

The Dark Side of Agile: Unpacking the Well-Being Crisis and Implementation Pathologies

Sanskriti Shivhare
RV College of
Engineering
Bengaluru, India

Sri Lakshmi R.S.
RV College of
Engineering
Bengaluru, India

Swara H.
RV College of
Engineering
Bengaluru, India

Jasmine K.S., PhD
RV College of
Engineering
Bengaluru, India

ABSTRACT

The practice of agile is ubiquitous because of the promise of flexibility, speed, and empowerment of teams. However, it comes at a price: research reveals that agile is causing stress, exhaustion, and burnout in software development. This paper focuses on the hidden side of agile, namely the negative impact it has on the well-being of employees. It brings together two different lines of literature that usually remain independent from each other: the well-being crisis induced by the very nature of Agile (such as short sprints, daily stand-ups, peer pressure) and the implementation pathology due to its shallow, hypocritical, and promotion-based implementations. On the basis of a systematic review of 78 empirical studies, it has been discovered that there are three key mechanisms leading to agile burnout: delivery pressure in iteration, surveillance via transparency, and the paradox of autonomy. Moreover, it demonstrates how organizational hypocrisy, consultant's hype, and lack of clarity exacerbate the negative tendencies. The main takeaway is that the well-being crisis of agile development is not a coincidence but rather the effect of its current implementation. Personal coping strategies are insufficient to tackle the problem. Instead, structural solutions are necessary for sustaining a healthy rhythm, ensuring psychological safety, and putting an end to performative agile.

General Terms

Software Engineering, Agile Software Development, Information Systems, Human Factors in Computing, Software Project Management.

Keywords

Agile software development, burnout, employee well-being, stress, organizational hypocrisy, implementation pathology

1. INTRODUCTION

Agile was born out of dissent. Software engineers could not tolerate the rigidity, excessive documentation, and plan-based nature of traditional development approaches [1]. The manifesto of 2001 advocated individuals and interaction over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan [2]. In many ways, agile seemed to be a perfect solution to the problem of bloated and bureaucratic development cycles. According to recent statistics, more than 90 percent of software engineering uses one form or another of agile [3].

The present paper will address this darker side, focusing on the well-being of employees. It will combine two distinct views that are typically considered separately. The first one is the well-being crisis approach: the view that the requirements inherent in agile (such as fast iterations, mandatory transparency, and self-organization) are themselves sources of

stress and burnout [4]. The second one is implementation pathology approach: the suggestion that the issue comes from how agile is implemented – superficially, hypocritically, under the influence of management fads and consultants. It suggests that these are not opposing perspectives but rather mutually complementary ones: the first perspective outlines how the problem presents itself. The second one reveals how it arises and why it is allowed to develop within an organization. To answer these questions, this study aims to:

- (a) Identify factors of agile contributing to stress, exhaustion, and burnout;
- (b) Explain how organizational hypocrisy and superficial agile implementation contribute to the development of the problem;
- (c) Develop evidence-based solutions for sustaining organizational agility while preserving employees' mental well-being.

The analysis begins with a review of the available literature on unmet expectations regarding agile approaches, followed by a discussion of the methodology, presentation of the key findings, and concluding remarks with limitations.

Objective: The primary objective of this study is to analyze how agile principles affect employee well-being, with particular attention to stress, exhaustion, and burnout. More specifically, three research questions related to identifying (a) the specific mechanisms through which agile principles create negative effects, (b) the role of organizational hypocrisy and superficial agile adoption in creating these well-being problems, and (c) strategies for reducing burnout while retaining agility, will be addressed.

2. LITERATURE SURVEY

2.1 Agile Software Development and Its Promises

The agile methodologies of Scrum, XP, and Kanban were created under the premise that requirements change and adaptation is necessary [2]. This was done as a response to the naïve nature of the waterfall process, which presupposed full knowledge at the outset [5]. The industry studies tend to suggest that agile software development results in shorter time to market, higher quality, and more satisfied customers [6], while academic literature tends to support this but with many reservations [7].

