AIS Transactions on Human-Computer Interaction

Volume 17
Issue 3 *Special Issue on Flow, Immersion, and Presence*

Article 5

9-30-2025

When the Flow is Just Too Much: The Adverse Outcomes of Flow in Software Developers' Work

Saima Ritonummi
University of Jyväskylä, saima.e.ritonummi@jyu.fi

Valtteri Siitonen
University of Jyväskylä, valtteri.m.e.siitonen@jyu.fi

Markus Salo
University of Jyvaskyla, markus.t.salo@jyu.fi

Henri Pirkkalainen
Tampere University, henri.pirkkalainen@tuni.fi

Recommended Citation

Ritonummi, S., Siitonen, V., Salo, M., & Pirkkalainen, H. (2025). When the Flow is Just Too Much: The Adverse Outcomes of Flow in Software Developers' Work. *AIS Transactions on Human-Computer Interaction, 17*(3), 414-447. https://doi.org/10.17705/1thci.00231
DOI: 10.17705/1thci.00231

Transactions on Human-Computer Interaction

Volume 17 Issue 3

9-2025

When the Flow is Just Too Much: The Adverse Outcomes of Flow in Software Developers' Work

Saima Ritonummi

Faculty of Information Technology, University of Jyväskylä, saima.e.ritonummi@jyu.fi

Valtteri Siitonen

Faculty of Information Technology, University of Jyväskylä, valtteri.m.e.siitonen@jyu.fi

Markus Salo

Faculty of Information Technology, University of Jyväskylä, markus.t.salo@jyu.fi

Henri Pirkkalainen

Department of Information and Knowledge Management, Tampere University, henri.pirkkalainen@tuni.fi

Recommended Citation

Ritonummi, S., Siitonen, V., Salo, M., & Pirkkalainen, H. (2025). When the flow is just too much: The adverse outcomes of flow in software developers' work. *AIS Transactions on Human-Computer Interaction*, *17*(3), pp. 414-447.

DOI: 10.17705/1thci.00231

Available at http://aisel.aisnet.org/thci/vol17/iss3/5

Transactions on Human-Computer Interaction

Research Paper DOI: 10.17705/1thci.00231 ISSN: 1944-3900

When the Flow is Just Too Much: The Adverse Outcomes of Flow in Software Developers' Work

Saima Ritonummi
Faculty of Information Technology,
University of Jyväskylä

Valtteri Siitonen
Faculty of Information Technology,
University of Jyväskylä

Markus Salo
Faculty of Information Technology,
University of Jyväskylä

Henri Pirkkalainen
Department of Information and Knowledge Management,
Tampere University

Abstract:

Experiencing flow at work is often associated with enhanced motivation, creativity, performance, and job satisfaction. However, little is known about the potential downsides of flow in professional settings. To explore the negative outcomes of flow in software developers' work, we conducted a qualitative study involving in-depth interviews of 25 software developers. Through thematic analysis, we identified eight adverse outcomes of flow in both work and personal domains: maladaptive immersion, impaired self-regulation, false sense of productivity, depleted cognitive resources, physical strain, adverse emotional responses, dependence on flow, and work–life conflict. Notably, four of these outcomes—maladaptive immersion, impaired self-regulation, false sense of productivity, and adverse emotional responses–have not been previously documented in flow literature. These findings shed light on how such negative outcomes emerge and persist, revealing both immediate and long-term effects on software developers' personal and workplace well-being. Our empirical findings highlight the less explored negative aspects of flow, contributing to a deeper understanding of work-related flow. We offer practical guidance for software developers and their managers on how to avoid the pitfalls of excessive flow and maintain healthier boundaries at work.

Keywords: Flow State, Flow Outcomes, Adverse Outcomes, Work Performance, Well-being, Software Development, Knowledge Management

Yi Maggie Guo was the accepting senior editor for this paper.

1 Introduction

Flow refers to the intrinsically rewarding experience of being fully absorbed in an activity (Csikszentmihalyi, 1975, 1990). To enter a state of flow, an activity must offer challenges that align with an individual's skills (Nakamura & Csikszentmihalyi, 2002), such as engaging in work tasks that one finds interesting and enjoyable (Bakker, 2008). Flow can have significant positive outcomes, such as enhanced motivation and improved work performance (Knierim et al., 2019; Quinn, 2005). In this study, we focus on software developers' flow experiences for three key reasons. First, software development requires advanced analytical skills (Wyrich et al., 2019) and involves creative problem-solving throughout the IS development process (Riemenschneider & Armstrong, 2021). These characteristics provide frequent opportunities for developers to challenge themselves, which makes software development an ideal context for studying flow (Zubair & Kamal, 2015). Second, software development relies heavily on intensive information technology (IT) use, a critical aspect of information systems (IS) research. Developers must use a wide range of IT tools and technologies to perform cognitively challenging tasks, which require both broad and deep technological knowledge (Mattern et al., 2024). While IT use significantly influences developers' flow experiences, the impact of flow extends beyond IT use and affects outcomes such as work quality and well-being (Graziotin et al., 2018). Third, software development offers effective, immediate user feedback loops on one's actions (Kuusinen et al., 2016), and feedback constitutes a key condition for flow (Schaffer & Feng, 2022). Integrated development environments (IDEs) and other tools provide developers with rapid feedback that can reveal both their accomplishments and their skill-related limitations. This constant feedback and learning process make their work intrinsically rewarding yet cognitively demanding (Baltes & Diehl, 2018; Mattern et al., 2024). Furthermore, developers' intrinsic motivation stems not only from the satisfaction that they gain from building and refining information systems but also from the desire to create systems that enable end users to succeed in their endeavors (Beecham et al., 2008).

While researchers generally regard flow as positive, some evidence suggests that it can also have negative effects on individual performance and well-being. In fact, what should enhance productivity and performance can occasionally hinder it (Cachon, 2012). Impaired productivity and well-being may lead to adverse outcomes for developers and organizations, such as reduced motivation and project delays (Graziotin et al., 2018). Therefore, in this exploratory study, we focus on examining the paradoxical effects of flow in digital work (Dehnert & Friedrich, 2022) and on exploring how its adverse outcomes manifest in the software-development context.

Although researchers have paid significant attention to flow in the IS field, we still do not sufficiently understand its adverse outcomes in work contexts. The adverse outcomes of flow in software development, in particular, represent an interesting phenomenon because research generally considers flow a positive experience (Csikszentmihalyi, 1975, 1990) and has associated work-related flow with enjoyment, motivation (Salanova et al., 2006), and improved individual and group performance (Lin et al., 2009; Quinn, 2005). However, studies have observed that work-related flow can result in physical and psychological strain, unethical or risky behavior (Aleksić, 2016; Guptill, 2012; Liu et al., 2022), and work–family conflict (Feng et al., 2024; Ramsey & Lorenz, 2020). Furthermore, IS research on the harmful consequences of flow in contexts such as personal IT use (e.g., Khang et al., 2013; Theotokis & Doukidis, 2009) and sport (e.g., Partington et al., 2009) has discussed addiction and dependence. Moreover, software industry practitioners have talked about situations where the flow can turn against itself and even advised coders to avoid the flow zone, which can be a meditative state for the individual but lead to a false sense of hyperproductivity (Martin, 2011; Zuill, 2019). However, we do not yet have a comprehensive understanding of these adverse outcomes, how they manifest in software development, and the potential negative impact or paradoxical outcomes flow experiences might have on software developers' work. These outcomes can be serious for work performance (e.g., making mistakes when coding for hours in a flow state) and well-being (e.g., neglecting nutrition, sleep, and social relationships). Thus, they are worth investigating and being aware of.

To understand why something that researchers generally consider an optimal and desirable state (and should improve performance) can hamper work performance, we pose the following research question:

RQ1: What adverse outcomes does flow have in software developers' work, and how do they manifest?

To gain a deep, contextual understanding of the topic, we used qualitative interviews to collect the data (Myers & Newman, 2007). After interviewing 25 software developers and conducting a data-driven thematic analysis (Braun & Clarke, 2006), we identified eight different adverse outcomes of flow. These outcomes have immediate and long-term effects at the individual and organizational levels. This study contributes to the research stream on the negative effects of flow (e.g., Liu et al., 2022; Ramsey & Lorenz, 2020) and to IS research by advancing our understanding of the adverse outcomes of flow in technology-mediated knowledge work (e.g., Adam et al., 2024; Nadj et al., 2023). The practical implications include improving employees' self-awareness of the pitfalls of flow, taking care of one's mental and physical resources, and avoiding overexertion. Managers can use these insights to assess when it is worthwhile to encourage employees to get into flow and when, on the contrary, it might not be advisable.

2 Related Work

2.1 Flow

The concept of flow refers to the "optimal experience". It is a state of deep absorption in an enjoyable, intrinsically motivating activity (i.e., autotelic activity) (Csikszentmihalyi, 1975, 1990). To reach flow, an activity needs to present increasingly complex challenges (action opportunities), and the individual needs a specific set of skills (action capabilities) to take up those challenges (Nakamura & Csikszentmihalyi, 2002). While researchers often consider the perceived balance between challenges and skills the most critical condition for flow, clear goals and immediate feedback about the progress one makes also play an important role in whether one reaches and stays in the flow state (Nakamura & Csikszentmihalyi, 2002). When these conditions (i.e., antecedents of flow) are met, six defining characteristics come into play for conceptualizing the flow state: focused concentration, loss of reflective self-consciousness, merging of action and awareness, altered sense of time, sense of control, and autotelic experience (the experience of engaging in the activity as rewarding in and of itself) (Nakamura & Csikszentmihalyi, 2002). Research also describes the metacognitions of flow, which describe individuals' beliefs about the usefulness of being in flow and their confidence in their ability to self-regulate flow experiences (i.e., exert control over the situation) (Wilson, 2016). Furthermore, flow is a positive emotional state that one can observe beyond individuals' own perceptions by observing their physiological (e.g., heart rate) and psychophysiological (brain activity) bodily activities (Rissler et al., 2018; Van der Linden et al., 2021).

2.2 Experiencing Flow at Work

In the work context, researchers have often described flow with three characteristics: absorption, enjoyment, and intrinsic work interest (i.e., motivation) (e.g., Bakker, 2008; Rodríguez-Sánchez et al., 2011; Salanova et al., 2006). Researchers have also investigated flow in terms of work performance (e.g., Quinn, 2005) and found an association between work-related flow and positive outcomes such as vigor, knowledge sharing, and inter-employee helpfulness in organizations (Lin et al., 2009). Work-related flow can also help mitigate adverse work outcomes such as exhaustion (Demerouti et al., 2012). Moreover, cooperation and experiencing flow in small workgroups can improve self-efficacy beliefs (Salanova et al., 2014), group performance, and satisfaction (Knierim et al., 2019). On the other hand, we lack an understanding of the connection between flow's outcomes and the paradoxes of digital working methods, such as the paradox of control and chaos (i.e., being unable to influence the process or outcome while using IT) (Dehnert & Friedrich, 2022), the productivity paradox (i.e., thinking something will increase productivity when it, in fact, does not) (Balijepally et al., 2009), and other adversarial phenomena associated with technology-mediated work (e.g., the draining effects of agile development practices (Mueller & Benlian, 2022)). These paradoxes share the same key characteristic: contradiction (i.e., simultaneous contradictory elements can be present in a situation and can create both positive and negative outcomes) (Schad et al., 2016).

Software development exemplifies the kind of work that involves cognitively demanding tasks (Mueller & Benlian, 2022) and that provides opportunities to challenge oneself, learn new skills, and apply both analytical skills and creativity (Kalantari & Lethbridge, 2022; Pratt et al., 2016). Thus, it provides an excellent opportunity to experience work-related flow. On the other hand, many development tasks require significant mental effort and impose a high cognitive load (Gonçales et al., 2021). Researchers have also examined systems development flow (e.g., O'Connor et al., 2024) and flow techniques for managing workflow (e.g., Dennehy & Conboy, 2017). However, we focus on software developers' psychological flow states. Studies on work-related flow in the software industry have demonstrated that experiencing flow is important for software developers' creativity, enjoyment, and motivation (Pratt et al., 2016; Ritonummi et al., 2023; Zubair & Kamal, 2015) as well as the productivity and success of development projects (Brown et al., 2023a, 2023b; Calais & Franzini, 2023). Some studies have investigated the barriers to experiencing flow in software developers' work (Ritonummi et al., 2024), which, by and large, include (both technology-mediated and non-technology-mediated) distractions and interruptions (Abad et al., 2018; Chen et al., 2022; Züger et al., 2017).

Moreover, a key challenge in the software industry is expertise coordination (Kudaravalli et al., 2017) and finding ways to provide software developers with enough—but not too many—challenges (Ritonummi et al., 2024). Another important consideration in developers' work is supporting flow in the IT use context, which includes, for example, improving the experience that developers have with development tools, such as IDEs (Kalantari & Lethbridge, 2022; Kaufhold et al., 2019; Kuusinen et al., 2016; Murphy, 2014; Sheth et al., 2011). Furthermore, some studies have examined flow recognition, including identifying whether a developer is in the flow state. These studies have examined biometric measures, such as heart rate, eye movement, and brainwaves (Müller & Fritz, 2015) and tracked interaction with IDEs (e.g., Cao et al., 2020; Cowley et al., 2022; Zheng et al., 2019).

Optimally challenging and stimulating work plays a crucial role in developers' flow, productivity, and overall fulfillment (i.e., how much they enjoy work and how they perceive their future in their organization) (Brown et al., 2023b; Greiler et al., 2022). Therefore, we need to examine how flow manifests in their work. While IS research often focuses on flow experiences in personal or work-related IT use, developers experience flow both in and beyond IT use. Their profession requires them to not only master complex information systems and perform challenging tasks with them but also continually invest time in learning new technologies (Baltes & Diehl, 2018; Mattern et al., 2024) while receiving constant feedback on their expertise and quality of their work (e.g., in code reviews) (Baltes & Diehl, 2018). As users of development tools such as IDEs, they value effective and timely feedback loops (Kuusinen et al., 2016), which play a critical role in not only their flow experience but also their work activities (Kalantari & Lethbridge, 2022; Ritonummi et al., 2023). Furthermore, their motivation arises from personal satisfaction in producing quality work and from helping others (Beecham et al., 2008) by creating high-quality documentation for fellow developers and designing user interfaces that benefit end users (Galster et al., 2023).

