Exploring barriers that prevent employees from experiencing flow in the software industry

Saima Ritonummi, Valtteri Siitonen and Markus Salo
Faculty of Information Technology, University of Jyvaskyla, Jyvaskyla, Finland, and
Henri Pirkkalainen
Unit of Information and Knowledge Management, Faculty of Management and Business, Tampere University, Tampere, Finland

Received 14 November 2022
Revised 23 March 2023
30 May 2023
Accepted 20 July 2023

Abstract

Purpose – The purpose of this study is to investigate the barriers that prevent workers in the software industry from experiencing flow in their work.

Design/methodology/approach – This study was conducted by using a qualitative critical incident technique-inspired questionnaire.

Findings – The findings suggest that workers in the software industry perceive that the most obvious obstacles to experiencing flow are related to work not presenting enough cognitive challenges and situational barriers related to the characteristics of the job (e.g. workdays having too many interruptions and distractions, timetables often being considered too tight for creative exploration and problem solving and having negative user experiences with development tools).

Originality/value – The findings provide insights into flow barriers, specifically barriers that prevent workers in the software industry from experiencing flow.

Keywords Flow experience, Software industry, Software development

Paper type Research paper

Introduction

The software industry is a considerable business segment in the global economy (Venkatesh *et al.*, 2020). Software development is a knowledge-intensive industry (Edison *et al.*, 2013), and working in the industry requires good analytical, creative and problem-solving skills (Pratt *et al.*, 2016). When deeply involved in this line of work, employees can be expected to experience flow, which is a state of absorption and enjoyment in intrinsically motivated activities (Csikszentmihalyi, 1990). However, the knowledge-intensive and technology-driven nature of the industry (Edison *et al.*, 2013) also makes work complex, as the employees' individual skills require constant updating to keep abreast of evolving technologies (Kudaravalli *et al.*, 2017), and the limited number of talented professionals can result in unreasonable expectations being placed on those working in the field (Pratt *et al.*, 2016). This study examines the barriers that prevent employees in the software industry from experiencing flow in their work.

This research has received funding from the Foundation for Economic Education, Finland, and the Academy of Finland (341359).

Understanding these barriers can help propose interventions to address these challenges and make work a greater source of flow (Nakamura and Csikszentmihalyi, 2002), which can, in turn, support employee happiness (Zubair and Kamal, 2015) and intrinsic work motivation (Bakker, 2008). Motivation is important for improving employees' job satisfaction and decreasing turnover intentions. Furthermore, fostering employees' creative potential and ability to experience flow is important for organizational innovation and gaining competitive advantage (Janssens and Zaytsev, 2022; Zubair and Kamal, 2015).

Although previous studies have investigated how certain aspects of work in software development can facilitate flow, the understanding of flow barriers in the software industry remains limited (Ritonummi *et al.*, 2022). Studies that have addressed flow in this context have investigated software developers' emotions and perceived progress in tasks (Müller and Fritz, 2015) and user experience (UX) (Kuusinen *et al.*, 2016) as facilitators of flow. Additionally, studies investigating work productivity have identified that interruptions and task or context switching can result in decreased performance and fewer chances of experiencing flow in software development (Abad *et al.*, 2018; Meyer *et al.*, 2017). This paper addresses the research gaps identified in the flow and software development literature. Flow literature has called for more studies on the factors that *inhibit* flow experiences at work (Csikszentmihalyi *et al.*, 2017). The software development literature has investigated flow from specific angles, such as how it relates to emotions, work progress or UX. Thus, there have been calls for forming a contextual understanding of the phenomenon of flow in software development (Kalantari and Lethbridge, 2022; Pratt *et al.*, 2016). To address these gaps, we aim to answer the following research question:

RQ1. What prevents employees in the software industry from experiencing flow in their work?

Furthermore, due to the technology-driven nature of the software industry, we are interested in the human–computer interaction (HCI) perspective and the degree to which software industry employees' flow experiences are obstructed by the use of technology.

This study applied a qualitative critical incident technique (CIT)-inspired questionnaire. Using a CIT-style questionnaire enabled us to collect real-life descriptions of instances in which software industry employees were unable to experience flow in their work (N = 696). The findings provide a comprehensive view of flow barriers specific to the industry while also contributing to flow research. The most frequent barriers to experiencing flow in this context were work not providing an appropriate amount of challenge or work involving negative situational barriers, such as interruptions, distractions, time-related challenges and negative UX. These findings open an avenue for further research on flow in the software industry and flow barriers in the work context. The practical implications include suggestions to address these challenges with appropriate interventions.

Theoretical background

Flow can be characterized as an optimal experience that involves a state of deep concentration and enjoyment (Csikszentmihalyi, 1990), associated with intrinsically motivated, goal-directed activities that provide increasing challenges (Abuhamdeh, 2020). Work in the software industry usually requires a specific skill set and involves solving complex problems, enabling workers to gain intrinsic rewards for their work (Pratt *et al.*, 2016). These skills include job-related knowledge (e.g. programming skills) (Pratt *et al.*, 2016), personal capabilities (e.g. analytical and decision-making skills) (Acuña *et al.*, 2006) and soft skills (e.g. social skills and pressure tolerance) (Kudaravalli *et al.*, 2017; Matturro *et al.*, 2019). We based our examination on Csikszentmihalyi's (1975, 1990) conceptualization of flow.