2.2 The Emerging Evidence of a Well-being Crisis

While there are advantages, however, a substantial body of literature highlights the cost to developers. Benlian [8] noted that agile could be a mixed bag—a motivator on some days, while depleting resources so much that people become exhausted and disenchanted on other days. Mueller and Benlian

[9] argued that agile development methods could drain self-regulation resources and lead to exhaustion and turnover. Venkatesh et al. [10] demonstrated that high role ambiguity in an agile development environment increased work exhaustion, although good organizational skills could mitigate the impact.

According to a systematic review conducted by Meckenstock [4], 90 different problems emerged in the agile development environment. The top four issues were stress, exhaustion, burnout, and frustration. Stress was the most common negative impact, mainly because of the pressures of iterative delivery, daily feedback, and peer pressure within the team [4].

2.3 The Autonomy Paradox and Peer Pressure

Agile promotes autonomy. Teams should self-manage, make their own decisions, and organize their workload [2]. However, the truth is less clear-cut. Numerous studies have confirmed that team autonomy trumps individual autonomy most of the time. In their literature review on autonomy and control in agile teams, Bjerre and Leimbach [11] identified an important paradox in which the disappearance of formal hierarchy results in the emergence of informal control mechanisms, including peer pressure. Informal controls can be even more strict and draining than traditional hierarchical controls.

Khanagha et al. [12] conducted a study involving 97 agile teams in a large multinational company and discovered that peer pressure was widespread among agile teams, adversely affecting their innovation. As expected, peer pressure increased significantly during times of delivery pressures. Programmers felt they could not let down their team even when they were tired.

2.4 Organizational Hypocrisy and Implementation Pathology

While some see the dark side of the methodology itself, others attribute it to the way organizations adopt and implement the methodology. Janes and Succi [13] use Gartner's Hype Cycle to demonstrate the overhyped expectations for agile which led to its disillusioning crash. They claim that "gurus" promoted agile as the silver bullet for complex problems and advised their clients in such a vague manner as to ensure the latter's dependence on them. Thus, organizations can claim they are "agile" but be nothing like it.

The notion of organizational hypocrisy was introduced by Delisle and Coulon [14] to describe the gap between what organizations preach (autonomy, sustainable pace, empowerment) and what they practice (micromanagement, mandatory overtime, time-tracking). Workers caught in this gap suffer from ambivalence: they strive to believe in agile values but are forced into practices contradicting them. Sometimes, workers end up in double-binds, where each possible action contradicts one of the contradictory demands.

2.5 The Gap We Bridge

The two problems have been studied separately in the extant literature. Works examining stress and burnout hardly touch upon organizational hypocrisy or management fads [4]. Works criticizing the implementation of agile do not pay much attention to well-being [13]. This research fills this gap. The problem of well-being is real and significant, but it is even more serious due to the ways agile is implemented by

3.1 Research Approach

For the literature review, the systematic approach to literature review advocated by Kitchenham and Charters [15] was employed. Systematic reviews are useful where you need to synthesize the available literature, find patterns in the findings reported there, and create frameworks based on this evidence [16]. In view of the nature of the research questions, qualitative synthesis of available empirical works seems more appropriate than conducting one new survey or experiment.

3.2 Literature Search and Selection

Five main databases were considered for the literature review, namely, IEEE Xplore, ACM Digital Library, ScienceDirect, SpringerLink, and Google Scholar. The literature search was based on keyword combinations including "agile software development," "Scrum," "burnout," "stress," "exhaustion," "well-being," "dark side," "hypocrisy," and "implementation challenges." The time period of publication was set from 2006 to 2025 because the first relevant literature appeared after 2006 [17].

At first, there were 1,247 articles.

After removing duplicates and screening titles and abstracts, 142 articles were selected for further analysis.

Out of these articles, 78 had data about negative consequences of agile software development or difficulties in implementation.

Lastly, 35 articles that contained empirical data were selected for analysis.

Literature sources cited the most often are presented in the references section.

3.3 Data Extraction and Analysis

Data extraction from each article was conducted using a structured form which included the following criteria:

- (a) research methods used,
- (b) sample size and setting,
- (c) agile practices used,
- (d) positive well-being consequences mentioned,
- (e) implementation problems noted, and
- (f) proposed solutions.