Furthermore, while some studies have investigated flow as originally conceptualized, most IS research has addressed flow specifically in technology-mediated environments. These conceptualizations characterize flow in the human–computer interaction (HCI) context via involvement, enjoyment (e.g., Ghani et al., 1991), control, attention focus, curiosity, and intrinsic interest (e.g., Webster et al., 1993). Other concepts in the IS literature that are based on the concept of flow include cognitive absorption (characterized by temporal dissociation, focused immersion, heightened enjoyment, control, and curiosity) (Agarwal & Karahanna, 2000) and technoflow (characterized by control, attention, curiosity, and intrinsic interest) (Califf et al., 2020). However, many IS flow studies have discussed flow in the personal IT use context, such as online customer experience (e.g., Guo & Poole, 2009; Jiang & Benbasat, 2004; Kim et al., 2013; Koufaris, 2002; Nah et al., 2011), social media use (e.g., Huang et al., 2014; Pelet et al., 2017), and gaming (e.g., Hsu & Lu, 2004; Sepehr & Head, 2018), or in educational settings, such as online learning (e.g., Guo et al., 2016; Wang & Hsu, 2014). Some IS studies have even considered flow experience according to its original conceptualization to examine work IT use (Ghani et al., 1991; Kaufhold et al., 2019; Keith et al., 2016; Knierim et al., 2019; Suh et al., 2017) and work-related flow beyond the HCI context (Adam et al., 2024; Dehnert & Friedrich, 2022; Nadj et al., 2023). This stream of IS research on work-related flow has focused on supporting and mitigating disruptions to flow (Adam et al., 2024; Nadj et al., 2023). However, while such research has increased our understanding of work-related flow, we have limited understanding of the adverse outcomes of flow among knowledge professionals. Thus, in this study, we focus on the "total" work-related flow experience (i.e., not constrained to the IT use context only) and on the outcomes of flow (i.e., flow consequences) and their effects on software developers' cognition, affect, behavior, and performance (Rissler et al., 2017).

2.3 The Negative Side of Flow

Researchers often consider flow as a primarily intrinsically motivated, enjoyable experience (Csikszentmihalyi, 1990) with positive work-related outcomes, such as better performance, better work engagement, and less burnout (Lavigne et al., 2012; Mosing et al., 2018; Weintraub et al., 2021). However, in this paper, we focus on possible adverse outcomes of flow in software developers' work. The adverse outcomes of flow have gained some interest in fields such as psychology, behavioral science, IS, and HCI, but we do not fully understand their implications in the work context. Research that has addressed the adverse outcomes of flow or excessive flow has discussed dependency and addiction, risky or impulsive behavior, mental load, and work–family conflict (see Table 1 for a summary).

Table 1. Research on Adverse Outcomes of Flow Experiences

Context	Authors	Topic (flow outcomes)	Key findings on adverse outcomes of flow
Sports	Partington et al. (2009)	Compulsive engagement (dependence) in big wave surfing	Experiencing flow in high-risk sports can contribute to: Becoming dependent on flow (Partington et al., 2009) Engaging in risky behavior (Schüler & Nakamura, 2013; Schüler & Pfenninger, 2011)
	Schüler & Pfenninger (2011)	Underestimation of risk in kayaking	
	Schüler & Nakamura (2013)	Impaired risk awareness and risky behavior in rock climbing and kayaking	
	Boudreau et al. (2020)	Review of the consequences of flow in adventure recreation	
IT use	Theotokis & Doukidis (2009)	Tendency for addiction with social information systems	Technology-mediated flow can result in: Compulsive IT use and addiction (Ahmed & Dixon, 2023; Jo & Beak, 2023; Johnson et al., 2020; Khang et al., 2013; Stavropoulos et al., 2018; Zhang et al., 2014) to social media use (Brailovskaia & Teichert, 2020; Brailovskaia et al., 2020; Theotokis & Doukidis, 2009), gaming (Hu et al., 2019, 2022; Wu et al., 2013), and online shopping (Barta et al., 2022; Mason et al., 2022) Counterproductive behavior, for example, in engaging in online gambling and feeling good despite losing money (Dixon et al., 2018; Lavoie & Main, 2019) Physical and psychological strain, such as increased cortisol levels and mental workload (Keller et al., 2011)
	Keller et al. (2011)	Strainful tensions and mental load in computer games	
	Khang et al. (2013)	Digital media addiction	
	Wu et al. (2013)	Online game addiction	
	Zhang et al. (2014)	Compulsive smartphone use	
	Andrade et al. (2016)	Addiction and dependence in a virtual learning environment	
	Stavropoulos et al. (2018)	Internet addiction	
	Dixon et al. (2018)	Gambling with multi-slot machines	
	Hu et al. (2019)	Internet gaming disorder	
	Lavoie & Main (2019)	Online gambling	
	Brailovskaia & Teichert (2020)	Addictive social media use	
	Brailovskaia et al. (2020)	Addictive social media use	
	Johnson et al. (2020)	Compulsive mobile use	
	Barta et al. (2022)	Impulse buying and consumer regret in online shopping	
	Hu et al. (2022)	Online gaming addiction	
	Mason et al. (2022)	Compulsive online buying	
	Ahmed & Dixon (2023)	Problematic social media use	
	Jo & Baek (2023)	Mobile app addiction	
Work	Guptill (2012)	Detrimental health effects for professional musicians	Work-related flow can result in Physical strain, such as injuries (Guptill, 2012) Unethical or risky behavior (Aleksić, 2016; Liu et al., 2022) Cognitive strain, such as decreased attentional performance (Liu et al., 2022) Work–family conflict (Feng et al., 2024; Ramsey & Lorenz, 2020)
	Aleksić (2016)	Unethical behavior at work	
	Ramsey & Lorenz (2020)	Unintended negative work–family conflict	
	Liu et al. (2022)	Stronger preference for risk (during) and compromised attentional performance (after)	
	Feng et al. (2024)	Flow rumination and work–family conflict	

First, research has addressed the link between flow and addiction or risky behavior in sports (and specifically high-risk, extreme sports such as big wave surfing (Partington et al., 2009) and rock climbing (Schüler & Nakamura, 2013)). This research has found extreme sports to be associated with becoming addicted to flow or dependent on the sensation of being in the flow state (Partington et al., 2009) and with risky behavior resulting from impaired risk awareness while in the flow state (Schüler & Nakamura, 2013; Schüler & Pfenninger, 2011). IS research has also discussed flow and addiction—in particular, addiction and compulsive IT use in activities such as gaming and gambling (e.g., Dixon et al., 2018; Hu et al., 2019, 2022; Lavoie & Main, 2019; Wu et al., 2013), social media use (Brailovskaia & Teichert, 2020; Brailovskaia et al., 2020; Theotokis & Doukidis, 2009), and online shopping (Barta et al., 2022; Mason et al., 2022). Furthermore, in their study, Keller et al. (2011) identified strainful tension and mental load as negative flow side effects in the HCI context. In the work context, studies have questioned the assumption that work-related flow is always positive and, by challenging this view, have found that flow can have negative outcomes on work–life balance (namely, work–family conflict) (Feng et al., 2024; Ramsey & Lorenz, 2020). Studies that have addressed potential adverse outcomes of flow on work performance have discussed risk-taking during flow (Aleksić, 2016; Liu et al., 2022) and compromised attentional performance shortly after flow due to individuals' consuming energetic and cognitive resources during the flow state (Liu et al., 2022). In addition, flow states might cause detrimental health effects (e.g., injuries) (Guptill, 2012) and (unintentional) unethical behavior (i.e., deviating from one's values and principles) at work (Aleksić, 2016). Another important question regarding the negative aspect of work-related flow is whether individuals experience sufficient recovery after flow (Liu et al., 2022).

Since the flow concept is synonymous with "optimal experience", one could question whether an experience that is anything but optimal (causing, for example, problematic behavior) is compatible with a flow experience (Barberis et al., 2022; Kuss & Griffiths, 2012). While the flow experience certainly constitutes an optimal experience, it does not necessarily have all positive and "optimal" outcomes (Zimanyi & Schüler, 2021) (similar to how stress comes in both a negative (distress) and positive (eustress) form (Keller et al., 2011; Tarafdar et al., 2019)). While many studies have found flow in the IT use context to be linked with problematic behavior and addiction, Sepehr and Head (2012) distinguished between addiction and the flow state and suggested that obsessive passion is linked with addiction and harmonious passion with flow, which can, respectively, decrease or increase subjective well-being (the outcome) (Barberis et al., 2022; Sepehr & Head, 2012). Similarly, Ahmed and Dixon (2023) considered absorption an adaptive correlate of the flow state and dissociation a maladaptive correlate of absorption (Ahmed & Dixon, 2023). However, as Csikszentmihalyi (1990) pointed out, flow is not good in an absolute sense: people risk learning to enjoy an activity so deeply that "normal" activities pale in comparison to flow experiences and, thus, becoming dependent on this very limited range of action opportunities such that they ignore other opportunities to develop their skills and find enjoyment (Csikszentmihalyi, 1999). Further, individuals could seek flow in destructive activities, such as in the case of antisocial behavior (Keller & Bless, 2008; Schüler & Pfenninger, 2011), rather than positive activities (Nakamura & Csikszentmihalyi, 2002). Thus, individuals need to find a balance between when it is appropriate to be in the flow and when it leads to undesirable outcomes, such as overinvolvement (Csikszentmihalyi, 1975), increased time to task completion (Webster et al., 1993), or task neglection (Trevino & Webster, 1992).

To summarize, research on the adverse outcomes of flow has identified:

1) Dependence/addiction in IT use and sports,
2) Risky, counterproductive, or unethical behavior when engaging in IT use, sports, or work,
3) Physical and psychological strain resulting from IT use or working, and
4) Work–family conflict resulting from the work-related flow.

In particular, the adverse outcomes identified at work include physical and psychological strain, unethical behavior, and work–family conflict. However, this body of work mostly focuses on one or two negative consequences at a time. Hence, we do not have a comprehensive understanding of adverse outcomes of flow in the work context. Moreover, we need to understand how these consequences manifest in a concrete way to mitigate the downsides of flow. Given these points, we have limited understanding of the adverse outcomes of flow among workers both outside and in the IT use context (i.e., regarding flow and technology-mediated flow).

3 Research Method

To understand flow experiences and their potential adverse outcomes in software developers' work, we used a qualitative research approach and conducted semi-structured, in-depth interviews. Qualitative methods are particularly well-suited for investigating complex social and organizational phenomena (Myers, 1997), and interviews are effective for capturing the nuanced and context-specific nature of such experiences (Myers & Newman, 2007). In this study, we focused on software developers' flow experiences and how they perceive the outcomes of work-related flow.

While much of the existing literature portrays flow as inherently beneficial (Zimanyi & Schüler, 2021), our objective was to go beyond this consensus by examining potential downsides of flow in the context of software development. To uncover the varied adverse outcomes of flow and how they manifest in software development, we gathered in-depth narratives and descriptions of actual flow episodes—an area where qualitative interviews offer rich and detailed insights. We analyzed the data using thematic analysis, a data-driven method well-suited for developing deep understanding of data (Schultze & Avital, 2011), especially when investigating an under-researched topic (Braun & Clarke, 2006).

3.1 Data Collection

We conducted the interviews between December 2022 and February 2023. We interviewed software developers since their work requires high analytical and problem-solving skills and involves intense IT use, which both provide great opportunities to experience flow. By focusing on software developers, we could uncover specific characteristics related to their flow. Using purposeful sampling (Myers, 2019), we looked for software developers who could provide rich information on their flow experiences. To reach professional developers, we contacted Finnish software houses and industry forums (e.g., Slack channels and trade unions) and asked them to forward an invitation to participate in the study to their employees/members. As criteria to participate in the study, individuals needed to work as a full-time software developer and to have experienced flow at some point in their career. Interviewees participated in the study on a voluntary basis and received a small product reward (coffee, tea, and chocolate) for their time. To ensure we collected data from developers who represented various voices (Myers & Newman, 2007), we interviewed 25 software developers with varying degrees of experience in the field and who worked in different roles (e.g., junior and senior developers, team leads, and scrum masters). The interviewees worked in software houses (consulting), in in-house roles in private organizations and the public sector, and as entrepreneurs in the software industry. We conducted semi-structured interviews to allow flexibility for concepts to emerge and to enable varied topics of interest to emerge that are meaningful to the participants. This format also allowed us to focus on the interviewees' individual lived experiences, enabling them to articulate their perspectives in their own words (Myers & Newman, 2007), thereby contributing to a rich and nuanced understanding of the topic. We continued to collect data until we reached theoretical saturation, when additional interviews no longer yielded new or relevant insights.

We conducted the interviews online as video interviews, and they lasted between 30 minutes and one hour each (53 minutes on average). We then fully transcribed the interviews. We informed the interviewees beforehand about the study topic ("an interview study about software developers' flow experiences at work") and the university's privacy policy on processing personal data. When scheduling the interviews, we also provided participants with a brief explanation of the concept of flow to ensure a shared understanding aligned with flow theory: "Flow experience is a state of intense concentration; you are completely absorbed in what you are doing and lose track of time. You experience the right amount of challenge, and the activity is rewarding in and of itself".

We show how two developers described their flow in the coding excerpts in Appendix A. Before we began each recording, we briefly introduced ourselves and the study's purpose to make the interviewees feel comfortable and create a trusting atmosphere (Myers & Newman, 2007). We also told them about confidential disclosure (Myers & Newman, 2007) and that we would anonymize the data and asked their permission to record the interview for transcription. The interviewees completed the background information questionnaire at this stage. At the end of each interview (after we stopped the recording), we briefly discussed with individual interviewees how they felt about the interview, asked them if they could think of any essential issues that we did not address in the interview (giving them the freedom to discuss these topics further if needed), and thanked them for their time.

The interview scheme comprised the following main themes:

1) Background questions about the work context (how long they had worked in the industry and what kinds of software they developed),
2) Describing a flow experience (what they were working on, what was happening around them, how they felt during the experience),
3) Work conditions regarding flow (describing how well the challenges that their current job presented matched their professional skills, if they enjoyed their work, and if they felt motivated and discussing if technology use played a role in their flow experiences), and
4) Flow outcomes (what happened after the flow, how they felt after the experience).

We designed the questions to gather detailed information about the experiences to make sure that the experiences in question constituted flow (as opposed to, for example, workflow). Because we used a semi-structured interview scheme, we used the interview scheme themes but also allowed the conversation to flow freely (Myers & Newman, 2007), which gave the interviewees the opportunity to discuss topics they found particularly interesting related to experiencing flow in their work. This approach allowed unexpected topics to emerge. The emergent topic that we discuss in this paper concerns the adverse outcomes of flow (i.e., the last theme presented in the previous paragraph), which the interviewees raised when reflecting on their experiences, and what happens after flow or how they felt after flow.

