Comm. Assoc. Inform. Systems* 54(1):699–735.
- Podsakoff PM, Mackenzie SB, Lee J, Podsakoff NP (2003) Common method biases in behavioral research: A critical review of the literature and recommended remedies. *J. Appl. Psych.* 88(5):879–903.
- Preacher KJ, Zhang Z, Zyphur MJ (2016) Multilevel structural equation models for assessing moderation within and across levels of analysis. *Psych. Methods* 21(2):189–205.
- Ramesh B, Mohan K, Cao L (2012) Ambidexterity in agile distributed development: An empirical investigation. *Inform. Systems Res.* 23(2):323–339.
- Reis HT, Wheeler L (1991) Studying social interaction with the Rochester Interaction Record. Zanna MP, ed. *Advances in Experimental Social Psychology* (Academic Press, San Diego), 270–318.
- Ribeiro ABC, Alves CF (2021) A survey research on feedback practices in agile software development teams. Santos R, Oliveira SRB, Santos IdS, Marczak S, Viana D, Canedo ED, Albuquerque AB, et al., eds. *Proc. XX Brazilian Sympos. Software Quality* (Association for Computing Machinery, New York), 1–10.
- Rietze S, Zacher H (2022) Relationships between agile work practices and occupational well-being: The role of job demands and resources. *Internat. J. Environ. Res. Public Health* 19(3):1258.
- Saldaña J (2021) *The Coding Manual for Qualitative Researchers* (SAGE Publications, Thousand Oaks, CA).
- Saravanan V, Berman GJ, Sober SJ (2020) Application of the hierarchical bootstrap to multi-level data in neuroscience. *Neuron Behav. Data Anal. Theory* 3(5).
- Sarker S, Ahuja M, Sarker S (2018) Work-life conflict of globally distributed software development personnel: An empirical investigation using border theory. *Inform. Systems Res.* 29(1):103–126.
- Schwaber K, Sutherland J (2020) The Scrum guide. The definitive guide to Scrum: The rules of the game. Accessed November 30, 2024, <https://www.scrum.org/resources/scrum-guide>.
- Singh K, Strobel J (2022) Exploring lived experiences of agile developers with daily stand-up meetings: A phenomenological study. *Behaviour Inform. Tech.* 42(4):403–423.
- Toutaoui J, Benlian A, Hess T (2022) Managing paradoxes in bi-modal information technology functions: A multi-case study. *Inform. Systems J.* 32(6):1177–1202.
- Tripp J, Sambamurthy V (2023) How agile feedback practice use impacts software quality. *J. Comput. Inform. Systems*, ePub

- ahead of print December 20, <https://doi.org/10.1080/08874417.2023.2286557>.
- Tripp JF, Riemenschneider C, Thatcher JB (2016) Job satisfaction in agile development teams: Agile development as work redesign. *J. Assoc. Inform. Systems* 17(4):267–307.
- van Oorschot KE, Sengupta K, Van Wassenhove LN (2018) Under pressure: The effects of iteration lengths on agile software development performance. *Project Management J.* 49(6):78–102.
- Venkatesh V, Goyal S (2010) Expectation disconfirmation and technology adoption: Polynomial modeling and response surface analysis. *MIS Quart.* 34(2):281–303.
- Venkatesh V, Brown SA, Bala H (2013) Bridging the qualitative-quantitative divide: Guidelines for conducting mixed methods research in information systems. *MIS Quart.* 37(1):21–54.
- Venkatesh V, Thong JYL, Chan FKY, Hoehle H, Spohrer K (2020) How agile software development methods reduce work exhaustion: Insights on role perceptions and organizational skills. *Inform. Systems J.* 30(4):733–761.
- Vidgen R, Wang X (2009) Coevolving systems and the organization of agile software development. *Inform. Systems Res.* 20(3):355–376.
- Vogel RM, Rodell JB, Sabey TB (2020) Meaningfulness misfit: Consequences of daily meaningful work needs-supplies incongruence for daily engagement. *J. Appl. Psych.* 105(7):760–770.
- Wang X, Conboy K, Pikkarainen M (2012) Assimilation of agile practices in use. *Inform. Systems J.* 22(6):435–455.
- Wiesche M (2021) Interruptions in agile software development teams. *Project Management J.* 52(2):210–222.
- Williams L, Cockburn A (2003) Agile software development: It's about feedback and change. *Computer* 36(6):39–43.
- Windeler JB, Maruping L, Venkatesh V (2017) Technical systems development risk factors: The role of empowering leadership in lowering developers' stress. *Inform. Systems Res.* 28(4):775–796.
- Wright TA, Cropanzano R (2000) Psychological well-being and job satisfaction as predictors of job performance. *J. Occupational Health Psych.* 5(1):84–94.