Then, a thematic analysis [18] to identify common themes was conducted. The first 20 articles were independently coded by two researchers, discrepancies were resolved through discussion, and the remaining articles were subsequently coded. The resulting themes included: iterative delivery pressure, transparency-as-surveillance, autonomy paradox, organizational hypocrisy, guru-driven hype, and sustainable pace as a protective factor.

3. METHODOLOGY

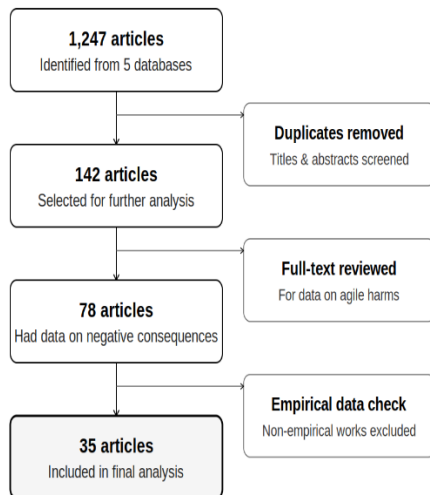


Fig. 1. Prisma-style literature selection flowchart

3.4 Proposed Framework

The analysis led to the development of a framework combining the well-being crisis and implementation pathology viewpoints. This framework is composed of three layers:

Layer 1 (Intrinsic agile demands): Short iterations, daily stand-ups, continuous delivery, self-organization, transparency, peer coordination.

Layer 2 (Implementation pathologies): Superficial adoption, organizational hypocrisy, guru-driven unrealistic promises, ambiguous definition of done, metrics-based management.

Layer 3 (Well-being outcomes): Stress, exhaustion, burnout, frustration, intention to leave, poor work-life balance.

The model suggests that Layer 2 dysfunctions exacerbate the negative impact caused by Layer 1 demands. The impact decreases when organizations adopt agility thoughtfully and follow a sustainable pace. The impact increases when adoption is superficial or hypocritical. To empirically test this framework, the instances in which organizations purposely decrease the harm through support interventions were examined.

The comparison shows how Agile can lead to very different outcomes depending on how it is practiced within an organization. In the existing approach, Agile creates manageable pressure that teams can usually cope with. However, the proposed system highlights that when Agile is affected by unhealthy practices like blame culture, superficial implementation, and unrealistic expectations, the stress experienced by teams becomes much more intense, increasing the chances of burnout and emotional exhaustion.

4. RESULTS AND DISCUSSION

4.1 Three Key Drivers of Agile-Induced Burnout

Through the review of 70 sources, three key drivers of stress, exhaustion, and burnout emerged repeatedly.

4.1.1 Iterative Delivery Pressure

Short sprints, typically one to four weeks in length, are an

integral part of Scrum and related approaches [2]. The presumption is that a team releases a working increment of the product in each sprint. Several studies noted that this leads to a stressful situation, where there is constant time pressure [4]. Developers complained about having no time to breathe since the next sprint is coming soon. McHugh et al. [19] reported that a developer stated that after finishing one sprint, they immediately moved into the next sprint without any downtime, resulting in continuous pressure.

According to Jafarzadeh et al. [20], such pressure results in developers adopting emotion-focused coping techniques, such as venting their frustration to colleagues. However, venting does not solve any problems; on the contrary, it aggravates them, since the problem remains unsolved (too much work with insufficient time). The researchers established a significant positive correlation between emotion-focused coping techniques and burnout ($b = 0.53, p < 0.001$). According to Mueller and Benlian [9], who studied the self-regulation costs of agile software development, the increased workload made even the beneficial practices, such as stand-up meetings, exhausting. They termed this situation as "being too drained from being agile".

Taken together, these findings suggest that the challenge is not simply the amount of work developers perform, but the continuous nature of agile delivery cycles. The absence of meaningful recovery time between sprints appears to amplify stress and gradually deplete individual coping resources. This indicates that sustainable pacing may be just as important as productivity when implementing agile practices.

4.1.2 Transparency as Surveillance

Agile thrives on transparency. The progress can be seen on the task boards; burndown charts are provided to management, while daily stand-ups require all individuals to disclose what they have done the previous day, what they will do that day, and any obstacles [2]. Although transparency is meant to foster trust, several studies show that it operates more as a means of surveillance.