On average, interviewees were 35 years old. They ranged from 25 to 47 years old, which concurs with Stack Overflow's (2023) developer profile that shows around two-thirds of professional developers fall between 25 and 44 years old. Although our sample contained predominantly men (76%) and, thus, only 24 percent women, these figures concur with the global gender distribution of software developers (Stack Overflow, 2022). Most interviewees had either a bachelor's degree (44%) or a master's degree (32%), and all lived in Finland (24 reported their nationality as Finnish and one as Chinese). All worked full-time as developers, and three worked as entrepreneurs in the software industry. In total, 80 percent of the interviewees had more than six years of experience working in software development. We present the interviewees' demographic information in Table 2.

Table 2. Interviewee Demographics

Age		**Working mainly**	
Average	35 years	In a team	22
Range	25–47 years	Independently	3
Gender		**Employment**	
Female	6	Full time	22
Male	19	Entrepreneur	3
Other	0		
Education		**Experience in software development**	
Secondary education	5	More than 10 years	15
Bachelor's degree	11	6–10 years	5
Master's degree	8	3–5 years	4
Doctoral degree	0	1–2 years	0
Other	1	Less than 1 year	1

Most interviewees (88%) worked in a team setting (e.g., in a developer team). Their roles varied from junior (e.g., junior web developer) to middle (e.g., software developer, frontend developer) and senior positions (e.g., senior software engineer, senior software developer). Four interviewees also had experience in one or more of the following roles: lead developer, scrum master, and team lead.

3.2 Data Analysis

Since the goal of this study was to gain in-depth insights into the less explored aspects of flow experiences, thematic analysis was deemed a suitable approach for analyzing the interview data. Thematic analysis provides a useful way to identify patterns in data and report them in rich detail (Braun & Clarke, 2006; Terry et al., 2017). At the beginning of the analysis process, we followed a data-driven approach to thematic analysis (i.e., we ensured a strong link between codes and the data) (Braun & Clarke, 2006) because the topic—the negative side and adverse outcomes of flow that the interviewees reported having experienced—represents a relatively under-researched area in the work context, and no pre-existing coding scheme (i.e., a set of codes) existed. The unit of analysis was an individual's perceptions of the outcomes of flow that had affected them in a harmful or detrimental way. As the analysis progressed, we used both a data-driven and theoretical coding approach (Terry et al., 2017) when finalizing the definition of each category.

To analyze the data, we followed the six-step instructions for thematic analysis suggested by Braun and Clarke (2006). First, the first author familiarized herself with the interview data by reading and re-reading the transcripts; looking for patterns, meanings, and potentially interesting topics in the data; and writing down initial ideas based on these notions. Second, the first author extracted the interesting details from the data and generated the initial codes via open coding (Glaser & Strauss, 1967) during which she moved back and forth between the data and the coded extracts. This process involved extracting all parts of each interview where the interviewee mentioned something about flow having adverse/negative outcomes or arousing conflicting or mixed emotions and collecting these quotes. The first author constantly compared the codes with existing codes to identify differences and similarities (i.e., to identify the need for a new code) and used the phrases that the interviewees used when labeling the codes whenever possible, since interviewees represent "creative interpreters of their worlds as we are of theirs" (Myers & Newman, 2007, p. 17). In accordance with the data-driven approach, the first author completed the first coding round before all authors engaged with the literature. In the third step, we (all authors) looked for themes (i.e., identifying what the codes revealed and gathering all relevant data under a potential outcome category) in the literature. For example, we coded over-engagement under maladaptive immersion. We ensured consistency by discussing the codes among ourselves. In the fourth step, we reviewed the identified categories by comparing the coded extracts and generating a thematic map. We then discussed the categorization and hierarchy of codes to establish categories (i.e., outcomes) and subcategories (i.e., manifestations). We collaboratively discussed intermediary results (e.g., we discussed each category's name and which manifestations we include in which category) to verify the conclusions that we drew from the data (Lune & Berg, 2017). In qualitative research (especially when using a data-driven approach), where coding is "an organic and flexible process" that requires detailed engagement with the data, this approach is useful for providing accuracy through immersion (Terry et al., 2017, p. 20) in contrast to practices such as calculating inter-coder reliability, which is based on a positivist conception of reliability that represents consistency of coding rather than coding accuracy (Braun & Clarke, 2013; Terry et al., 2017). To reach an agreement on the categorization, we discussed and refined the categories. Through these discussions, we agreed on names for three categories (i.e., *maladaptive immersion*, *impaired self-regulation*, and *false sense of productivity*) and continued to discuss the others. In our discussions, we focused on how to best label the categories (i.e., find the most descriptive and accurate name for each category) and on clarifying similarities and differences between categories to ensure we defined them in a mutually exclusive manner to avoid overlaps. Ultimately, we excluded everything that did not explicitly constitute an adverse or negative flow outcome (e.g., descriptions of being on "autopilot" or "in a bubble" while in flow as opposed to maladaptive immersion, which described a state of immersion so deep that it affected the interviewees' other activities negatively). In the fifth step, we finalized the category labels for the remaining five categories: *depleted cognitive resources, physical strain*, *adverse emotional responses*, *dependence on flow*, and *work–life conflict*. We gave each adverse outcome a name and a definition, which included the different ways it could manifest (i.e., subcategories). While we developed some codes based on the data and how the interviewees described their experiences, in some instances, we used existing phrasings from prior research, such as reduced attentional performance (Liu et al., 2022), dependence (Partington et al., 2009), and flow rumination (Ramsey & Lorenz, 2020). Finally, in the sixth step, we created a collected "codebook" that listed the adverse outcomes, their manifestations, descriptions for each manifestation, and compelling extracts/quotes that captured the phenomenon well (Braun & Clarke, 2006). While we followed a data-driven analysis approach in the first four steps, we used a more theoretical approach in the last two steps. We provide examples of how we coded two interviewees' transcripts in Appendix A.

Unlike qualitative content analyses, thematic analyses do not focus so much on reporting quantified measures but on ensuring that themes capture something relevant in terms of answering a research question (i.e., more instances do not equal more critical themes). While we encountered an ideal situation in that we identified several instances in each identified outcome across the dataset, we do not report them based on quantifiable measures (e.g., how many instances of each theme are in the dataset). Because we focused on understanding *how* the adverse outcomes of flow manifest in software developers' work, the analysis process focused on that aspect. Thus, in the analytic process, we focused not only on describing what was going on in the data but also on interpreting what those patterns mean and their significance (Braun & Clarke, 2006). Finally, we considered how other factors may have contributed to the formation of the adverse outcomes of flow experiences. However, while factors such as work pressure may have influenced the developers' work overall, flow played a significant role in the reported experiences, and the interviewees associated these adverse outcomes with having experienced flow.

4 Findings

From interviews with 25 software developers, we identified eight distinct adverse outcomes of flow: maladaptive immersion, impaired self-regulation, false sense of productivity, depleted cognitive resources, physical strain, adverse emotional responses, dependence on flow, and work–life conflict. While the interviewees predominantly viewed flow as a positive phenomenon in their work, they vividly described the adverse outcomes of work-related flow in varying ways. For instance, they described how the flow state could begin as an enjoyable state with the right amount of focus and engagement but later became excessive in situations where they experienced these adverse outcomes. They described the outcomes as varying in having short- or long-term effects. For example, maladaptive immersion usually has an immediate and shorter-term effect, whereas work–life conflicts could have a more prolonged impact on an individual's well-being. Furthermore, while we focus here on individuals' experiences and the adverse outcomes affected the interviewees personally, some of them also had a negative impact at the team or even organizational level.

4.1 Maladaptive Immersion

Maladaptive immersion manifested as over-engagement (i.e., as being too immersed in the flow state). This state of maladaptive immersion, which six interviewees described, involved ignoring everything else happening around them, either because they consciously blocked out everything other than the task at hand or because they "organically drift[ed] into the flow state" (Application specialist 1). While flow involves deep immersion in an activity, which researchers generally consider a positive characteristic, the interviewees identified circumstances where maladaptive immersion was not appropriate for the setting. For example, senior software engineer 1 said:

> *I don't think it's a healthy flow if you cannot be interrupted. I'm not trying to go after, like I'd be in some kind of trance. That's because I think it's an unpleasant hyperfocus, which leads to you being for hours on end—whether you are working alone or with two or 50 people—doing things with tunnel vision and not paying attention to what is happening around you.*

The developers also discussed the immersiveness of using IT: while physical space creates certain constraints on engaging in activities (e.g., limitations on how long or how deeply one can become immersed in a work-related task), software development (i.e., using IT) seemingly imposes no boundaries on how deeply one can immerse oneself in the flow state. Another concern related to maladaptive immersion was related to whether it was healthy to be so immersed that one did not register anything about one's surroundings or the passage of time. Being too deeply immersed in a flow state caused the interviewees to forget to eat, drink, or take breaks. In addition, they reported a strong flow state to easily lead to working overtime and even working during the night. Although maladaptive immersion is an immediate consequence, the interviewees also raised it in relation to rumination and difficulty detaching from work (or thoughts related to work) and neglecting their well-being (see also Section 4.8). Note that maladaptive immersion differs from descriptions such as "being on autopilot" or "in a bubble". Whereas the interviewees interpreted these kinds of flow states as neutral (or even positive) experiences, they saw maladaptive immersion as a negative, undesirable state of being *too* immersed in a hyperfocus-like state in which they could not change contexts or step outside the situation.

4.2 Impaired Self-regulation

Moreover, 13 developers described impaired self-regulation as a negative side effect of flow in this excessively immersed state. Sometimes, developers' intense flow states resulted in their feeling that they no longer had control over their actions (e.g., getting stuck in the flow and unable to get out of it). Senior software engineer 1 described the experience as follows:

> *Usually, if I jump into flow alone at work, it takes me several hours before I realize where I am and what I'm doing. I forget sometimes to stop… So usually, for me, they have been pretty harmful sessions—those flow situations—where I've gotten stuck in doing something.*

Lack of control was also embodied in having trouble switching context to the next action, either at work (e.g., hopping into a meeting in the middle of or right after flow) or from work to leisure. Software developer 1 described this feeling as having a hard time winding down or "coming down to earth" after intense flow:

> *You will notice afterward, for example, that if you have listened to fast-paced heavy metal music and concentrated intensely [in flow], your brain will remain in a pretty fast-paced mode afterward, and it may be difficult to come back down to Earth… If you have a meeting right after a flow of this kind, you suddenly feel that everyone is talking so slowly, and you're like, "forward, forward, forward, forward.*

Further, the trouble switching context not only affected interviewees' ability to move on to the next task but also their ability to come back to the task that they worked on during the flow state. The interviewees reasoned that this phenomenon occurred because, in the flow state, one has such a clear vision when absorbed in the task that it can be very hard to re-gather its context and goals when snapped out of the flow state, or, if the workday gets interrupted mid-flow, it can take considerable effort the next day to come back to the task. On the other hand, if one is not so deeply immersed, one might be able to switch the context and level of thinking more easily, which can prove beneficial for solving the problem. The interviewees saw this ability to switch more easily as important, especially when working on complex tasks (i.e., when not being able to step outside of a situation might hamper their ability to solve a problem). For example, developer 1 said: "I have come to the conclusion that it [flow] is not always a good thing; sometimes, you have to have a kind of non-flow mode where you can change the context or the level of thinking".

4.3 False Sense of Productivity

Six developers described experiencing a false sense of productivity phenomenon, which they characterized as *movement without momentum*—a state in which a great deal appears to be happening, and one feels productive and efficient, yet meaningful progress is lacking. Hence, *feeling* productive does not equal *being* productive. For example, developers may be engaged in tasks that do not contribute meaningfully to the development project or are misaligned with project goals. As senior software engineer 1 reflected:

> *And then, after many hours, I get the feeling that, okay, I haven't been able to keep track of what I've been doing. I get a lot done, but whether it's a good result is another matter.*

A second element of this phenomenon involved *feedback gaps*—delays or deficiencies in receiving feedback that could help guide or adjust work. Although software development often features efficient feedback loops through development tools, rapid development feedback can create a false sense of progress. Developers may build functioning features that, while technically correct, do not actually solve the underlying problem. Several interviewees questioned whether, during prolonged flow states, they were truly advancing the project or merely keeping themselves busy with tasks they found enjoyable. For instance, software developer 1 said:

> *It [flow] usually makes you feel like you've accomplished things. I'm just saying that I feel like that, and not like I've achieved things necessarily, because, well, there have been times when I've coded something in the flow and later realized that it was wrong and didn't make any sense, so I've thrown it all in the trash. So, yes, it makes me feel that I was efficient, at least. At least there was movement in some direction—not necessarily that the movement would always have gone in the right direction.*

Another problem associated with having movement without direction involves questioning oneself often enough. When one cannot step outside a situation (i.e., too immersed in the flow), one lacks feedback (or is not receptive to it), which might affect their ability to solve the problem at hand. For instance, senior software developer 1 said:

> *There are also negative things, and in fact, I've started to notice more and more that a flow state is not always a good thing… I have such a tendency to hyperfocus. Sometimes, in my own mind, I'm doing really good work, and then I return to it the next day, and it's total scheisse [shit]. You either don't remember what you did anymore or, at worst, solve the problem all wrong because you didn't question yourself enough.*

Alternatively, when one lacks feedback, no matter how amazing one finds the flow state, one's contributions to a project might be unhelpful or even insignificant, as senior software engineer 1 said:

> *And the fact that if I'm in [flow] mode myself, it feels good—when you don't notice that time flies, and so on. But it's just that in those situations, I don't get feedback on what I'm doing at a certain cycle. The fact that I jump up in the morning and start to write, like, "Wow, I feel like I can just code all day long", and then at the end of the day, someone says that you've been coding shit all day… It's easier [to get feedback] if you have a colleague with whom you're working to see if we're going in the right direction and if we're learning something from the problem at the moment or not. Or are we just coding something into the ether and hoping that maybe it will work for someone someday?*

The reason that interviewees perceived movement without momentum as problematic is because of its impact on project outcomes (as opposed to individuals' personal experience *in* the flow state, which might have been positive). When one does not receive any immediate feedback and one gets so immersed in the flow state that one forgets to step outside the situation and evaluate the big picture often enough (i.e., What are we doing here? What is the problem we are trying to solve?), one risks proceeding in the wrong direction, which can happen quite quickly in the flow state (especially when solving complex or "wicked problems"). Furthermore, interviewees noted that the software industry continues to go in such a direction that software developers need to solve technically complex problems and, thus, need constant evaluation and correction. As developer 1 said:

> *You have such a complicated issue at hand that you don't know if you're making progress in the right direction, and then when it progresses, you always have to quickly come back to thinking about the big picture… These wicked problems come up that are so complicated… It's such a complex problem that you have to keep coming back to the drawing board again and again, so the flow mode can be bad in the sense that you might go too far in a direction where you're not supposed to go. The same feeling often comes in these complex computer problems, which easily and quickly go in a direction in which they were not originally intended to go, and then this flow can even be harmful at its worst.*

4.4 Depleted Cognitive Resources

The more challenging a task, the more cognitive resources it requires and the more load it places on the brain. Software developers' work often involves solving complex problems, which can induce flow but can result in long, intense sessions that deplete their cognitive resources and, thus, reduce attentional performance. As software developer 2 said:

> *It [flow] took a great deal of mental resources. After that, I took a week off and recovered from it. It's a great feeling, but it's also tough.*

The developer described this experience as feeling "one with the code" as though he connected directly to the computer, while the physical world's constraints slowed him down. This flow experience consumed a great deal of his mental capacity. Here, we discuss instances of depleted cognitive resources resulting from flow. However, with this and the following categories (physical strain and adverse emotional responses), combined effects from both intense or prolonged flow and other factors may occur, such as the cognitive load that a task itself imposes. Here, the eight developers who discussed depleted cognitive resources referred specifically to the additional load caused by being in the flow state.