Flow experience

The dimensions of flow include a perceived balance between one's skills and the challenges presented by an activity, clear proximal goals, immediate feedback, intense concentration, merging of action and awareness, loss of reflective self-consciousness, a sense of control over one's actions, distorted temporal experience (e.g. losing track of time) and an autotelic experience (i.e. experiencing the activity as intrinsically rewarding) (Nakamura and Csikszentmihalyi, 2002; Abuhamdeh, 2020). The first three dimensions are important for reaching flow and are often referred to as the conditions of flow. The following six dimensions describe the characteristics of being in a flow state. Flow usually occurs when the body or mind (or both) is stretched to its limit, and a challenging task is accomplished as a result (Csikszentmihalyi, 1990). Therefore, a balance between perceived challenges and perceived skills is essential for entering flow. Too low a challenge can result in boredom; conversely, too high a challenge can result in anxiety. In addition to the perceived challenge, the motivation to act on perceived opportunities arises from clear, proximal goals and the immediate feedback provided in these interactions (Nakamura and Csikszentmihalyi, 2002).

The factors that facilitate flow have received much more attention in flow research than the flow-inhibiting factors (Csikszentmihalyi *et al.*, 2017), which is not surprising considering that the concept of flow focuses on experiences involving intrinsic motivation and enjoyment (Csikszentmihalyi, 1990). Studies addressing flow-preventing factors (barriers) have mostly investigated flow in the context of sports (Chavez, 2008). In the work context, the understanding of flow barriers has been mostly limited to personal-level inhibitors (Csikszentmihalyi *et al.*, 2017). While Csikszentmihalyi *et al.* (2017) used the term *flow inhibitors*, we adopted the term *flow barriers*, following Kaufhold *et al.* (2019), who investigated UX-related barriers to initiating flow episodes in software development. The categorization of flow barriers that emerged in the results of this study was based on Csikszentmihalyi *et al.*'s (2017) grouping of flow-inhibiting factors into personal, interpersonal and situational factors. Personal-level barriers inhibit flow at the individual level, including emotional (e.g. negative affect), cognitive (e.g. mental exhaustion), behavioral (e.g. maladaptive coping) and motivational dispositions. Interpersonal factors explain how the involvement or presence of other people can influence flow in an organizational setting. Situational factors include the overall work situation (e.g. job characteristics and resources) and non-work-related situational factors that can affect the chances of experiencing flow (Csikszentmihalyi *et al.*, 2017). Only a high-strain job (i.e. a job with high demands but a low level of personal control) has been identified as a situational flow barrier (Fagerlind *et al.*, 2013), and perceived team climate (an interpersonal factor) has been found to affect individual flow experiences in development teams (Aichroth *et al.*, 2022).

Experiencing flow in the software industry

Flow has been widely researched in the work context (Bakker, 2008; Quinn, 2005; Taser *et al.*, 2022). Work in the software industry has often been attributed to providing excellent opportunities for work-related learning (Gijbels *et al.*, 2012) and experiencing flow, as it offers an opportunity to apply and improve one's skills (Pratt *et al.*, 2016) and engage in analytical thinking and creativity (Acuña *et al.*, 2006). In the software development context, empirical studies have investigated emotions as indicators of being in a flow state (Müller and Fritz, 2015) and the role of flow in the intention to code (Pratt *et al.*, 2016). Developers' emotions have been found to correlate with their perceived progress during work tasks (Müller and Fritz, 2015), and enjoyment has been found to increase intentions to engage in future coding projects (Pratt *et al.*, 2016). Moreover, sustained work-related flow can increase developers' happiness and excitement about their job, positively affecting their cognitive performance, creative output and flow (Graziotin *et al.*, 2018; Zubair and Kamal, 2015).

However, there is a gap in understanding flow-inhibiting factors, especially in forming a comprehensive view of different flow barriers software industry employees might encounter (Ritonummi *et al.*, 2022).

Another important aspect of working in the software industry is the heavy information technology (IT) use it involves. Although the IT users are usually the end users for whom the software is developed, in the context of this study, we refer to developers as IT users, as their work is heavily dependent on using various development tools, such as integrated development environments (IDEs). IT use can be a great source of flow, as it often involves tasks that offer goal-directed activities that present increasing challenges (Abuhamdeh, 2020) and which are enjoyable and possible to master (Sharafi *et al.*, 2006). Having an autotelic experience and positive UX with development tools can help developers reach flow (Kuusinen *et al.*, 2016). UX comprises both the pragmatic and hedonic aspects of HCI: in addition to fulfilling pragmatic, functional needs, the interaction must also fulfill hedonic needs, such as the need for fun, immersion and flow (Hassenzahl and Tractinsky, 2006). Interestingly, Kuusinen *et al.* (2016) found that, for developers, intrinsic motivation and autotelic experience are more important predictors of positive UX than the pragmatic and hedonic quality of the tool itself.