According to Delisle and Coulon [14], there have been instances where daily stand-up meetings have become daily reporting sessions for the management team, rather than meetings between peers. As one of the participants said daily scrums become daily report to management. End-of-sprint becomes an occasion to fail rather than reflect on workload.

Singh & Strobel [21] conducted a study using a phenomenological approach about the experience of developers in stand-up meetings. In their findings, developers felt exposed, particularly those days when development is not progressing well. One participant said they feel bad having to go to a meeting every day to report my progress, particularly on those days where they spent hours trying to solve a bug. What was the finding? That stand-ups, despite their positive intentions, actually lead to anxiety rather than provide support. Khanagha et al. [12] explained that the feeling of being exposed was further aggravated by the pressure from peers since developers feel observed by both management and their peers. This "concertive control" [11] makes individuals work themselves to death just to avoid appearing incompetent among their peers.

Across these studies, transparency emerges as a double-edged feature of agile development. While it is intended to encourage collaboration and accountability, it can also create a constant sense of visibility and evaluation when implemented without psychological safety. As a result, practices designed to support

teams may unintentionally contribute to stress and anxiety.

4.1.3 The Autonomy Paradox

Agile offers freedom, yet it provides collective freedom rather than personal freedom [11]. The team makes decisions collectively, commits collectively, and succeeds or fails collectively. This might seem like a loss of freedom to an individual programmer.

Bjerre and Leimbach [11] explained that one of the major paradoxes of Agile is that although individual autonomy allows employees to choose when and how to perform tasks, this freedom is often overridden by the pressure to meet the expectations of other team members. Their study was based on previous research, according to which self-managing teams may control their members even more strictly than conventional managers could.

In Kakar [22]'s research, individual autonomy has been shown to have a negative impact on team collaboration in agile teams.

Developers who tried to do something differently than their peers met resistance or even silent exclusion. That is why they were forced to follow the herd in order to avoid conflict, despite knowing the alternative way of doing things. Such an approach kills innovation [8], but also leads to frustration and feelings of helplessness. Moreover, when examining the relationship between workforce agility requirements and burnout, it turned out that 72 percent of respondents in skill-based organizations were constantly changing tasks, while 67 percent were continuously reskilling. The correlation coefficient was $r = 0.64$.

The reviewed literature therefore suggests that autonomy in agile environments is more complex than it first appears. Although agile promotes independence and self-management, the collective nature of decision-making can create informal social pressures that limit individual freedom. This paradox helps explain why practices intended to empower employees may, under certain conditions, contribute to burnout instead.

Table 1. Summary of key Agile burnout drivers and their well-being impacts

Driver	Evidence from Literature	Well-being Impact
Iterative Delivery Pressure	Short sprints, continuous deadlines	Stress, exhaustion
Transparency as Surveillance	Daily reporting, visibility pressure	Anxiety, stress
Autonomy Paradox	Peer control, collective accountability	Frustration, burnout

4.2 How Implementation Pathology Worsens the Crisis

The three mechanisms above are problematic on their own. However, the analysis suggests that they become more severe when organizations adopt agile superficially or hypocritically.

4.2.1 Organizational Hypocrisy

According to Delisle and Coulon [14], organizational hypocrisy is the difference between rhetoric and practice in the organization. In agile environments, hypocrisy can be understood as proclaiming autonomy and sustainable pace while practicing micromanagement and overtime. Employees can spot this right away.

Among the most vivid examples of hypocrisy that Delisle and Coulon mention is the story from the Reddit post "I'm really fed up with Scrum please enlighten me". The poster describes how their manager said to them: "The team commits itself to a certain number of stories... It does not mean that you have to work overtime, but the stories must be delivered. Whatever it takes. Do not work overtime! But honor your commitment! PLEASE!!!!".

According to Jafarzadeh et al., [20], mindful individuals employ both problem-focused coping and emotion-focused coping. Problem-focused coping becomes ineffective if the hypocritical party is the organization itself, since one individual programmer lacks the capacity to solve the problem of inconsistent expectations set by the organization's management.