The challenge with knowing one's limits when working in an intense and long flow is that it can be hard to know when the good flow turns into something that strains one's mental resources so much that, after it, one feels like one has used one's resources in advance. If one does not register the situation in time while in the flow state, one might have pushed one's limits a little too much. As team leader 1 said:

> *When there have been some terribly long flow states, one thing I've noticed is that I'm really exhausted after them. Even though during the work, it doesn't feel exhausting, the next day might be like I can't get anything done; it's a bit like I used up all my cognitive resources beforehand.*

Another issue that a developer raised was related to how she felt others might not fully understand how mentally demanding high-intensity work that involves intense IT use could be, particularly in terms of the cognitive effort required:

> *And it's so hard, especially for people who don't understand what this job is like, to understand that sitting here next to the computer is very hard when you're really, really coding—when it feels like even your brain cells are so exhausted afterward that you can't think about anything. In a way, it [flow] really exerts everything out of your brain, and then you're completely finished.* (Scrum master 2)

4.5 Physical Strain

Whereas depleted cognitive resources constitute mental strain, ten developers noted the physical strain they experienced (e.g., feeling physically exhausted, tired, fatigued, and strained after a flow session). Moreover, they raised concerns about work ergonomics and even felt physical pain from working for hours and hours in the flow state (e.g., back pain). Of course, mental and physical strain share a link, and many developers described flow as being both mentally and physically straining. For instance, developer 1 said:

> *You notice that you might be quite tired afterward, after a session like that—mentally, and physically, too—so it does take energy.*

Furthermore, some described being extremely tired and needing to take time off after intense work sprints in flow. For example, software developer 2 said:

> *It was just the terrible tiredness that followed. It felt like I had given everything I had.*

Accumulating physical strain can also result in prolonged overexertion and insufficient recovery from intense flow sessions at work. Insufficient recovery could start to accumulate from pursuing or drifting into flow experiences at times that did not sync with other duties (e.g., office hours or family life).

Note that the interviewees discussed physical strain related to *flow experiences*. While working for extended periods, even outside of flow, can result in physical strain, the instances discussed here specifically relate to the developers' flow states. The interviewees felt that the strain occurred *due to* the flow state, which drove them to the point where they stopped registering the passage of time, bodily signals, or work ergonomics. The developers reflected that, had they not been in this flow state, they would have taken more breaks to get up from their desks (e.g., to get water or a snack) or would not have had such long or intense work sessions in the first place. The interviewees discussed how software development is inherently intense, and, due to their intrinsic interest in and enjoyment they derived from coding, they risked overexerting themselves if they experienced long, intense flow states too frequently. Flow can also be risky for certain personalities. For example, if one is conscientious or has a tendency toward perfectionism or workaholism, one might find it hard to draw the line between "healthy" flow and pushing oneself too much (not letting oneself get enough recovery). As scrum master 2 said:

> *I have a typical workaholic nature. I specifically enjoy it when I* really *get to work and get something done without interruptions [in flow]… And, well, I do have a history of pushing myself over and beyond and a bit more. I'm not very good at recognizing that myself at all.*

We discuss dependence on flow in Section 4.7.

4.6 Adverse Emotional Responses

Eight developers reported negative emotions, such as being easily irritated after flow. Also, interruptions while mid-flow (e.g., a colleague tapping one on one's shoulder or messages that disrupted the flow state) could agitate the developers. Some interviewees pointed out that (unpredictable) obstacles in the development work could also cause irritability, frustration, or anger after flow; when things worked out in the flow state, it was all well and good, but when they did not, it would affect their mood. Software developer 5 said:

> *I mean, as I said, I get really happy about successes in the flow. Then I talk and laugh and stuff, but if things don't work out, I'm pretty much sulking, and I don't feel like talking to anyone… It affects my mood. It affects my mood a lot*.

For some interviewees, being in a bad mood or feeling cross after the flow state made them want to withdraw from any social interaction, while others pondered that they should find some kind of constructive way to work off feeling like they had a shortened fuse and that everything annoyed them after flow. Scrum master 2 said:

> *In retrospect, it's like my head is messed up, and my fuse has gotten a lot shorter. My fuse is getting shorter, and I'm really cross, but then when it's done, it's… done. So, there are not necessarily positive feelings afterward, and you need to do something to work that feeling off somehow.*

The interviewees also expressed feeling guilt after being isolated and immersed in the flow state—for example, about prioritizing their own flow state over their teammates' needs (e.g., more junior colleagues asking questions and requiring support with something they needed help with) or simply about forgetting everything else around them after they entered a deep flow state. The guilt related to not being able to control themselves or to not being present for others and, thus, also had links to other outcomes (e.g., impaired self-regulation). Software developer 1 said:

> *Of course, flow also means that I'm aware that I don't check my chat, email, or anything else. That's because I'm prioritizing my work, and I need to concentrate, so when the [flow] time is over, and especially if you're so immersed in it that you spend a really long time [in the flow], you might*

> *get a little hangover—like, oh, there are five persons in the chat asking for help, and I've just been in my own bubble. So, that kind of hangover can come from the fact that I have neglected my other work duties.*

Finally, the adverse emotional responses that the interviewees reported included insecurity. As we point out above, one might become stuck in the flow state because one wants to make first-class code and feels that such code should result from a great flow experience. However, such thinking can easily lead to the "perfectionism trap" where experiencing great flow paradoxically results in one's becoming overly critical of one's own work, which can ultimately cause one to feel insecure about its quality. Senior software engineer 1 said:

> *And I have a regretful tendency toward perfectionism in the sense that if I do things for a long time [in flow], I get the feeling that, well, does this solve the problem, is this good code, and so on? I go into a bubble, and I wonder if I dare to show this. I have spent eight hours making this. This has to be good, or others will laugh at me or something. It is not a healthy way of thinking.*

The senior software engineer added that such negative feelings indicate that the flow state went too far and was no longer healthy. Thus, even though the flow experience itself may feel good, one might become insecure and fear that the outcome should have been something out of the ordinary, while worrying it is not.

4.7 Dependence on Flow

We also found that one can also become addicted to the feeling and dependent on the flow state. In particular, eight developers pointed out that, while development tasks provide a good foundation for entering flow, it can sometimes go too far, such as when the flow becomes so exceptional that nothing else compares to it, which can lead one to try sustain it to a harmful degree or even attempt to force it (despite the fact that one cannot initiate the flow state simply by deciding to enter it). In these situations, one does not want to stop because the feeling of being in the flow state is so enjoyable. For example, after an amazing flow experience, "normal" programming work can feel empty or underwhelming as software developer 2 said:

> *But now, after that [experience], this kind of basic work—basic programming where you're not in the flow—feels really poor. The feeling was so great when it was crystal clear what we were doing*.

Although development tasks, such as coding, can be addictive even without flow, the developers felt that their dependence on flow arose from the experience itself: the crystal-clear vision and exhilaration of being in an exceptional flow state made their work even more addictive. Many developers entered the profession because they had always enjoyed using different technologies and programming. Interestingly, several interviewees also suggested that having an addictive personality could partly explain this phenomenon. This personality trait might influence why someone chooses the profession in the first place or why they pursue the highs of flow to an unhealthy degree. Software developer 6 shared how she became hooked on the feeling while learning to code and began experiencing flow more frequently:

> *You get hooked on it. Coding is like you have to try something many times. You can't give up after a couple of times and be like, 'Maybe I can't do this,' but you just have to keep banging your head against the wall until it works. It's very unique. And then that feeling when you solve it, when you've been banging your head against the wall for so long, is really euphoric. That's what keeps me in this profession, when you've been banging your head against the wall for so long—like, why can't I get this to work? And when you finally do, it's really addictive.*

This quote illustrates how efficient feedback loops, from a user perspective, can lead developers to become dependent on the flow they experience during software development. Another developer, scrum master 2, speculated whether being inherently prone to addiction might have drawn her to the profession in the first place:

> *I've been wondering if it's the case that many, or probably quite a lot of, coders are, to some extent, addicted people—the kind that gets easily hooked because, in coding, the reward comes pretty quickly when things work. And then, the brain gets used to it and gets a kick out of it. It's difficult because it starts to get dangerous then. So that's how it's always been for me, yes. And when we are in the flow, and we are doing test-driven development, it's constantly, 'Yes, it works' or 'Oh no, I broke it all up.' But even so, we usually get pretty quickly to the situation that 'Yes, it works again.' Then, you kind of form an addiction to it.*

Alternatively, one may feel such a strong drive to fix things or solve problems that it becomes an obsession, which can lead to compulsive behavior and an unhealthy pursuit of deep flow states. Developer 1 stated:

> *How should I put it? Enjoyment is a bit of a funny word… It's maybe more, well, not an obsession, but it's more in that direction than enjoyment… There is a drive to do, but it does not necessarily feel enjoyable, like being on a beach drinking a piña colada, but more like… The world of experience is perhaps kind of obsessive.*

In some cases, this drive shifts the focus from solving problems to simply trying to achieve flow. Some developers became preoccupied with not being in flow, which created a cycle of worry that further hindered their ability to experience flow. This cycle often led them to neglect their well-being as being in flow became their sole focus (see also Section 4.8). Paradoxically, the mindset resulted in their attempts to force flow even though they knew that they could not deliberately initiate it as application specialist 1 said:

> *It's a bit tricky because the flow can occur once a month, once every six months, or once a year, and the length of it can vary a lot in terms of how long it lasts. Sometimes, it can even last a week or two at best, so you can get into that flow state every day. But when you try to start to force it, and you start to feel anxious about why you have not gotten into flow today, it may lead to you forcing it, like you're not seeing your friends, don't visit your parents, and skip hobbies because you're like, I have to get these things done, and I have to get the flow state back. And then, when it's, like, being forced, of course, it doesn't happen, and it can take a long time before you awaken and realize that you need a break from this whole thing.*

4.8 Work–Life Conflict

Finally, nine interviewees found that having trouble with being excessively immersed in the flow state and becoming dependent on these experiences hurt their work–life balance. However, the difference between maladaptive immersion and difficulties in detaching from work thoughts is that, in the former, one focuses on the *action at the moment* (e.g., having trouble switching context), whereas, in the latter, one ruminates on *thoughts after work*. For example, a frontend developer said:

> *First, it's very hard for me to get into [flow]. And then it's hard to get out. If I don't finish my work, I really don't want to go out if someone asks me to go out. I feel like I haven't finished my work yet; I don't want to get out of that concentration mode. Even if they drag me out, my brain will still stay on the issues I am working on… Sometimes, I can't get myself out of the work. So, then, what happens? I work all day. Like, even at night—even when I'm not working—my brain is thinking about how to fix this and how that works.*

Further, the interviewees reported dependence on flow and trying to sustain it to have led to work–life conflict in the form of neglecting social relationships, such as seeing one's family or friends. Some found it hard to break out from the flow state once really absorbed in it, which could lead them to work overtime despite the workday being over and having family waiting at home. A senior data engineer said:

> *You just focus on the work. How shall I put it? I guess some people would say it feels good. The work absorbs you, and sometimes, it can be a bit hard to break [the flow] when it's on… In fact, last Friday, my kids came home, and I just kept going. So, you go overtime you just can't stop.*

Poor work–life balance also resulted from flow states that occurred at inopportune moments (i.e., moments that did not sync with family life). As loud architect 2 pointed out, he would experience flow in the nighttime, which lacked compatibility with either being a parent of small children or office hours.

Although good flow is a nice thing to experience, the more senior developers brought up a valid point that everything in moderation is good and questioned whether they should strive for these trance-like flow states that last for hours on end or whether it would be more sustainable to work less but more efficiently. For example, entrepreneur 1 said:

> *Of course, you might ask yourself whether it's tiring in the long run—whether I'd still be able to do 10-hour days anymore—or if you really do work for those 10 hours. The fact is that as I get older, I feel that I want to spend less time doing work but do it more efficiently*.

In conclusion, many developers stated that one needs to intentionally take care of work–life balance because, sometimes, one becomes so immersed in the flow state that one does not want to stop at the end of the workday.

We summarize the adverse outcomes of flow that we identified in Table 3 and organize them based on their immediacy. Thus, we present short-term effects first followed by long-term effects. Generally, one can consider maladaptive immersion, impaired self-regulation, a false sense of productivity, depleted cognitive resources, physical strain, and adverse emotional responses as short-term effects and dependence on the flow state and work-life conflict as more long-term effects. However, one cannot always easily distinguish between short-term and long-term effects as some outcomes, such as physical strain, can have both immediate and lasting impacts.