Furthermore, multiple studies have investigated productivity in software development, indirectly addressing flow as a component of a productive workday, among other staples of good work performance. While not investigating the psychological phenomenon of flow, these studies suggest that developers feel productive at work when they have clear goals and the opportunity to reach these goals, and when they can experience flow without too many interruptions, distractions or task and context switches (Abad *et al.*, 2018; Meyer *et al.*, 2017). As software development often involves high levels of cognitive demands, interruptions at inopportune moments can have detrimental effects on concentration (Vella and Porter, 2022). These can include coworkers' questions, being sidetracked to doing other tasks, priority changes and IT-mediated interruptions (e.g. email notifications), which can either be related to the task or completely shift their attention from the task at hand (Abad *et al.*, 2018). However, although often harmful to productivity, unplanned interruptions or switching can boost developers' productivity in some circumstances (e.g. if the interruption helps solve the task at hand) (Abad *et al.*, 2018). Furthermore, the ability to maintain flow and get back into the flow after an interruption can be considered an important skill (Janssens and Zaytsev, 2022).

Method

This study was conducted as a qualitative CIT-inspired online questionnaire. CIT offers a qualitative method for collecting self-reported critical incidents that have significantly negatively or positively affected an individual's activities (Flanagan, 1954; Gremler, 2004). This method is useful for exploring individual behavior in a work context, particularly in situations that are episodic in nature (Gogan *et al.*, 2014). CIT studies can rely either on direct observations (Schlichter and Rose, 2013) or retrospective critical incident reports (Salo and Frank, 2017). Moreover, retrospective self-reporting is common in flow studies (Bakker, 2008). In this approach, respondents are provided with a description of flow and then asked whether they have experienced it; if so, they are asked to describe their experiences in more detail (Abuhamdeh, 2020). As flow is a state characterized by intense absorption in an activity, Nakamura and Csikszentmihalyi (2002) argue that the most effective way to access instances of deep flow is through retrospection.

Data collection

We designed an online questionnaire using LimeSurvey. The aim was to gather descriptions of flow and barriers to experiencing flow in the software industry. The questionnaire consisted of 11 demographic background questions and 14 questions about flow, including open-ended and closed-ended questions. In the five open-ended questions, we instructed the participants to describe an outstanding flow experience they had and what they thought prevented them from experiencing flow (Appendix). The nine closed-ended questions gathered additional information about the experience, such as how long the experience lasted and how often the respondent experienced a similar flow. As the scope of the current study is to identify flow barriers in the respondents' work, we will be focusing on the question of flow-preventing factors. The questionnaire was pretested by three developers to ensure that the questions were understandable.

Data collection was conducted in two phases between December 2021 and April 2022. In the first phase, we collected data by contacting the 100 largest software companies by market cap in Finland and globally. We also contacted Finnish IT industry trade unions and software development forums. The questionnaire was internally shared by 15 organizations and in 13 software development and trade union forums ($n = 61$). In the second phase, primary data were collected using the Prolific online panel ($n = 674$), a service that has been found helpful in collecting high-quality data (Peer *et al.*, 2017). No significant changes to the questionnaire design were made, except for the addition of the following screening criteria: respondents who worked in the software industry (in Prolific, this entailed software development, video games and information services and data processing), a minimum of 20 submissions and an approval rate of 97% on Prolific and two attention check questions. Respondents who fulfilled the prescreening criteria were eligible to answer the questionnaire on a first-come, first-served basis. In these two phases, we collected 735 responses in total, of which 696 were included in the final sample.

The age range of the final sample was 18–73 years ($M = 32.06$). The respondents were 72.3% men, 26.9% women and 0.7% others (0.1% did not disclose). The majority (78.4%) reported working mainly on a team, and 21.4% worked independently (0.1% did not disclose). Further demographic information about the final sample is presented in Table 1.

The most frequently mentioned countries of residence were the UK (16.1%), South Africa (10.5%), Portugal (10.0%), Finland (8.6%) and the USA (8.3%). The most frequently mentioned job titles included software developer, software engineer, front-end developer, Web developer and full-stack developer.

Data analysis

The objective of the data analysis was to gain insights into the barriers that prevent employees in the software industry from experiencing flow. The data analysis included responses from software developers and other IT professionals working in the software industry. We applied the following exclusion criteria:

- students whose experiences were not related to work ($n = 21$);
- "I do not know" responses ($n = 10$);
- failed attention check questions ($n = 1$); and
- those who had misunderstood the question or answered it incoherently ($n = 7$). This resulted in the exclusion of 39 responses from the final sample.