4.2.2 Guru-Driven Hype and Unrealistic Promises

According to Janes and Succi [13], some of the success of agile was attributed to the unrealistic promises made by "gurus." Consultants assured management that agile would increase team efficiency by tenfold without any changes to the infrastructure, additional training, or support. When these impossible expectations did not come true, the failure was attributed to the developers.

Delisle and Coulon [14] described a situation in which a consultant assured a vice president that Agile would improve efficiency tenfold. When the first sprint failed, the consultant blamed the developers by claiming that they were incapable of performing their jobs properly, while no training, infrastructure changes, or workload reductions were provided to support them.

Such a cycle of hype and blame results in what Janes and Succi [13] refer to as "The Dark Agile Manifesto," which is an ironic version of the true values of agility. One such value would be the following: "Individuals and interactions over processes and tools" turns into "Not having a process allows us to talk to people and do whatever we want."

4.2.3 Vague Definitions and Superficial Rituals

Janes and Succi [13] also highlighted that agile processes are often defined so vaguely that organizations can claim to practice agility simply by following rituals such as daily stand-ups and sprint planning, without truly embracing values like

trust, empowerment, and sustainability. They noted that employees were aware of this superficial implementation, with one respondent explaining that so-called “fake agile” or “fake scrum” was frustrating because organizations used agile terminology while implementing practices that were not genuinely agile.

Workforce agility research [23] found that role ambiguity due to dynamic job borders is an enormous burnout predictor. Role ambiguity always occurs in superficial agile implementation, causing workers to work more hours, agree to everything, and eventually burn out.

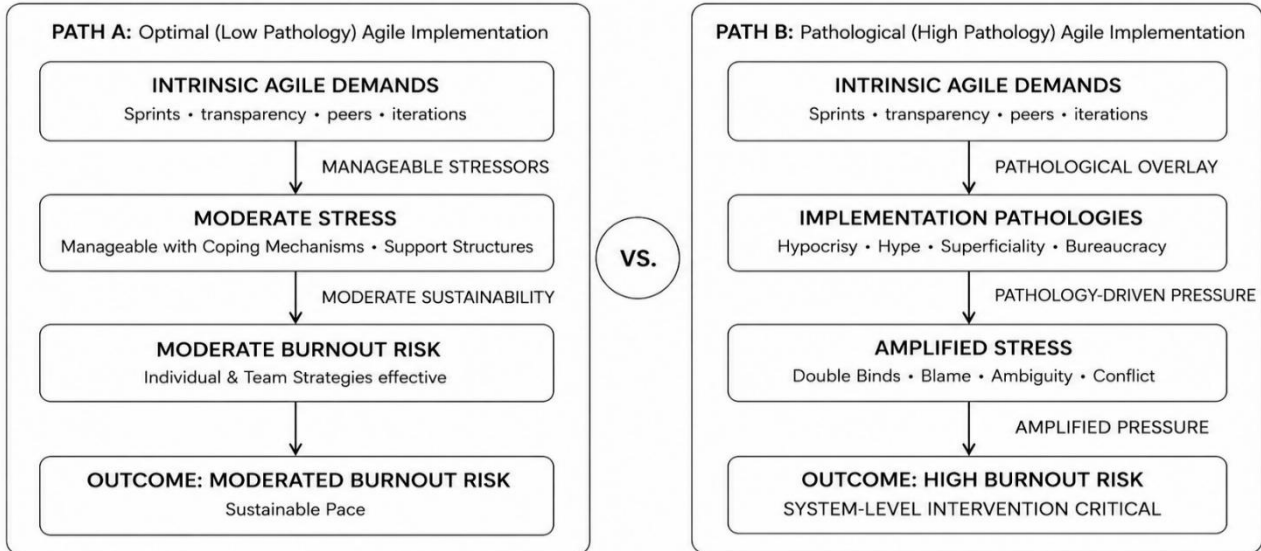


Fig.2. Comparative Model of Burnout Risk in Agile Environment

4.3 Comparison with Existing Frameworks

There are two fundamental differences between the theory proposed in this paper and existing ones. Firstly, the bulk of literature examines the well-being crisis separately from implementation failures. For instance, Meckenstock [4] provides 90 challenges, yet none of them are tied to organizational hypocrisy. Jafarzadeh et al. [20] concentrate solely on coping mechanisms and not on the problem itself. Secondly, critical literature about agile implementation [13] pays considerable attention to hypocrisy and hype; however, it does not assess the well-being of employees adequately.