Table 3. Adverse Outcomes of Flow in Software Developers' Work

Adverse outcome	Manifestation	Description
Maladaptive immersion	Over-engagement	Perception of being too immersed in a flow state; losing track of time and the environment; intense focus on programming such that one does not feel or register anything else.
Impaired self-regulation	Lack of control	Getting stuck in the flow state; having trouble moving on to the next activity (either to the next work task (e.g., switching from coding to a meeting) or from work to leisure).
False sense of productivity	Movement without momentum	Movement but no forward momentum, one feels efficient and/or productive and things are happening but not necessarily the right things (e.g., coding for hours something that ends up not being important).
	Feedback gaps	Lack of timely feedback on one's actions from others and/or lack of self-reflection hinders one's ability to adjust the direction in which one takes consequent actions if too deep in the flow state while coding.
Depleted cognitive resources	Reduced attentional performance	Coding in a flow state can strain the brain; attentional resources and cognitive performance decline after.
Physical strain	Fatigue	Feeling very tired, strained, and fatigued after a flow state; in the long run, the risk of overexerting oneself by insufficient recovery.
	Pain	Pain (e.g., back pain) resulting from being deeply immersed in coding for an extensive period and disregarding ergonomics.
Adverse emotional responses	Anger, irritability, frustration	Feeling irritated/frustrated/angry/agitated after an intense flow state (e.g., having a "shortened fuse").
	Guilt, shame	Feeling guilty after being immersed in a personal "flow bubble" and having neglected other work duties or role-related obligations (e.g., to be reachable and help other developers).
	Insecurity	Doubting the quality of one's work (e.g., worrying if one produces sufficiently good code while in the flow state).
Dependence on flow	Addictive quality of flow	Coding provides immediate feedback and fast rewards when succeeding in tasks, which can become addictive and can create a dependence on the flow state.
	Obsessive behavior	Obsessing about sustaining flow or trying to force it to a harmful degree because, after an amazing flow experience, normal programming work in a non-flow state feels underwhelming and "empty".
Work–life conflict	Flow rumination	Impaired work–life balance due to increased flow rumination outside of work (i.e., ruminating on the problems that one worked on in the flow state and difficulty detaching from them).
	Compromised social relationships	Skipping important aspects of personal well-being, such as seeing family or friends, due to being immersed in flow.

The interviewees linked some adverse outcomes (e.g., maladaptive immersion, false sense of productivity, and dependence on flow) more closely to the intensive IT use required in software development and other outcomes more to the nature of working in a knowledge-intensive profession that requires creative problem-solving. In the software industry, becoming overly absorbed in a flow state can be risky. Working in this fast-paced, seemingly productive state may feel rewarding, but it does not guarantee that one prioritizes the right things or that one's progress aligns with a project's goals. Software development projects often involve high interdependency and ongoing objective evaluations, which makes excessive immersion in flow potentially counterproductive. Additionally, developers described coding as an inherently addictive activity with intense emotional highs when one makes progress and lows when one does not; the code either works or it does not, which requires persistence and focus on the task until one achieves success. Altogether, the adverse outcomes of flow that we identify in this study have the potential to strain employees momentarily and in the long run.

5 Discussion

5.1 Results

Overall, the interviewees described flow experiences that involved intense and focused concentration, an altered sense of time, and an autotelic experience (i.e., the developers experienced engaging in their work activities as rewarding and enjoyable) to be consistent with the dimensions of the flow concept (Nakamura & Csikszentmihalyi, 2002). All interviewees reported experiencing adverse outcomes along their flow experiences or encountering friction when their flow-related preferences conflicted with organizational or team needs and other work-related duties. The developers emphasized that, while flow is primarily a positive and productive state, one needs to understand these adverse effects. This finding challenges the common view that flow typically leads only to positive work-related outcomes, such as improved performance, increased work engagement, and reduced burnout (Lavigne et al., 2012; Mosing et al., 2018; Weintraub et al., 2021).

Researchers consider being deeply focused and immersed as the essence of being in flow (Csikszentmihalyi, 1990). However, our findings show that individuals' perceptions of their degree of immersion affect how they feel about the flow state. While the developers described a fair amount of temporal dissociation and focused immersion (Agarwal & Karahanna, 2000) as typical for their flow experiences, they perceived **maladaptive immersion** as over-engagement—being absorbed to the degree that it became harmful (e.g., it leads one to neglect other work duties that one needs to complete) (Trevino & Webster, 1992) or to focus on the wrong task for too long). At times, the developers' maladaptive immersion became similar to dissociation, which researchers generally consider a maladaptive state, whereas they generally consider flow an adaptive state (Ahmed & Dixon, 2023). While studies on compulsive behavior and addiction to IT use have addressed maladaptive immersion (e.g., Khang et al., 2013; Mason et al., 2022; Stavropoulos et al., 2018; Theotokis & Doukidis, 2009; Zhang et al., 2014), they have focused more on the behavioral mechanism behind why using IT becomes compulsive, even to the degree of addiction. Moreover, these studies have investigated personal IT use and absorption only in IT use. In this study, due to the nature of software developers' work tasks, IT played a key role in their flow experiences—but it did not necessarily constitute the *reason* for maladaptive immersion. Although researchers have suggested that overinvolvement (Csikszentmihalyi, 1975), even to the degree that one neglects other work duties (Trevino & Webster, 1992), leads to undesirable outcomes, we find it interesting that empirical studies in the work context have not previously investigated this issue.

Further, although control (among other characteristics) has been characterized as flow in the HCI context (Webster et al., 1993), the interviewed software developers reported the opposite: **impaired self-regulation** (manifested as lack of control) while immersed in the flow state. For example, when in flow, they struggled with other work duties (Trevino & Webster, 1992), such as attending meetings, because they found it difficult to disengage from the flow state. This difficulty in disengaging was closely linked to maladaptive immersion and over-engagement. Although a sense of control (Csikszentmihalyi, 1975, 1990; Nakamura & Csikszentmihalyi, 2002) and one's belief that one can exert control over the situation (Wilson, 2016) often characterize flow, in these states of over-engagement in flow, the developers reported the opposite: a lack of control. The interviewees felt that, when flow "went overboard" they experienced impaired self-regulation and could no longer control the situation anymore despite feeling great and productive while in flow. This paradox resembles the productivity paradox and the paradox of control and chaos (Balijepally et al., 2009; Dehnert & Friedrich, 2022). While researchers consider a sense of control as a characteristic of being in flow, they have explored the metacognitions of flow states, such as self-regulation, less frequently (Wilson, 2016). We view this finding as particularly interesting because, while challenges and work pressures (e.g., deadlines) often hinder flow (e.g., Ritonummi et al., 2024), our findings suggest that, once in the flow state, developers may become oblivious to other obligations, which can lead to distress from a lack of self-regulation.

We also found that the flow state can create a **false sense of productivity**, and the quick progress and lack of outside feedback in an absorbed state can quickly take things in the wrong direction. We identified movement without momentum manifestation as a significant drawback of flow. Although the industry acknowledges the problem of a false sense of productivity when coding in flow (e.g., Martin, 2011), this finding represents a novel one in IS research. Further, researchers generally consider flow a useful state in the work context (Wilson, 2016). However, while flow often constitutes a useful, productive, and enjoyable state, individuals might sometimes perceive it as useful in instances where it is not (cf. the productivity paradox of thinking that programming in pairs will exceed the performance of its best member because “two heads are better than one” (Balijepally et al., 2009)). Industry practitioners’ views also support this finding (Martin, 2011; Zuill, 2019). Because software development involves many moving parts and projects require collaboration between all involved parties, very intense and long flow sessions might not always be optimal. Our interviewees discussed this topic from both an individual (e.g., not wanting to get too absorbed into the flow because they found it personally disadvantageous) and other perspectives (e.g., how the flow will affect others’ work, does the project stay on track, etc.) The developers often had the desire to get into flow while working, but, at the same time, thought about the common good, and some had concluded that, actually, it was not necessarily optimal to always enter the flow state. Although previous research has not addressed this specific productivity paradox problem, we can see some analogy between having movement without momentum, which can skew how an individual perceives a situation (having a false sense of productivity), and impaired risk awareness in the flow (Schüler & Nakamura, 2013; Schüler & Pfenninger, 2011).

Moreover, most interviewees agreed that, although flow is an amazing thing to achieve and one feels great while absorbed in it, it often takes a mental and physical toll. The findings corroborate that the cognitive load experienced during flow can result in **depleted cognitive resources** that manifest in reduced attentional performance afterwards (Keller et al., 2011; Liu et al., 2022). The developers described situations where a development task’s cognitive demands (Mueller & Benlian, 2022) while in flow exceeded their capacity, which disrupted the flow state and reduced their attentional resources and performance for the following moments (e.g., upcoming workdays). Also, the cognitive load associated with their work (Gonçales et al., 2021) may contribute to reduced attentional performance following intense work sessions. Surprisingly, previous work has addressed physical and mental strain from flow mainly in the non-work IT-use context (Keller et al., 2011). However, Liu et al. (2022) discussed compromised attentional performance after flow (depleted cognitive depletion resources), but researchers have addressed the physical strain associated with flow experiences only with respect to injuries from playing an instrument professionally (Guptill, 2012). Interestingly, while individuals typically associate flow with well-being as a bodily sensation, some professional groups (e.g., musicians) reportedly employ strategies to disrupt their flow in order to preserve their health and continue in their profession (Guptill, 2012). The developers reported that these overly intense flow states sometimes result in **physical strain**, which includes both immediate fatigue and pain after flow in addition to prolonged exhaustion. Some even speculated whether this overexertion from intense and prolonged flow states related to severe outcomes, such as having gone through burnout at some point or several points in their career. Contrary to most flow research, which associates work-related flow with more vigor and less exhaustion (e.g., Demerouti et al., 2012), these findings highlight that flow can also increase physiological strain (Keller et al., 2011). With depleted cognitive resources and physical strain, other factors (e.g., cognitive load) may influence how strained one feels after flow. However, the developers attributed these effects to having experienced flow and noted that working in a non-flow state would not have led to severe outcomes.

By the same token, we identified that flow could induce **adverse emotional responses** such as irritability or guilt after flow. We identified several negative states that flow can evoke after the fact. Our interviewees struggled with feeling irritated even to the degree that they became angry or cross. While researchers have generally associated positive emotions with work-related flow (e.g., Salanova et al., 2006) and adverse emotional responses with task disruptions (e.g., technology-mediated interruptions) (Weinert et al., 2022), research on the adverse outcomes of flow in the work context has mostly focused on mental or physical strain or behavioral outcomes (e.g., Feng et al., 2024; Guptill, 2012; Liu et al., 2022) and, thus, overlooked the emotionally draining aspects of flow. Although extensive research has examined flow experiences’ emotional (e.g., Nah et al., 2011; Sepehr & Head, 2012) and neurological sides (Adam et al., 2024; Nadj et al., 2023; Rissler et al., 2018; Van der Linden et al., 2021), they have overlooked negative emotions after flow.

As for perhaps the most harmful adverse outcomes of flow on an individual’s well-being, dependence on flow (and often subsequent work–life conflict) not only affected the software developers’ work but also their free time. At times, these effects lasted for a considerable period (in comparison to, for example, adverse emotional responses, which often had immediate and short-term effects). We found an association between **dependence on flow** and addiction to coding in the flow state (some interviewees even viewed coding as synonymous with coding in a flow state). The developers reported that they sometimes behaved obsessively and tried to sustain or get into flow by any means possible. Contrary to Sepehr and Head's (2012) distinction between harmonious passion (associated with flow) and obsessive passion (associated with addiction), our findings demonstrate that both harmonious and obsessive passion can be associated with flow states. While researchers have discussed dependence and addiction in sports (Partington et al., 2009) and personal IT use (e.g., Khang et al., 2013; Mason et al., 2022; Stavropoulos et al., 2018; Theotokis & Doukidis, 2009; Zhang et al., 2014), our findings highlight that dependence and addiction can also be problematic in the work context. Although researchers have investigated compulsive use and addiction in personal IT use (e.g., Khang et al., 2013; Theotokis & Doukidis, 2009; Zhang et al., 2014), they have not discussed dependence on flow in work-related IT use. Software developers' work is highly IT-intensive, and we found that they form a dependence on the flow state in and beyond the HCI context: one can become obsessed or addicted to the technology-mediated activity (e.g., coding) or with the feeling one gets from being in flow to the degree that one forms an unhealthy habit. Although the motivations for engaging in work activities might not be intrinsically motivated and hedonic compared to surfing or online gaming, for example, we found that software developers found themselves obsessing about getting into the flow or trying to sustain the flow because "normal" work pales in comparison to being in a wonderful flow state. Becoming dependent on this narrow idea of what brings one joy can be dangerous (Csikszentmihalyi, 1999) not only because it can lead one to miss opportunities for learning and enjoyment but also because dependence can cause problems with work–life balance.

Finally, the negative effects of flow in software developers' work included **work–life conflict**. Although researchers often consider flow a factor that can prevent negative phenomena related to problematic work–life balance (which can lead to, for example, exhaustion and burnout) (Lavigne et al., 2012; Mosing et al., 2018; Weintraub et al., 2021), we found it to have a potentially deteriorating effect on software developers' work–life balance. While ruminating on flow experiences after work is not a harmful thing per se and the rumination itself has a positive valence (Feng et al., 2024), researchers have found it to cause negative outcomes, such as work–family conflict (Feng et al., 2024; Ramsey & Lorenz, 2020). We also identified that these conflicts could arise from work-related flow rumination, social relationship neglection, or simply timing conflicts between flow sessions and family duties. Previous studies have identified impaired work–life balance and flow rumination (Feng et al., 2024; Ramsey & Lorenz, 2020), but our findings demonstrate that, in addition to increasing the difficulties of detaching from work (partially mediated by dependence on flow states), flow states could even compromise well-being by causing individuals to neglect important aspects of their personal life, such as their social life outside beyond work.

5.2 Research Contributions

This study makes two contributions to our knowledge on work-related flow and its effects on software developers' affect, behavior, cognition, and performance (Rissler et al., 2017). First, we contribute to the emerging stream of flow research that challenges the assumption that flow in the work context is exclusively advantageous (e.g., Liu et al., 2022; Ramsey & Lorenz, 2020) by thoroughly explaining the adverse outcomes of flow. While earlier research has identified individual non-ideal flow outcomes, such as work–family conflict (e.g., Ramsey & Lorenz, 2020), this exploratory study marks a step toward providing a comprehensive understanding of the negative aspects of flow. We identified four novel adverse outcomes of flow in software developers' work: maladaptive immersion, impaired self-regulation, a false sense of productivity, and adverse emotional responses. While researchers have mostly associated flow with positive emotions, creativity, and job satisfaction (e.g., Salanova et al., 2006; Zubair & Kamal, 2015), the findings highlight that flow can be a "too much of a good thing" phenomenon. Research has generally found that flow has positive consequences, such as vigor, self-efficacy, knowledge sharing, inter-employee helping, better performance, and satisfaction (Demerouti et al., 2012; Knierim et al., 2019; Lin et al., 2009; Salanova et al., 2014) and that it can drive software developers' creativity, enjoyment, motivation, and productivity (Brown et al., 2023a, 2023b; Calais & Franzini, 2023; Pratt et al., 2016; Ritonummi et al., 2023; Zubair & Kamal, 2015). Furthermore, researchers have generally considered absorption in work tasks a positive aspect of flow because it indicates that the activity is intrinsically motivating and enjoyable (Bakker, 2008; Rodríguez-Sánchez et al., 2011; Salanova et al., 2006). However, our findings indicate a threshold after which being immersed can turn disadvantageous, such as when the developers' perceived control over the immersion diminished (e.g., inability to change contexts or stop the flow) and the positive immersion turned into an unpleasant "hyperfocus", which suggests that the extent to which one becomes immersed in flow represents a continuum and that individuals' perceptions about it affects whether they interpret it as appropriate or maladaptive at a given moment. We summarize this study's contributions to flow theory in Table 4.