In the first stage of the data analysis, relevant emergent themes (flow barriers) from the responses were identified and open coded (Myers, 2020). This stage was conducted mainly

Age	
≤ 24	21.0%
25–34	46.1%
35–44	19.5%
≥ 45	10.8%
Employment	
Full-time	78.9%
Part-time	9.5%
Freelancer	6.3%
Entrepreneur	3.2%
Other	1.7%
Unemployed	0.3%
Education	
Doctoral degree	1.7%
Master's degree	27.6%
Bachelor's degree	49.6%
High school or equivalent	19.8%
Less than a high school education	0.6%
Experience in current role	
Less than a year	14.5%
1–2 years	24.3%
3–5 years	29.7%
6–10 years	15.4%
Over 10 years	15.9%

Source: Authors' own

Table 1. Demographics ($N = 696$)

by the first author. Preliminary flow barriers were established after reading the data from the first phase ($n = 59$). Examples of the preliminary barriers were *distractions and interruptions* and *technology-related barriers*. These barriers were refined after the primary data ($n = 637$) were analyzed and when a sufficient level of thematic saturation was reached (i.e. no additional knowledge was expected from collecting more responses) (Myers, 2020). At this point, the barriers and their categorization were discussed among all authors. We aimed for mutually exclusive barriers whenever possible and labeled each response with all flow barriers mentioned in the response. Most responses included descriptions of multiple barriers. The data-driven content analysis was followed by the second stage of analysis, which involved sorting the identified barriers into literature-based categories. We compared the barriers identified in the first stage of the analysis with the categories of flow-inhibiting factors in Csikszentmihalyi *et al.* (2017). The barriers were then sorted into these categories after determining whether they were personal, interpersonal or situational in nature. The identified barriers remained mainly the same, except for minor tweaks to the descriptions of some barriers.

Results

We identified 21 flow barriers to the software industry employees' work. The identified barriers were grouped into personal, interpersonal and situational, according to Csikszentmihalyi *et al.* (2017). The barriers are shown in Table 2, along with their respective categories and the number of mentions per barrier. The software development context-specific findings are in bold in the column describing the characteristics of each barrier.

Category	Barrier	Characteristics *(software development-specific)*	*n*
Situational (503)	Interruptions and distractions	Physical and virtual interruptions, distractions	149
		Task/context switching between development tasks	
	Time-related challenges	Time pressure on deadlines	95
		Unexpected changes (e.g. changing requirements)	
		Waiting for others' input	
		Getting stuck with a task	
	Negative UX	Dissatisfactory UX	92
		Development tools not working (e.g. buggy code editor)	
		Poor code quality (e.g. unclear code)	
	Insufficient resources	Lack of human resources	63
		Budget constraints	
		Lack of proper tools (hardware/software)	
		Inadequate or missing documentation	
	Lack of clear goals	Unclear work tasks or project objectives	50
		Insufficient requirements analysis	
	Lack of autonomy	Inability to affect one's work	25
	Excessive or inflexible policies and practices	Bureaucracy, strict procedures	20
		Too many development/project management practices	
	Blurred work–home boundaries	Working from home causes troubles with work–life balance	9
		Distractions from home life while working	
Personal (350)	Work does not present enough challenges	Tasks do not present enough cognitive challenge	152
	Work presents too much challenge	Tasks present too much cognitive challenge	84
	Lack of motivation	Lack of work engagement, disinterest in work	43
	Inappropriate focus	Lack of focus, concentration problems	23
	Stress	Stress from work and private life	20
	Non-optimal arousal level	Being tense/nervous or not alert enough	13
	Negative emotions	Sadness, self-doubt, etc.	11
	Procrastination	Procrastination at work	4
Interpersonal (94)	Poor management	Lack of good leadership or efficient management	23
	Poor team dynamics	Lack of cooperation and teamwork	21
	Poor client relations	Having trouble with clients (e.g. non-engaged clients)	19
	Lack of communication	Lack of communication within the organization/with clients	19
	Lack of support	Insufficient support or help from supervisors/ colleagues	12

Source: Authors' own

Table 2. Flow barriers among workers in the software industry

Additionally, 26 respondents stated that nothing had prevented them from experiencing flow in their work (e.g. "There's nothing stopping me from experiencing flow more often in my work. It's just my choice.").

Situational barriers

Situational flow barriers included interruptions and distractions (159), time-related challenges (95), negative UX (92), insufficient resources (63), lack of clear goals (50), lack of autonomy (25), excessive or inflexible policies and practices (20) and blurred work–home boundaries due to working from home (9). Unsurprisingly, *interruptions and distractions* were experienced as the second-most prominent barrier to flow. These entailed both physical and virtual interruptions (e.g. colleagues' questions at the office, emails and instant messaging), distractions (e.g. background noise and a busy environment), task and context

switching between development tasks and projects and the frequency of meetings (having "too many meetings" or "unnecessary meetings"). Interruptions were not necessarily considered negative *per se*, but the respondents felt that interruptions were too frequent, which caused them to have trouble being absorbed in their work. One respondent commented, "The order of terror: Slack, teams, calendar bookings, email." In particular, developers felt that task and context switching were detrimental to their ability to experience flow at work, as they often had to jump between many development tasks, projects or different user interfaces in the same task. Moreover, some respondents said that they had managerial tasks in addition to development work. This was considered harmful because there seemed to be insufficient time for blocks of focused work and being absorbed in a task:

> Fragmented workdays. Previously, as a full-time software developer, time was better allocated to app development. Now, all sorts of questions from other developers and meetings break up the day. For me, a flow state requires an extended, uninterrupted moment. (Technical architect).