By bringing these two perspectives together, the present study argues that the negative outcomes associated with agile development cannot be fully understood by examining either implementation failures or well-being issues in isolation. Instead, the interaction between these factors provides a more complete explanation of the burnout phenomenon.

4.4 Recommendations for Practice

Based on the findings in this paper, there are five recommendations for organizations serious about reducing agile burnout:

- Make sustainable pace actually mean something. While the Agile Manifesto values sustainable development, it doesn't really do much more than that. Make it a requirement, limit overtime, protect focus time, and adjust sprint requirements when there is an imbalance.
- Switch the meeting format around so that the pressure is not there. Stand-up meetings should be a time for coordinating efforts and asking for help, not reporting your progress to management. The manager must not attend the stand-up meeting as the evaluator. He or she can ask for an update in writing.
- Expose organizational hypocrisy. Management needs to

reflect on its own behavior. Are you preaching the importance of autonomy and yet requiring timesheets down to the hour? Is it all talk and no substance about sustainable pace while rewarding managers for delivering early?

- Ignore gurus' nonsense. If a consultant promises large efficiency improvements without mentioning any downsides, he is a salesman. Require proof and testable claims [13]. Make consultants prove not only what the teams have to do to implement agile, but also what they must give up to achieve it.
- Fix the system, not the developers. According to Jafarzadeh et al. [20], problem-focused coping leads to decreased levels of burnout, while emotion-focused coping, which includes venting, leads to increased levels of burnout. Do not hold “stress management” courses; rather, fix the overloaded workload, ambiguous roles, and inconsistent demands causing this stress.

4.5 Limitations

However, there are some limitations to this research. Firstly, this study is a systematic literature review where the results obtained previously by others were synthesized. Secondly, most of the studies included in the review were carried out in the field of software development in Europe and North America. Thus, the results may be not applicable to other industries and cultures. Thirdly, the extensive use of cross-sectional designs in the sources used limits the ability to establish the causality between agile practices and their effects. For instance, stressed developers can be more prone to noticing problems associated with agile practices, although reverse causality seems unlikely in this case due to consistency.

5. CONCLUSION

The question motivating this paper was how the agile methodology, which was meant to emancipate developers,

could be causing so many of them to suffer. With an integration of the well-being crisis approach, which emphasizes the pressures inherent in agile practices, and the implementation pathology approach, which stresses organizational hypocrisy, hype, and superficiality, things become much more clear. By reviewing 78 articles, three main mechanisms that cause agile burnout were identified: iterative pressure, transparency as surveillance, and the autonomy paradox. Individually, these mechanisms are harmful. However, when organizations implement agile hypocritically, the harm is amplified. Hypocrisy results in double binds for employees. Hype from gurus sets unrealistically high expectations, resulting in blame when the organizations fail. Superficiality allows organizations to do all the right things but ignore the values that could protect their workers. It is therefore unsurprising that a well-being issue has emerged in agile environments. After all, it is completely understandable, considering the current implementation of the methodology. However, this does not mean that agile must be given up altogether or developers must be asked to merely “cope” with the situation. What needs to be done is to adopt the values of agile, avoid making any false promises, and include well-being into the list of success criteria. Future research must avoid the trap of using cross-sectional survey data. The actual need is to study well-being longitudinally, i.e., in relation to pre and post-organizational transformation. Specifically, there is a need to contrast organizations that are committed to enhancing the well-being of their employees by means such as sustainable pace and psychological safety with those that are not. Moreover, future research should consider the impact of various contextual factors, including industry, country, and organization size. So long as the organizations do not put well-being first, agile shall keep sucking the very life out of its own advocates. This is the darker side of the phenomenon.