Table 4. Summary of Research Contributions

Identified adverse outcomes of flow (novel findings*)	Contributions to flow theory (flow dimensions)
Maladaptive immersion*	We found an association between **maladaptive immersion** and flow's most prominent dimension: intense concentration (Nakamura & Csikszentmihalyi, 2002). This finding offers a new perspective on flow by illustrating how, beyond a certain point, high focus can become counterproductive and lead to an undesirably deep state of immersion.
Depleted cognitive resources	Moreover, prolonged or overly intense flow sessions in a state of intense concentration can take a toll on one's physical and mental well-being. Research has linked such experiences to **depleted cognitive resources** (Keller et al., 2011; Liu et al., 2022), **physical strain** (Guptill, 2012; Keller et al., 2011), and **adverse emotional responses**—especially when one performs complex development tasks (Mueller & Benlian, 2022). These findings underscore cognitively demanding work's draining effects.
Physical strain	
Adverse emotional responses*	
Impaired self-regulation*	Although researchers have typically associated flow with a sense of control (Nakamura & Csikszentmihalyi, 2002), it can reach a point where one perceives a decrease in their ability to self-regulate, which results in **impaired self-regulation**. Paradoxically, while individuals may feel in control of a task, their behavioral control can simultaneously diminish (Dehnert & Friedrich, 2022). As a result, they may struggle to disengage from the flow state and, thus, neglect other work duties.
False sense of productivity*	The altered sense of time and loss of time awareness during flow (Nakamura & Csikszentmihalyi, 2002) can create a **false sense of productivity** by giving people the illusion that they have made progress even though they used their time ineffectively. This phenomenon resembles the productivity paradox in which one perceives a process or behavior to enhance productivity even though it may have the opposite effect in reality (Balijepally et al., 2009; Cachon, 2012). Additionally, the merging of action and awareness, loss of reflective self-consciousness (i.e., awareness of oneself), and autotelic experience (Nakamura & Csikszentmihalyi, 2002) can further contribute to this false sense of productivity. These flow characteristics reduce conscious, reflective thought and cause individuals to engage in tasks primarily due to their intrinsically rewarding nature rather than because they yield productive outcomes.
Dependence on flow	Additionally, the autotelic experience that flow provides can become so rewarding that individuals develop a **dependence on the flow state** itself (Partington et al., 2009; Wu et al., 2013). While intrinsic task enjoyment is a defining flow characteristic (Nakamura & Csikszentmihalyi, 2002), this research highlights how pursuing such experiences can lead to problematic outcomes.
Work–life conflict	Ultimately, excessive flow and its adverse effects (e.g., altered sense of time together with **maladaptive immersion**) can contribute to **work–life conflict** (Feng et al., 2024; Ramsey & Lorenz, 2020). The difficulty of detaching from work and prioritizing social relationships underscores the potential downsides of excessive or uncontrolled flow experiences.

Second, the adverse outcomes we identify in this study enhance the extent to which we understand the dimensions of flow (Nakamura & Csikszentmihalyi, 2002) and the overall effects that can emerge from working in a cognitively demanding profession (Mattern et al., 2024) and, thus, provide further insights into work-related flow (e.g., Adam et al., 2024; Knierim et al., 2019; Nadj et al., 2023; Salanova et al., 2006). Moreover, our findings contribute to explaining the paradoxes of knowledge work (Balijepally et al., 2009; Dehnert & Friedrich, 2022; Mueller & Benlian, 2022) via explaining how these adverse outcomes manifest in software developers' work. By identifying the eight adverse outcomes of flow with contextual, nuanced information on how they manifest in software developers' work, this study contributes to explaining work-related flow and its paradoxical outcomes in knowledge work better. Moreover, our findings highlight how the intense IT use played a role in their adverse experiences. In their work, software developers often experienced flow in various coding activities (when they fulfilled its antecedents) (Ritonummi et al., 2023), and they actively sought to experience it (Meyer et al., 2014); however, we found that they sometimes experienced flow with adverse consequences, such as going in the wrong direction too fast or remaining too long in the flow, which could harm their efforts to advance development projects. For example, these adverse consequences could occur when developers became immersed in an efficient feedback loop from an IDE but did not receive external feedback (from outside the interaction) on whether they achieved the task goals. Our findings also highlight how coding's intensity and immersiveness can lead to depleted cognitive resources and even dependence on flow. Furthermore, researchers generally understand that flow benefits work performance (Knierim et al., 2019; Quinn, 2005), that interruptions are bad, and that working in a focused flow is desirable (e.g., Abad et al., 2018; Chen et al., 2022; Ritonummi et al., 2024; Züger et al., 2017). However, knowledge sharing and inter-employee helping play an important role in organizations (Lin et al., 2009). While Lin et al. (2009) found that flow positively influences knowledge sharing and inter-employee helping, we found that flow can, at worst, have an inhibiting effect on knowledge sharing and collaboration because, when an individual becomes maladaptively immersed in the flow state, their actions can conflict with a team's or organization's needs. Knowledge sharing exemplifies a common collaborative goal that can experience negative effects if an individual does not contribute to it but instead prioritizes their personal preferences (or becomes too immersed to notice others' needs). We found instances from both perspectives in our data. While one developer group preferred working alone uninterrupted in the flow but had received negative feedback about not being a "team player", and the latter group wanted to collaborate more and work jointly at the office but had not managed to entice their colleagues to join them.

In conclusion, our findings demonstrate that, while flow mostly constitutes a positive phenomenon with desirable consequences in software developers' work, such as increased motivation, creativity, and happiness, it also has an unfavorable side. Thus, an optimal experience is not necessarily positive in an absolute sense and can result in various adverse outcomes (Csikszentmihalyi, 1990; Zimanyi & Schüler, 2021). Rather than enforcing beliefs about only the positive consequences of flow, a more fruitful perspective would involve increasing awareness about its negative consequences, which can help foster workers' confidence to self-regulate flow (Wilson & Moneta, 2023). However, if developers experience less flow due to interruptions and other barriers (e.g., Abad et al., 2018) or because they want to avoid the negative outcomes of flow, they could miss out on other benefits of flow for their work, such as increased productivity (e.g., Brown et al., 2023a) and motivation (e.g., Kuusinen et al., 2016).

5.3 Practical Implications

Our findings suggest that software professionals and managers should become better at recognizing when flow can be a useful thing and when it can turn against itself. First, they suggest that employees need to better recognize the pitfalls of getting too absorbed in flow and identifying when it may not be the best time to immerse themselves too much in performing a task (e.g., when one's personal flow state at a given moment may not support a project's overall goals or one's team's needs or when flow causes negative affective responses that affect either oneself or one's colleagues). This suggestion applies especially to maladaptive immersion, impaired self-regulation, a false sense of productivity, and adverse emotional responses that can result from intense flow states. Possible mitigation strategies that the interviewees discussed included writing notes about a task's goals before starting it and drifting into the flow or writing notes about what one worked on in the flow so that, when the flow ends, one can more easily come back to the task later on and not have to stress about forgetting what one was doing (because, in the flow state, one often feels such momentum that one cannot easily keep track of problems that one has solved and how one did so). Also, forcing oneself to step outside of the situation and evaluate (alone or with the team) what one has accomplished during flow can be a useful technique to keep the flow healthy so that one does not drift into doing "busy work" (a false sense of productivity) and to force oneself to stop at the end of the time allotted to work so that the flow does not overextend.

Second, software developers need to take care of their physical and mental resources and learn to recognize in advance when they may go overboard and get carried away by either trying to sustain the flow or chase the flow state to an unhealthy degree. This challenge maps onto both the depleted cognitive resources and physical strain in addition to dependence on flow and work–life conflict. Software developers need to make sure they have enough time to recover from flow and that their work–life balance stays healthy, which can be quite challenging, especially in software development, which often attracts people with a deep passion for their work (e.g., programming). As our findings demonstrate, not taking breaks either in the high momentum "coding frenzy" or because one fears that one will not be able to reenter flow can actually have the opposite effect. Forgetting or refusing to take breaks in the flow state can have adverse outcomes on one's cognitive performance and physical and mental well-being. Thus, mindfully taking short breaks at work, even mid-flow, can help one sustain one's physical and mental well-being and work performance. Furthermore, the interviewees mentioned trying to find ways to calm the mind at work after an intense flow, having hobbies outside work, or just simply going for a walk after an intense day in the flow as useful mitigation strategies to combat the adverse outcomes of flow and look after their well-being to sustain their performance and work–life balance.

Third, managers can use the above insights as guidance to assess whether they should encourage employees to have deeply concentrated flow sessions or to discourage them because they may not provide the optimal way for software developers to reach common goals or work in the right direction. For example, managers could encourage developers to enter flow when working on modules that do not require that much team input. On the other hand, managers could also share information about healthy working practices and remind people that they should not prioritize flow over such practices. This implication maps onto all our findings. Communicating goals clearly and providing timely feedback can help developers not only reach flow but also keep projects on track. While developers may need to sometime reflect on whether they have met expectations and to consider other team members' and their organization's needs (with respect to the extent to which and for how long they should remain immersed in the flow state), we found that, most importantly, development teams should not focus on efficiency (busy work) or productivity (getting things done) but rather effectiveness (getting the right things done) (Zuill, 2019).

5.4 Limitations and Future Research

Collecting qualitative data, especially from in-depth interviews, represents a useful approach to understand a topic. However, this study's limitations include challenges related to interviews (e.g., the artificiality of the interview setup). We tried to mitigate such limitations by following Myers and Newman's (2007) guidelines for interviews. Further, studying flow itself raises limitations. First, it constitutes a highly subjective experience (Quinn, 2005), and self-reporting is subject to recall bias (Moneta, 2012). However, since we focused on understanding the interviewees' experience as they interpreted it (Myers & Newman, 2007), we deemed retrospective self-reporting the most suitable data-collection technique. To mitigate the challenges of retrospection, we gave the interviewees time to think about their answers and to provide as detailed descriptions as possible. We also defined and discussed flow with the interviewees when we began each interview in addition to informing them about it beforehand. Furthermore, more adverse outcomes or their manifestations than we identified in this study could exist. For example, more negative emotions (e.g., disappointment and distress) associated with flow could exist. However, by identifying an array of adverse outcomes of flow, this study establishes a basis for future research.

Because we focused on software developers' experiences, future research could validate our findings by studying a wider range of knowledge workers to identify the extent to which the adverse outcomes and their manifestations resemble or differ from the ones we identified in different professions among knowledge workers. Another promising research avenue would involve examining the flow experience at the workgroup level (e.g., Salanova et al., 2014) and explaining how collective flow manifests in knowledge work (in particular, if experiencing collective flow has adverse consequences as individual flow does). Furthermore, future research could examine how to mitigate the adverse outcomes of flow and what practical implications such mitigation strategies would have for software developers' work. The mitigation suggestions that we discuss above likely exemplify a wider variety of coping and mitigation strategies. Moreover, since we conducted a qualitative study, we did not statistically compare junior versus senior software developers' experiences, but doing so could represent an interesting avenue to pursue in future research to see if early-career developers have experiences that differ from more established developers. Similarly, research could further develop the distinction between the short- and long-term adverse effects of flow. Finally, future research could explore the prevalence of adverse outcomes among developers, such as how frequently they occur and which outcomes are most common.

6 Conclusion

In this paper, we used a qualitative approach to identify different adverse outcomes that flow experiences can have in software developers' work. Our findings detail the characteristics of the adverse outcomes of flow and how they manifest. The manifestations provide contextual and nuanced information on the challenges related to how software developers experience flow in the IT-use intensive work. Furthermore, our results highlight how these outcomes can have short-term and long-term impacts at the personal, team, and organizational levels. This paper contributes to research on understanding work-related flow and the unfavorable effects of flow.

7 Acknowledgments

This research was supported by the Foundation for Economic Education, the Research Council of Finland (341359, 346479), and the Emil Aaltonen Foundation.

References

Abad, Z. S. H., Noaeen, M., Zowghi, D., Far, B. H., & Barker, K. (2018). Two sides of the same coin: Software developers' perceptions of task switching and task interruption. In *Proceedings of the International Conference on Evaluation and Assessment in Software Engineering*.

Adam, M. T. P., Bonenberger, L., Gimpel, H., & Lanzl, J. (2024). Human-centered design and evaluation of a NEUROIS tool for flow support. *Journal of the Association for Information Systems*, *25*(4), 936–961.

Agarwal, R., & Karahanna, E. (2000). Time flies when you're having fun: Cognitive absorption and beliefs about information technology usage. *MIS Quarterly, 24*(4), 665–694.

Ahmed, S., & Dixon, M. J. (2023). Instagram, depression, and dark flow—using social media as a maladaptive coping mechanism. *Journal of Addiction Research*, *7*(2), 61–69.

Aleksić, D. (2016). *Negative aspects of flow: Examining relationships between flow and unethical behavior* (doctoral dissertation). University of Ljubljana.

Andrade, F. R. H., Mizoguchi, R., & Isotani, S. (2016). The bright and dark sides of gamification. In *Proceedings of the International Conference on Intelligent Tutoring Systems*.

Bakker, A. B. (2008). The work-related flow inventory: Construction and initial validation of the WOLF. *Journal of Vocational Behavior*, *72*(3), 400–414.

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

Baltes, S., & Diehl, S. (2018). Towards a theory of software development expertise. In *Proceedings the ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*.