Time-related challenges were related to feeling pressure due to overly restricted or unrealistic timetables, facing unexpected changes, waiting for others' input and getting stuck in a task. Unexpected changes comprised last-minute changes, changing priorities (e.g. client changing requirements halfway through the project) and having to improvise when a project was not progressing as planned, which added to time pressure. Waiting was also considered flow-inhibiting, as it was perceived as leading to unnecessary idleness that disrupted flow. For example, respondents often had to wait for replies to submitted tasks or wait for help. Furthermore, getting stuck in a task was considered to disrupt flow (e.g. running into problems or hitting a dead end):

> The deadline is too close and worrying about not having enough time. (Software developer).

Negative UX included dissatisfactory UXs and the use of "non-user-friendly" tools. Specifically, developers mentioned having a negative UX with IDEs as a barrier to their flow:

> When the software or technology becomes a barrier (limited features, too cumbersome to use, etc.) to deliver the output I envisioned. (Project manager).

Negative UX also involved experiencing technical difficulties and tools that did not work as expected (e.g. slow or buggy code editor, glitches, incompatible systems and unexpected data losses). Furthermore, negative UX entailed poor code quality, including issues such as bugs, entropy and legacy code (i.e. being forced to work with a specific codebase or trying to determine solutions derived by others in the past).

Insufficient resources were described as a lack of needed resources, such as staff, budgets and proper tools. This was mostly characterized by being too short-staffed to manage the workload or having insufficient technical resources, including hardware and software (e.g. working with outdated systems). Specifically characteristic to software development, this barrier included inadequate documentation. This was described as either insufficient or completely missing documentation, which inhibited flow because of having to use the time to find missing information rather than solving the task:

> Poor technology choices, poor documentation (especially for the technology used), unclear task assignment or requirements. (CTO, Full-stack developer).

In addition, many respondents stated that uncertainty or ambiguity in their work regarding the *lack of clear goals* prevented them from experiencing more flow. This was characterized by unclear or poorly defined tasks and scope of work, conflicting demands and insufficient

requirements (i.e. the requirements analysis was either poorly done or not clearly communicated to the developers):

> Having unclear requirements, I have to constantly bounce back and forth between my IDE and other communication tools. Also, having meetings that interrupt my flow and productivity. (Software developer).

A lack of autonomy (25) was characterized by a feeling of not having a voice, a lack of control or an inability to affect one's work (e.g. not being able to choose tasks or being assigned tasks that no one else wanted):

> I rarely get to choose what I'm being thrown into. (Team lead).

Excessive or inflexible policies and practices comprised excessive bureaucracy, strict company procedures and guidelines (including company-imposed technologies), inflexible working hours, performing inefficient routines and too many development or project management practices. Finally, situational flow barriers included *blurred work–home boundaries* as a result of working from home. Because of the increase in remote work during the last few years, respondents described having trouble balancing work and home life. Some said that working from home negatively affected the atmosphere at home, whereas others said it was challenging to focus on work at home because they were distracted by family members or pets.

Personal barriers

Although personal barriers were the second most prominent category, they included the most prominent barrier to experiencing flow in the software industry: work tasks not presenting enough challenge (152) in relation to one's skills. Conversely, personal barriers also involved work presenting too much challenge (84). However, the mismatch between perceived challenge skill–balance is not entirely dependent on the individual's own cognitive resources but also on the job/task characteristics and the kind of challenges they pose. Therefore, these two barriers intersect with situational factors. Because being too low of a challenge or too high of a challenge is essentially about individuals' inability to use their resources in a meaningful way (i.e. they are either bored from not having any mental challenges or anxious about having excessive challenges), they were categorized into personal barriers. *Work does not present enough challenges* was characterized by too easy, boring or mundane tasks, tasks that did not require strong commitment and a lack of learning experiences and creativity (e.g. not being able to innovate or do creatively demanding tasks). In addition, some respondents described how the cyclical nature of software development, in which only some phases of the development process provide enough challenge for them, affects their ability to experience flow at work:

> My work includes a lot of mundane tasks, such as little alterations and additions to existing programs so that they meet the new needs of clients. This requires no creativity on my part. It is not challenging, and frankly, it is plain boring. (Senior developer).

By contrast, *work presents too much challenge* was also a considerable barrier to flow. This included responses that tasks were too challenging (i.e. hard-to-resolve and complex problems), the skills or experience required to solve such tasks were lacking (i.e. technical skills or overall experience in the field), and there was constant pressure to learn (i.e. steep learning curves and having to learn and master an ever-increasing number of technologies constantly), all of which were related to the feeling that the challenges exceeded the respondents' competency levels:

> Fairly regularly, I face tasks and challenges that I'm unprepared to complete, coupled with tight time frames. I need to look for an answer, learn a new skill, or improve in another way. (Mobile developer).