6. REFERENCES

- [1] K. Beck et al., "Manifesto for Agile Software Development," 2001. Available: <https://agilemanifesto.org/>
- [2] K. Schwaber and J. Sutherland, "The Scrum Guide," Scrum.org, 2020.
- [3] VersionOne, "The State of Agile Report," 2021.
- [4] J.-N. Meckenstock, "Shedding light on the dark side - A systematic literature review of the issues in agile software development methodology use," *Journal of Systems and Software*, vol. 211, p. 111966, 2024.
- [5] W. W. Royce, "Managing the development of large software systems," in *Proceedings of IEEE WESCON*, 1970, pp. 1-9.
- [6] Digital.ai, "17th State of Agile Report," 2023.
- [7] T. Dingsøyr, S. Nerur, V. Balijepally, and N. B. Moe, "A decade of agile methodologies: Towards explaining agile software development," *Journal of Systems and Software*, vol. 85, no. 6, pp. 1213-1221, 2012.
- [8] Benlian, "Sprint zeal or sprint fatigue? The benefits and burdens of agile ISD practices use for developer well-being," *Information Systems Research*, vol. 33, no. 2, pp. 557-578, 2022.
- [9] L. Mueller and A. Benlian, "Too drained from being agile? The self-regulatory effects of the use of agile ISD practices and their consequences on turnover intention," *Journal of the Association for Information Systems*, vol. 23, no. 6, pp. 1420-1455, 2022.
- [10] V. Venkatesh, J. Y. L. Thong, F. K. Y. Chan, H. Hoehle, and K. Spohrer, "How agile software development methods reduce work exhaustion: Insights on role perceptions and organizational skills," *Information Systems Journal*, vol. 30, no. 4, pp. 733-761, 2020.
- [11] S. Bjerre and T. Leimbach, "Understanding multilevel paradoxes in agile projects: between autonomy and control in self-managing teams," *Project Leadership and Society*, vol. 7, p. 100213, 2026.
- [12] S. Khanagha, H. W. Volberda, A. Alexiou, and M. C. Annosi, "Mitigating the dark side of agile teams: Peer pressure, leaders' control, and the innovative output of agile teams," *Journal of Product Innovation Management*, vol. 39, no. 3, pp. 334-350, 2022.
- [13] Janes and G. Succi, "The dark side of agile software development," in *Proceedings of the 2012 Agile Conference*, 2012, pp. 215-222.
- [14] J. Delisle and T. Coulon, "The dark side of agile: between hypocrisy, ambivalence and double binds," *Revue Internationale de Psychosociologie et de Gestion des Comportements Organisationnels*, vol. 31, no. 85, pp. 81-102, 2025.
- [15] Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," Keele University and Durham University, 2007.
- [16] G. Paré, M.-C. Trudel, M. Jaana, and S. Kitsiou, "Synthesizing information systems knowledge: A typology of literature reviews," *Information & Management*, vol. 52, no. 2, pp. 183-199, 2015.
- [17] Baham and R. Hirschheim, "Issues, challenges, and a proposed theoretical core of agile software development research," *Information Systems Journal*, vol. 32, no. 1, pp. 103-129, 2022.
- [18] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative Research in Psychology*, vol. 3, no. 2, pp. 77-101, 2006.
- [19] O. McHugh, K. Conboy, and M. Lang, "Using agile practices to build trust in an agile team: A case study," in *Proceedings of the International Conference on Information Systems Development*, 2011, pp. 91-102.
- [20] H. Jafarzadeh, H. Mosaber, and J. Sarabadani, "Burnout in agile teams: The role of mindful software development," *Information and Software Technology*, 2024.
- [21] K. Singh and J. Strobel, "Exploring lived experiences of agile developers with daily stand-up meetings: a phenomenological study," *Behaviour & Information Technology*, vol. 42, no. 4, pp. 403-423, 2022.
- [22] Kakar, "Engendering cohesive software development teams: Should we focus on interdependence or autonomy?" *International Journal of Human-Computer Studies*, vol. 111, pp. 1-11, 2018.
- [23] S. B. Saravanan and G. A. Kumar, "Workforce Agility Vs Employee Well-Being: Managing Burnout in Skills-Based Organisations," *Journal of Research and Development*, vol. 17, no. 11(III), pp. 52-55, 2022.