Barberis, N., Cannavò, M., Costa, S., & Cuzzocrea, F. (2022). Problematic behaviours and flow experiences during screen-based activities as opposite outcomes of the dual process of passion and basic needs. *Behaviour and Information Technology*, *41*(14), 3110–3123.

Barta, S., Gurrea, R., & Flavián, C. (2022). The role of flow consciousness in consumer regret. *Internet Research, 32*(3), 875–896.

Beecham, S., Baddoo, N., Hall, T., Robinson, H., & Sharp, H. (2008). Motivation in software engineering: A systematic literature review. *Information and Software Technology*, *50*(9–10), 860–878.

Boudreau, P., Mackenzie, S. H., & Hodge, K. (2020). Flow states in adventure recreation: A systematic review and thematic synthesis. *Psychology of Sport and Exercise*, *46*(2020), 101611.

Brailovskaia, J., Bierhoff, H. W., Rohmann, E., Raeder, F., & Margraf, J. (2020). The relationship between narcissism, intensity of Facebook use, Facebook flow and Facebook addiction. *Addictive Behaviors Reports*, *11*(2020).

Brailovskaia, J., & Teichert, T. (2020). "I like it" and "I need it": Relationship between implicit associations, flow, and addictive social media use. *Computers in Human Behavior*, *113*, 1-7.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology, 3*(2), 77–101.

Braun, V., & Clarke, V. (2013). *Successful qualitative research: A practical guide for beginners*. Sage.

Brown, A., D'Angelo, S., Holtz, B., Jaspan, C., & Green, C. (2023a). Using logs data to identify when software engineers experience flow or focused work. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*.

Brown, A., Chang, A., Holtz, B., D'Angelo, S. (2023b). Developer productivity for humans, part 6: Measuring flow, focus, and friction for developers. *IEEE Software*, *40*(6), 16–21.

Cachon, G. P. (2012). What is interesting in operations management? *Manufacturing and Service Operations Management*, *14*(2), 166–169.

Calais, P., & Franzini, L. (2023). Test-driven development benefits beyond design quality: Flow state and developer experience. In *Proceedings of the International Conference on Software Engineering.*

Califf, C. B., Stumpf, T. S., & Frye, J. J. (2020). Revisiting technology and flow: A call for an alternative perspective and directions for the future. In *Proceedings of the Hawaii International Conference on System Sciences*.

Cao, Y., Wang, L., Zheng, Z., & Tao, X. (2020). A tool for non-intrusive and privacy-preserving developers' programming activity data collection. In *Proceedings of the Annual Computers, Software, and Applications Conference*.

Chen, Y., Rigby, P. C., Chen, Y., Jiang, K., Dehghani, N., Huang, Q., Cottle, P., Andrews, C., Lee, N., & Nagappan, N. (2022). Workgraph: Personal focus vs. interruption for engineers at Meta. In *Proceedings of the ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*.

Cowley, U. B., Hellas, A., Ihantola, P., Leinonen, J., & Spape, M. (2022). Seeking flow from fine-grained log data. In *Proceedings of the International Conference on Software Engineering: Software Engineering Education and Training*.

Csikszentmihalyi, M. (1975). *Beyond boredom and anxiety*. Jossey-Bass.

Csikszentmihalyi, M. (1990). Flow: The psychology of optimal experience. HarperCollins.

Csikszentmihalyi, M. (1999). If we are so rich, why aren't we happy? *American Psychologist, 54*(10), 821–827.

Dehnert, M., & Friedrich, S. (2022). How to get into flow with it: Measuring the paradoxes in digital knowledge work. In *Proceedings of the European Conference on Information Systems*.

Demerouti, E., Bakker, A. B., Sonnentag, S., & Fullagar, C. (2012). Work-related flow and energy at work and at home: A study on the role of daily recovery. *Journal of Organizational Behavior*, *33*(2012), 276–295.

Dennehy, D., & Conboy, K. (2017). Going with the flow: An activity theory analysis of flow techniques in software development. *Journal of Systems and Software*, *133*(2017), 160–173.

Dixon, M. J., Stange, M., Larche, C. J., Graydon, C., Fugelsang, J. A., & Harrigan, K. A. (2018). Dark flow, depression and multiline slot machine play. *Journal of Gambling Studies*, *34*(1), 73–84.

Feng, X., Han, P., & Terpstra Tong, J. (2024). Joy at work turns to sorrow at home: The influence of flow experience on work–family conflict and a three-way interaction effect. *Applied Psychology*, *73*(2), 1–29.

Galster, M., Mitrovic, A., Malinen, S., Holland, J., & Peiris, P. (2023). Soft skills required from software professionals in New Zealand. *Information and Software Technology*, *160*, 1-14.

Ghani, J. A., Supnick, R., & Rooney, P. (1991). Experience of flow in computer-mediated and F2F groups. In *Proceedings of the International Conference on Information Systems.*

Glaser, B. G., & Strauss, A. L. (1967). *The discovery of grounded theory: Strategies for qualitative research*. Aldine Publishing Company.

Gonçales, L. J., Farias, K., & da Silva, B. C. (2021). Measuring the cognitive load of software developers: An extended systematic mapping study. *Information and Software Technology, 136*, 1-30.

Graziotin, D., Fagerholm, F., Wang, X., & Abrahamsson, P. (2018). What happens when software developers are (un)happy. *Journal of Systems and Software*, *140*, 32–47.

Greiler, M., Storey, M. A., & Noda, A. (2022). An actionable framework for understanding and improving developer experience. *IEEE Transactions on Software Engineering*, *49*(4), 1411–1425.

Guo, Y. M., & Poole, M. S. (2009). Antecedents of flow in online shopping: A test of alternative models. *Information Systems Journal, 19*(4), 369–390.

Guo, Z., Xiao, L., Van Toorn, C., Lai, Y., & Seo, C. (2016). Promoting online learners' continuance intention: An integrated flow framework. *Information and Management, 53*(2), 279–295.

Guptill, C. (2012). Injured professional musicians and the complex relationship between occupation and health. *Journal of Occupational Science, 19*(3), 258–270.

Hsu, C. L., & Lu, H. P. (2004). Why do people play on-line games? An extended TAM with social influences and flow experience. *Information and Management, 41*(7), 853–868.

Hu, E., Stavropoulos, V., Anderson, A., Scerri, M., & Collard, J. (2019). Internet gaming disorder: Feeling the flow of social games. *Addictive Behaviors Reports, 9*, 1-8.

Hu, H., Zhang, G., Yang, X., Zhang, H., Lei, L., & Wang, P. (2022). Online gaming addiction and depressive symptoms among game players of the Glory of the King in China: The mediating role of affect balance and the moderating role of flow experience. *International Journal of Mental Health and Addiction, 20*(5), 3191–3204.

Huang, L. Y., Hsieh, Y. J., & Wu, Y. C. J. (2014). Gratifications and social network service usage: The mediating role of online experience. *Information and Management, 51*(6), 774–782.

Jiang, Z., & Benbasat, I. (2004). Virtual product experience: Effects of visual and functional control of products on perceived diagnosticity and flow in electronic shopping. *Journal of Management Information Systems, 21*(3), 111–147.

Jo, H., & Baek, E. M. (2023). Exploring the dynamics of mobile app addiction: The interplay of communication, affective factors, flow, perceived enjoyment, and habit. *BMC Psychology*, *11*(1), 1–16.

Johnson, C. D., Bauer, B. C., & Singh, N. (2020). Exploring flow in the mobile interface context. *Journal of Retailing and Consumer Services, 53*, 1-7.

Kalantari, R., & Lethbridge, T. C. (2022). Preliminary results of measuring flow experience in a software modeling tool: UmpleOnline. In *Proceedings of the International Conference on Model Driven Engineering Languages and Systems.*

Kaufhold, M.-A., Reuter, C., & Ludwig, T. (2019). Flow experience in software engineering: Development and evaluation of design options for Eclipse. In *Proceedings of the European Conference on Information Systems.*

Keith, M., Anderson, G., Dean, D., Gaskin, J., Wiser, T., Gremmert, S., & Payne, K. (2016). The effects of video gaming on work group performance. In *Proceedings of the International Conference on Information Systems.*

Keller, J., & Bless, H. (2008). Flow and regulatory compatibility: An experimental approach to the flow model of intrinsic motivation. *Personality and Social Psychology Bulletin, 34*(2), 196–209.

Keller, J., Bless, H., Blomann, F., & Kleinböhl, D. (2011). Physiological aspects of flow experiences: Skills-demand-compatibility effects on heart rate variability and salivary cortisol. *Journal of Experimental Social Psychology, 47*(4), 849–852.

Khang, H., Kim, J. K., & Kim, Y. (2013). Self-traits and motivations as antecedents of digital media flow and addiction: The internet, mobile phones, and video games. *Computers in Human Behavior, 29*(6), 2416–2424.

Kim, H., Suh, K. S., & Lee, U. K. (2013). Effects of collaborative online shopping on shopping experience through social and relational perspectives. *Information and Management, 50*(4), 169–180.

Knierim, M. T., Nadj, M., Li, M. X., & Weinhardt, C. (2019). Flow in knowledge work groups – Autonomy as a driver or digitally mediated communication as a limiting factor? In *Proceedings of the International Conference on Information Systems.*

Koufaris, M. (2002). Applying the technology acceptance model and flow theory to online consumer behavior. *Information Systems Research, 13*(2), 205–223.

Kudaravalli, S., Faraj, S., & Johnson, S. L. (2017). A configural approach to coordinating expertise in software development teams. *MIS Quarterly*, *41*(1), 43–64.

Kuss, D. J., & Griffiths, M. D. (2012). internet gaming addiction: A systematic review of empirical research. *International Journal of Mental Health and Addiction, 10*(2), 278–296.

Kuusinen, K., Petrie, H., Fagerholm, F., & Mikkonen, T. (2016). Flow, intrinsic motivation, and developer experience in software engineering. In *Proceedings of the International Conference on Agile Software Development*.

Lavigne, G. L., Forest, J., & Crevier-Braud, L. (2012). Passion at work and burnout: A two-study test of the mediating role of flow experiences. *European Journal of Work and Organizational Psychology, 21*(4), 518–546.

Lavoie, R. V., & Main, K. J. (2019). When losing money and time feels good: The paradoxical role of flow in gambling. *Journal of Gambling Issues*, *41*(2019), 53–72.

Lin, W. K., Ma, H. C., Lin, C. P., & Chiu, C. K. (2009). Modeling knowledge sharing and interemployee helping from a perspective of flow theory: A survey of online knowledge workers. In *Proceedings of the Americas Conference on Information Systems*.

Liu, W., van der Linden, D., & Bakker, A. B. (2022). Strengths use and work-related flow: an experience sampling study on implications for risk taking and attentional behaviors. *Journal of Managerial Psychology*, *37*(1), 47–60.

Lune, H., & Berg, B. L. (2017). *Qualitative research methods for the social sciences*. Pearson.

Martin, R. (2011). *The clean coder: A code of conduct for professional programmers*. Pearson Education.

Mason, M. C., Zamparo, G., Marini, A., & Ameen, N. (2022). Glued to your phone? Generation Z's smartphone addiction and online compulsive buying. *Computers in Human Behavior*, *136,* 1-13.

Mattern, J., Tarafdar, M., Schellhammer, S., & Klein, S. (2024). Thriving in a bruising job: How high achieving IT professionals can cope with occupational demands. *Information Systems Journal*, *34*(6), 1902-1934.

Meyer, A. N., Fritz, T., Murphy, G. C., & Zimmermann, T. (2014). Software developers' perceptions of productivity. In *Proceedings of the ACM SIGSOFT Symposium on the Foundations of Software Engineering*.

Moneta, G. B. (2012). On the measurement and conceptualization of flow. In S. Engeser. (Ed.), *Advances in flow research* (pp. 23–50). Springer.

Mosing, M. A., Butkovic, A., & Ullén, F. (2018). Can flow experiences be protective of work-related depressive symptoms and burnout? A genetically informative approach. *Journal of Affective Disorders*, *226*, 6–11.

Mueller, L., & Benlian, A. (2022). 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*, *23*(6), 1420–1455.

Müller, S. C., & Fritz, T. (2015). Stuck and frustrated or in flow and happy: Sensing developers' emotions and progress. In *Proceedings of the International Conference on Software Engineering*.

Murphy, G. C. (2014). Getting to flow in software development. In *Proceedings of the ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming & Software*.

Myers, M. D. (1997). Qualitative research in information systems. *MIS Quarterly*, *21*(2), 241–242.

Myers, M. D. (2019). *Qualitative research in business and management*. Sage.

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

Nadj, M., Rissler, R., Adam, M. T. P., Knierim, M. T., Li, M. X., Maedche, A., & Riedl, R. (2023). What disrupts flow in office work? The impact of frequency and relevance of IT-mediated interruptions. *MIS Quarterly, 47*(4), 1615–1646.

Nah, F. F. H., Eschenbrenner, B., & DeWester, D. (2011). Enhancing brand equity through flow and telepresence: A comparison of 2D and 3D virtual worlds. *MIS Quarterly, 35*(3), 731–747.

Nakamura, J., & Csikszentmihalyi, M. (2002). The concept of flow. In C. R. Snyder & S. J. Lopez (Eds.), *Handbook of positive psychology* (pp. 89–105). Oxford University Press.

O'Connor, M., Coboy, K., Dennehy, D., & Carroll, N. (2024). Temporal complexity in information systems development flow: Challenges and recommendations. *Communications of the Association for Information Systems*, *54,* 699–735.

Partington, S., Partington, E., & Olivier, S. (2009). The dark side of flow: A qualitative study of dependence in big wave surfing. *Sport Psychologist*, *23*(2), 170–185.

Pelet, J. É., Ettis, S., & Cowart, K. (2017). Optimal experience of flow enhanced by telepresence: Evidence from social media use. *Information and Management, 54*(1), 115–128.

Pratt, J. A., Chen, L., & Cole, C. (2016). The influence of goal clarity, curiosity, and enjoyment on intention to code. *Behaviour and Information Technology, 35*(12), 1091–1101.

Quinn, R. W. (2005). Flow in knowledge work: High performance experience in the design of national security technology. *Administrative Science Quarterly*, *50*(2005), 610–641.

Ramsey, J. R., & Lorenz, M. P. (2020). Every flow has its ebb: The impact of flow on work-family conflict and adjustment in global careers. *Human Resource Management Journal*, *30*(3), 441–460.

Riemenschneider, C. K., & Armstrong, D. J. (2021). The development of the perceived distinctiveness antecedent of information systems professional identity. *MIS Quarterly*, *45*(3), 1149–1186.