Motivational inhibitors were related to a *lack of motivation* (43), which was also described as a lack of passion, commitment, engagement, or disinterest in work. Some respondents described the circumstances that caused them to lose their motivation as follows: work feels meaningless, stagnant or does not evoke interest and not being able to do what they love:

> Not all tasks are motivating or stimulating enough for me. There are many duties that I deem trivial, procedural, and not engaging. (Team lead).

Personal barriers also included *inappropriate focus* (23), described as a lack of focus, inattention, concentration problems, being easily distracted or bored and having trouble sustaining intense focus for extended periods:

> I have concentration problems, so it is really hard for me to get absorbed in something. (Software developer).

Furthermore, personal barriers involved stress (20), non-optimal arousal levels (13) and negative emotions (11). *Stress* was characterized by being stressed or burned out due to work or from stress in private life. Sources of stress from personal life arose from duties related to family, friends and love life, as well as from societal crises (e.g. economic and military crises). The *non-optimal arousal level* concerned either feeling restless or anxious, or not being alert enough, mainly due to a lack of sleep or rest, a lack of exercise, poor diet, or health conditions. *Negative emotions* included sadness, feeling overwhelmed, worrying, overthinking and self-doubt (e.g. fear of not being competent enough). These barriers were often intertwined:

> Rush, stress, and self-doubt about my own skills. (Specialist).

> It's hard to get into the flow when I'm overwhelmed or burned out. (Web designer).

Finally, personal barriers involved the behavioral act of *procrastination* (4), which was mentioned by only a handful of respondents and was simply stated as procrastination or the urge to put off tasks.

Interpersonal barriers

Interpersonal barriers to flow included having trouble with management, team dynamics, or clients and a lack of communication or support. *Poor management* (23) was described as a lack of good leadership, pressure from above and micromanagement. *Poor team dynamics* (21) involved a lack of teamwork, a competitive or toxic environment and interpersonal conflicts:

> Working as a team of people with different opinions and criticisms. (Web developer).

Poor client relations (19) were related to clients being unclear about what they needed or wanted, constantly changing their requirements, and not being engaged with the development team or open to ideas. *Lack of communication* (19) was characterized by not having enough communication within the team or organization or with clients. Lack of communication also entailed trouble communicating or expressing oneself. *Lack of support* (12) involved descriptions of a lack of recognition, insufficient support or help, neglect from supervisors and having to work alone:

> Too often, the needs of the company are placed before the needs of the individual. Not enough thought is given to how individuals work, just how tooling benefits the company.

Different interpersonal barriers were often mentioned alongside each other (i.e. lack of proper management and conflicts within the team usually involved a lack of communication, teamwork or mental support).

Discussion

The findings of this study demonstrated that the most frequently mentioned barriers to experiencing flow in the software industry employees' work were situational. Specifically, interruptions and distractions, time-related challenges and negative UX were considered flow-preventing. Among the most mentioned barriers were also personal-level barriers, especially a mismatch between the perceived challenge–skill balance. This study also identified five interpersonal barriers to experiencing flow. These findings can help guide future research on flow barriers in the work context and flow in the software industry.

Research contributions

This study makes six contributions to research. First, it identifies a comprehensive set of barriers to experiencing flow in the software industry, contributing to initial findings on flow barriers in software developers' work (Ritonummi *et al.*, 2022). The barriers include not only personal but also situational and interpersonal flow inhibitors, which contribute to the understanding of flow barriers in the work context (Csikszentmihalyi *et al.*, 2017). These barriers included both work-related and software development-specific flow barriers.

Second, the findings highlight how situational barriers, especially interruptions and distractions, can have a detrimental effect on software industry employees' ability to experience flow. As a flow state requires highly disciplined mental activity (Csikszentmihalyi, 1990), it is not surprising that external interruptions disturb immersion in the activity and prevent reaching flow and sustaining this mental state. Specific to software development, being interrupted or distracted involved frequent task and context switching between different development tasks, projects or even tools while performing one task (e.g. switching between IDE and communication channels). This complements research on developers' productivity and performance (Abad *et al.*, 2018; Meyer *et al.*, 2017) by highlighting that interruptions and distractions affect not only employees' productivity but also their inability to experience flow. Combined with other situational barriers (e.g. lack of autonomy or inflexible policies and practices, such as excessive development practices adopted by the organization), interruptions were linked to decreased work motivation and losing one's passion for the job or the whole industry. Moreover, the respondents pointed out that time pressure and a constant sense of urgency about project deadlines did not allow for exploration and finding pleasure in performing tasks. This resulted in not having enough time for experimenting, discovering or learning, which are important aspects of being able to experience flow, especially in the IT use context (Sharafi *et al.*, 2006).

Third, we identified flow barriers related to IT use. While positive UX has been associated with facilitating flow, the findings of the present study call attention to the different ways negative UX can contribute to inhibiting flow. Specifically, working with tools that prevented making progress by causing different kinds of problems (e.g. buggy software, data losses), working on poor-quality code or having an overall dissatisfactory UX were considered barriers to flow. Having negative UX with the tools was reported to shift attention from the task at hand to solving other secondary problems. As developers have been found to appreciate the efficiency, informativeness and flexibility of the development tools they use (Kuusinen *et al.*, 2016), in light of these findings, it can be inferred that UX can not only increase the intrinsic motivation to use IT and enable more flow, fun and immersion (Hassenzahl and Tractinsky, 2006) but also work as a barrier to experiencing flow.

Fourth, this study identified many personal-level barriers to experiencing flow in the software industry. The most frequently mentioned barrier in this study was work not presenting enough challenges. The fifth most frequently mentioned barrier was work tasks presenting too much challenge. In other words, a lack of perceived challenge–skill balance,

which is often considered the most important condition for flow (Nakamura and Csikszentmihalyi, 2002), prevented the respondents from experiencing flow in their work. The absence of intellectual stimulation can result in lower engagement in work (de Moura and de Oliveira Rosas, 2021), which was reported by many respondents. In addition, learning experiences and skill improvement are important in the software industry (Edison *et al.*, 2013; Gijbels *et al.*, 2012). The current findings demonstrated that not having any learning experiences or being faced with frequent learning or excessively steep learning curves prevented software industry employees from experiencing flow. In the HCI context, it has been suggested that a match or mismatch between the expected and actual performance levels is linked to experiencing flow (Palomäki *et al.*, 2021). In the software industry, this can manifest in developers feeling frustrated with lower-than-expected performance when learning new skills and applying them in practice, causing them to question their skills and self-efficacy, leading to less flow. These findings contribute to a better understanding of the role of perceived challenge–skill balance in experiencing flow in the software development context (Pratt *et al.*, 2016) by suggesting that its absence can be a considerable barrier to experiencing flow.

Fifth, the findings provide much-needed insights into interpersonal flow barriers in the work context. Although ways of improving the work environment have been addressed in flow research, the importance of interpersonal and social factors as flow inhibitors has been overlooked (Csikszentmihalyi *et al.*, 2017). Apart from Aichroth *et al.* (2022), who found that the perceived team climate could affect individual flow in development teams either positively or negatively, most studies on interpersonal factors or collective flow have focused on the facilitators of flow (Salanova *et al.*, 2014). The findings of the present study demonstrated that an unsupportive or even toxic climate at work could be a barrier to flow, whether caused by problematic interactions with management, the team or clients. In addition, a lack of communication and mental support negatively affected the respondents' ability to experience flow.

Finally, the findings showed that trouble balancing work and personal life when working from home was considered stressful. This increased the understanding of non-work-related flow barriers, particularly because working from home has mostly been considered to facilitate flow (Peters *et al.*, 2014). Contrary to Taser *et al.* (2022), who found that remote work had a significant positive effect on flow and improved employee well-being and mental health, these findings brought to light the flow-preventing side of working from home and the consequent blurring of work and home boundaries.

Practical implications

The three major practical implications practitioners can draw from the present study are addressing the challenges of interruptions and distractions, negative UX and a mismatch of the perceived challenge–skills balance. First, practical measures for shaping work environments to support flow include reducing or optimizing the number of interruptions and distractions (e.g. planning the timing of meetings carefully and allowing sufficient time blocks for focused work). Although resilience to distractions can help employees recover from negative incidents and experience more positive emotions and flow (Janssens and Zaytsev, 2022; Zubair and Kamal, 2015), mitigating the harmful effects of constant interruptions requires not only individual practices but also an initiative from management to support employees' ability to focus deeply on their work. This includes acknowledging the discrepancy in what is regarded as meaningful and productive use of time in different roles and the tradeoff between facilitating individual flow and team productivity.

The second implication is mitigating the problems caused by negative UX. Negative UX can cause a variety of challenges (e.g. attention shifting from the task at hand or not being able to complete the tasks because of technical difficulties). Hence, it is important that the development stack supports positive UX and provides timely feedback about the interactions. This can include, for example, IDEs that provide simplified project resource management or a recommendation system that suggests further tasks similar to the finished task, as suggested by Kaufhold *et al.* (2019).

Third, the ability to fulfill one's potential without being bored by the lack of challenges or being confronted with excessively difficult tasks or overly steep learning curves is crucial for achieving flow. The respondents hoped that there would be opportunities to prove their skills and to be assigned more meaningful and interesting tasks. Although coordinating expertise in the software industry is challenging (Kudaravalli *et al.*, 2017), providing these opportunities and allowing more autonomy is important for facilitating flow. While many highly skilled professionals are proactive in choosing interesting projects, this implication also requires managerial initiative. Ultimately, understanding how to enable flow and obstruct it less in organizations is important for recruiting employees who demonstrate intrinsic motivation and curiosity, and for keeping them engaged and satisfied with their work.