Rissler, R., Nadj, M., & Adam, M. T. P. (2017). Flow in information systems research: Review, integrative theoretical framework, and future directions. In *Proceedings of Internationalen Tagung Wirtschaftsinformatik*

Rissler, R., Nadj, M., Li, M. X., Knierim, M. T., & Maedche, A. (2018). Got flow? Using machine learning on physiological data to classify flow. *In Proceedings of the ACM CHI Conference on Human Factors in Computing Systems.*

Ritonummi, S., Siitonen, V., Salo, M., & Pirkkalainen, H. (2024). Exploring barriers that prevent employees from experiencing flow in the software industry. *Journal of Workplace Learning, 36*(3), 223–238.

Ritonummi, S., Siitonen, V., Salo, M., Pirkkalainen, H., & Sivunen, A. (2023). Flow experience in software engineering. In *Proceedings of the ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*.

Rodríguez-Sánchez, A. M., Schaufeli, W., Salanova, M., Cifre, E., & Sonnenschein, M. (2011). Enjoyment and absorption: An electronic diary study on daily flow patterns. *Work and Stress*, *25*(1), 75–92.

Salanova, M., Bakker, A. B., & Llorens, S. (2006). Flow at work: Evidence for an upward spiral of personal and organizational resources. *Journal of Happiness Studies*, *7*(1), 1–22.

Salanova, M., Rodríguez-Sánchez, A. M., Schaufeli, W. B., & Cifre, E. (2014). Flowing together: A longitudinal study of collective efficacy and collective flow among workgroups. *The Journal of Psychology*, *148*(4), 435–455.

Schad, J., Lewis, M. W., Raisch, S., & Smith, W. K. (2016). Paradox research in management science: Looking back to move forward. *Academy of Management Annals*, *10*(1), 5–64.

Schaffer, O., & Feng, X. (2022). The feedback loop of flow: Controlled experiment shows task-relevant feedback increases flow. *AIS Transactions on Human-Computer Interaction*, *14*(3), 356–389.

Schultze, U., & Avital, M. (2011). Designing interviews to generate rich data for information systems research. *Information and Organization*, *21*(1), 1–16.

Schüler, J., & Nakamura, J. (2013). Does flow experience lead to risk? How and for whom. *Applied Psychology: Health and Well-Being*, *5*(3), 311–331.

Schüler, J., & Pfenninger, M. (2011). Flow impairs risk perception in kayakers. In: B. D. Geranto (Ed.), *Sport psychology* (pp. 237–246). Nova Publisher.

Sepehr, S., & Head, M. (2012). Dualistic model of passionate video gameplay: Addiction or flow? In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*.

Sepehr, S., & Head, M. (2018). Understanding the role of competition in video gameplay satisfaction. *Information and Management*, *55*(4), 407–421.

Sheth, S., Bell, J., & Kaiser, G. (2011). HALO (Highly Addictive, sociaLly Optimized) software engineering. In *Proceedings the International Workshop on Games and Software Engineering*.

Stack Overflow (2022). *Stack Overflow developer survey 2022*. Retrieved from https://survey.stackoverflow.co/2022#developer-profile-demographics

Stack Overflow. (2023). *Stack Overflow developer survey 2023*. Retrieved from https://survey.stackoverflow.co/2023/#developer-profile-demographics

Stavropoulos, V., Griffiths, M. D., Burleigh, T. L., Kuss, D. J., Doh, Y. Y., & Gomez, R. (2018). Flow on the internet: A longitudinal study of internet addiction symptoms during adolescence. *Behaviour and Information Technology*, *37*(2), 159–172.

Suh, A., Cheung, C. M. K., Ahuja, M., & Wagner, C. (2017). Gamification in the workplace: The central role of the aesthetic experience. *Journal of Management Information Systems*, *34*(1), 268–305.

Tarafdar, M., Cooper, C. L., & Stich, J. F. (2019). The technostress trifecta—techno eustress, techno distress and design: Theoretical directions and an agenda for research. *Information Systems Journal*, *29*(1), 6–42.

Terry, G., Hayfield, N., Clarke, V., & Braun, V. (2017). Thematic analysis. In C. Willig & W. Stainton-Rogers (Eds.), *The Sage handbook of qualitative research in psychology* (pp. 17–37). Sage.

Theotokis, A., & Doukidis, G. (2009). When adoption brings addiction: A use-diffusion model for social information systems. In *Proceedings of the International Conference on Information Systems*.

Trevino, L. K., & Webster, J. (1992). Flow in computer-mediated communication. *Communication Research*, *19*(5), 539–573.

Van der Linden, D., Tops, M., & Bakker, A. B. (2021). Go with the flow: A neuroscientific view on being fully engaged. *European Journal of Neuroscience*, *53*(4), 947–963.

Wang, C. C., & Hsu, M. C. (2014). An exploratory study using inexpensive electroencephalography (EEG) to understand flow experience in computer-based instruction. *Information and Management, 51*(7), 912–923.

Webster, J., Trevino, L. K., & Ryan, L. (1993). The dimensionality and correlates of flow in human-computer interactions. *Computers in Human Behavior*, *9*(4), 411–426.

Weinert, C., Maier, C., Laumer, S., & Weitzel, T. (2022). Repeated IT interruption: Habituation and sensitization of user responses. *Journal of Management Information Systems*, *39*(1), 187–217.

Weintraub, J., Cassell, D., & DePatie, T. P. (2021). Nudging flow through “SMART” goal setting to decrease stress, increase engagement, and increase performance at work. *Journal of Occupational and Organizational Psychology*, *94*(2), 230–258.

Wilson, E. E. (2016). *Metacognitions of flow experience: Towards an understanding of the self-regulation of optimal experience* (doctoral dissertation). London Metropolitan University.

Wilson, E. E., & Moneta, G. B. (2023). Flow metacognition and flow at work: Initial evidence of a directional relationship. *Personality and Individual Differences*, *214*.

Wu, T. C., Scott, D., & Yang, C. C. (2013). Advanced or addicted? Exploring the relationship of recreation specialization to flow experiences and online game addiction. *Leisure Sciences*, *35*(3), 203–217.

Wyrich, M., Graziotin, D., & Wagner, S. (2019). A theory on individual characteristics of successful coding challenge solvers. *PeerJ Computer Science*. Retrieved from https://doi.org/10.7717/peerj-cs.173

Zhang, H., Lu, Y., Gupta, S., & Zhao, L. (2014). What motivates customers to participate in social commerce? The impact of technological environments and virtual customer experiences. *Information and Management, 51*(8), 1017–1030.

Zheng, Z., Wang, L., Cao, Y., Zhuang, Y., & Tao, X. (2019). Towards non-invasive recognition of developers' flow states with computer interaction traces. In *Proceedings of the Asia-Pacific Software Engineering Conference*.

Zimanyi, Z., & Schüler, J. (2021). The dark side of the moon. In S. Engeser (Ed.), *Advances in flow research* (pp. 171–190). Springer.

Zubair, A., & Kamal, A. (2015). Work related flow, psychological capital, and creativity among employees of software houses. *Psychological Studies, 60*(3), 321–331.

Züger, M., Corley, C., Meyer, A. N., Li, B., Fritz, T., Shepherd, D., Augustine, V., Francis, P., Kraft, N., & Snipes, W. (2017). Reducing interruptions at work: A large-scale field study of FlowLight. In *Proceedings of Conference on Human Factors in Computing Systems*.

Zuill, W. (2019). *Mob programming and the power of flow.* GOTO Conferences. Retrieved from https://www.youtube.com/watch?v=28S4CVkYhWA

Appendix A: Coding Samples

Table A1. Coding Sample: Interview Number 2 (Software Developer)

Adverse outcome category	Manifestation subcategory	Description	Quote
False sense of productivity	Movement without momentum	In flow, there is a feeling of accomplishment; feeling efficient and/or productive, but the movement might take him in the wrong direction (i.e., there is a false sense of productivity).	*It [flow] usually makes you feel like you've accomplished things. I'm just saying that I feel like that, and not like I've achieved things necessarily, because, well, there have been times when I've coded something in the flow and later realized that it was wrong and didn't make any sense, so I've thrown it all in the trash. So, yes, it makes me feel that I was efficient, at least. At least there was movement in some direction—not necessarily that the movement would always have gone in the right direction.*
Adverse emotional responses	Guilt	Feeling "hungover" and guilty after being so immersed in a flow bubble that he has not been there for teammates. Likes to prioritize his own focused work but feels obliged to be available for junior colleagues' questions (hence, feelings of guilt).	*Of course, flow also means that I'm aware that I don't check my chat, email, or anything else. That's because I'm prioritizing my work, and I need to concentrate, so when the [flow] time is over, and especially if you're so immersed in it that you spend a really long time [in the flow], you might get a little hangover—like, oh, there are five persons in the chat asking for help, and I've just been in my own bubble. So, that kind of hangover can come from the fact that I have neglected my other work duties.*
Depleted cognitive resources	Reduced attentional performance	Flow is really straining and taxing on the brain.	*And it's quite taxing on the brain as well. Even though the flow state is a pretty good thing from the outset, and I feel it's a really positive thing to have reached it, it's definitely straining.*
Maladaptive immersion	Over-engagement	After being immersed in a fast-paced flow state, it is hard to come down to earth and adjust to the next activities. This can be either having trouble switching context to the next work task, such as a meeting, or from work to leisure.	*You will notice afterward, for example, that if you have listened to fast-paced heavy metal music and concentrated intensely, your brain will remain in a pretty fast-paced mode afterward, and it may be difficult to come back down to Earth (…) If you have a meeting right after a flow of this kind, you suddenly feel that everyone is talking so slowly, and you're like, "forward, forward, forward, forward". It's difficult to get away from it completely, or at least you should develop some ways to calm your brain a little afterward.*
Adverse emotional responses	Irritability	When engrossed in the flow, having to attend other work duties (e.g., meetings) causes irritability.	*Well, in the middle of the working day, some meetings can be so difficult to attend after flow. Maybe in the flow, you're so engrossed that five minutes before the meeting, you notice a pop-up saying that you have a meeting in five minutes, and it's just like, "Oh, no, why now!" Then you kind of go to the meeting with the wrong attitude. You mainly come to the meeting interrupted, and your brain is rolling faster than the others.*

Table A2. Coding Sample: Interview Number 24 (Scrum Master 2)

Adverse outcome category	Manifestation subcategory	Description	Quote
Physical strain	Pain	She feels physical strain afterward; back hurts.	*It feels afterwards like my back is pretty broken because I just sat there for six hours, and I didn't realize that.*
Maladaptive immersion	Over-engagement	Being too immersed in a flow state: Losing track of time and environment, focus on the programming task is so intense that she does not feel anything. Has conflicting feelings about this: Ponders if it is healthy flow to be so immersed.	*Time is disappearing; you notice that, oh, we've gone somewhere again—that's really how you notice it afterward. You don't necessarily notice it in the middle at all. So, it feels like you're so focused on it that you can't really feel what it feels like because then you kind of fall out of it, from that…of course, there would probably be, like, a healthier flow, but for me, it's just that I don't understand anything about time or place; I just write the code.*
Physical strain	Fatigue	Extreme exhaustion after intense thinking during flow.	*In retrospect, it's like my head is messed up, and my fuse has gotten a lot shorter. My fuse is getting shorter, and I'm really cross, but then when it's done, it's…done. So, there are not necessarily positive feelings afterward, and you need to do something to work that feeling off somehow.*
Adverse emotional responses	Irritability; anger	Feeling cross/angry, having a "shortened fuse".	
Depleted cognitive resources	Reduced attentional performance	Sometimes she feels that others do not understand how mentally straining coding, especially during an intense flow state, can be	*And it's so hard, especially for people who don't understand what this job is like, to understand that sitting here next to the computer is very hard when you're really, really coding—when it feels like even your brain cells are so exhausted afterward that you can't think about anything. In a way, it [flow] really exerts everything out of your brain, and then you're completely finished.*
Dependence on flow state	Addictive quality of flow	Coding provides immediate feedback and fast rewards, which is addictive, especially when you receive this rapid feedback during flow. Contemplates if coding attracts certain kinds of personalities—for example, people who get easily addicted.	*I've been wondering if it's the case that many, or probably quite a lot of, coders are, to some extent, addicted people—the kind that gets easily hooked because, in coding, the reward comes pretty quickly when things work. And then, the brain gets used to it and gets a kick out of it. It's difficult because it starts to get dangerous then. So that's how it's always been for me. And when we are in the flow, and we are doing test-driven development, it's constantly, "Yes, it works" or "Oh no, I broke it all up". But even so, we usually get pretty quickly to the situation that "Yes, it works again". Then, you kind of form an addiction to it.*

About the Authors

Saima Ritonummi is a Doctoral Researcher in the Information Systems department at the University of Jyväskylä, Finland, and a member of the IT & Human Behavior research group. Her research interests include the human aspects of software engineering, particularly flow experiences and technostress. She is currently completing her doctoral thesis on experiencing flow in software development. She has published in outlets such as the *Journal of Systems and Software*, *Journal of Workplace Learning*, and *Proceedings of the International Conference on Information Systems*.

Valtteri Siitonen is a Doctoral Researcher and Lecturer at the University of Jyvaskyla, Finland. His research focuses on themes involving IT and human behavior, with a particular focus on technostress, the use of conversational technologies, and flow experiences in knowledge work. His work has been published in several peer-reviewed academic outlets, including *Journal of Systems and Software*, *Journal of Workplace Learning*, and the *Proceedings of the International Conference on Information Systems (ICIS)*.

Dr. Markus Salo is a Professor of Information Systems and Academy Research Fellow at the University of Jyvaskyla, Finland. His research interests include personal/organizational use of IT and its impacts, technostress, coping, IT and human behavior, and games. Salo's research has been published in outlets such as *MIS Quarterly*, *Journal of Management Information Systems*, *Information Systems Journal*, *Communications of the Association for Information Systems*, *Journal of Service Research*, *International Journal of Information Management, Internet Research*, and the *Proceedings of the International Conference on Information Systems*. He has acquired major research funding and received several awards.

Henri Pirkkalainen, D.Sc. (Econ.), is a Professor of Information and Knowledge Management at the Tampere University (Finland). He received the doctoral degree in information systems from the University of Jyvaskyla (Finland) in 2014. His research interests include technostress, dark side of information systems use, and knowledge management. He has published in journals such as the *Management Information Systems Quarterly*, *Journal of Management Information Systems*, the *Information Systems Journal*, *Communications of the ACM*, and the *International Journal of Information Management*.