Limitations

First, the methodological limitations of this study include the possible shortcomings of the CIT-style questionnaire: self-reports are subject to misunderstanding and recall bias. Moreover, category labels and coding rules are subject to ambiguity in the content analysis approach (Gremler, 2004). As flow is a subjective experience, it is difficult, if not impossible, to measure it directly (Quinn, 2005). Further, it can be challenging to set boundaries between flow and non-flow states (Pratt *et al.*, 2016), as people do not often report sudden transitions between being in flow and not being in flow (Abuhamdeh, 2020). Second, data collection-related limitations involve challenges in collecting responses from online panels and the size of the data set. In online panels, it is possible that respondents do not give their full attention to the task (Peer *et al.*, 2017). However, we tried to mitigate this shortcoming by applying prescreening and elimination criteria to identify those who would not give their full attention to the task. Finally, it is possible that not all barriers related to experiencing flow in the software industry were present in the findings and that more barriers could emerge from a larger dataset.

Future research

An interesting avenue for future research on flow in software development is the further examination of flow-enabling factors in the software industry. There could be similarities to the identified flow-facilitating factors in the work context (Csikszentmihalyi *et al.*, 2017) and to the barriers identified in this study but also differences (e.g. the personal-level factors could be more important than the situational factors for facilitating flow). Interpersonal factors could be investigated further by focusing on the role of management, team dynamics and communication skills in being able to experience flow at work, as well as how *collective flow* (Aichroth *et al.*, 2022; Salanova *et al.*, 2014), could be facilitated by balancing teamwork and meetings so that they allow sufficient time blocks for focused work and reaching flow. Given these points, the findings of this study provide many avenues for future research on flow in software development.

References

Abad, Z.S.H., Karras, O., Schneider, K., Barker, K. and Bauer, M. (2018), "Task interruption in software development projects", *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018*, doi: 10.1145/3210459.3210471.

Abuhamdeh, S. (2020), "Investigating the 'flow' experience: key conceptual and operational issues", *Frontiers in Psychology*, Vol. 11, doi: 10.3389/fpsyg.2020.00158.

Acuña, S.T., Juristo, N. and Moreno, A.M. (2006), "Emphasizing human capabilities in software development", *IEEE Software*, Vol. 23 No. 2, pp. 94-101, doi: 10.1109/ms.2006.47.

Aichroth, L.S., Nuszbaum, M. and Campoy-Gómez, L. (2022), "Do you feel it? The relationship between the perceived team climate for innovations and the experience of flow and worry", *International Journal of Innovation and Technology Management*, Vol. 19 No. 6, pp. 2250019-1-2250019-16, doi: 10.1142/s0219877022500195.

Bakker, A.B. (2008), "The work-related flow inventory: construction and initial validation of the wolf", *Journal of Vocational Behavior*, Vol. 72 No. 3, pp. 400-414, doi: 10.1016/j.jvb.2007.11.007.

Chavez, E.J. (2008), "Flow in sport: a study of college athletes", *Imagination, Cognition and Personality*, Vol. 28 No. 1, pp. 69-91, doi: 10.2190/ic.28.1.f.

Csikszentmihalyi, M. (1975), *Beyond Boredom and Anxiety*, Jossey-bass Publishers, San Francisco, CA.

Csikszentmihalyi, M. (1990), *Flow: The Psychology of Optimal Experience*, HarperCollins, New York, NY.

Csikszentmihalyi, M., Khosla, S. and Nakamura, J. (2017), "Flow at work", in Steger, M. (Ed.), *The Wiley Blackwell Handbook of the Psychology of Positivity and Strengths-Based Approaches at Work*, Wiley Blackwell, Chichester, pp. 99-109.

de Moura, P.J., Jr. and de Oliveira Rosas, N. (2021), "Perceptions about flow and boredom in the information technology profession", *International Journal of Human Capital and Information Technology Professionals*, Vol. 12 No. 4, pp. 1-17, doi: 10.4018/ijhcitp.2021100101.

Edison, H., Bin Ali, N. and Torkar, R. (2013), "Towards innovation measurement in the software industry", *Journal of Systems and Software*, Vol. 86 No. 5, pp. 1390-1407, doi: 10.1016/j.jss.2013.01.013.

Fagerlind, A.-C., Gustavsson, M., Johansson, G. and Ekberg, K. (2013), "Experience of work related flow: does high decision latitude enhance benefits gained from job resources?", *Journal of Vocational Behavior*, Vol. 83 No. 2, pp. 161-170, doi: 10.1016/j.jvb.2013.03.010.

Flanagan, J.C. (1954), "The critical incident technique", *Psychological Bulletin*, Vol. 51 No. 4, pp. 257-272.

Gijbels, D., Raemdonck, I., Vervecken, D. and Van Herck, J. (2012), "Understanding work-related learning: the case of ICT workers", *Journal of Workplace Learning*, Vol. 24 No. 6, pp. 416-429, doi: 10.1108/13665621211250315.

Gogan, J.L., McLaughlin, M.D. and Thomas, D. (2014), "Critical incident technique in the basket", *Thirty Fifth International Conference on Information Systems (ICIS)*.

Graziotin, D., Fagerholm, F., Wang, X. and Abrahamsson, P. (2018), "What happens when software developers are (un)happy", *Journal of Systems and Software*, Vol. 140, pp. 32-47, doi: 10.1016/j.jss.2018.02.041.

Gremler, D.D. (2004), "The critical incident technique in service research", *Journal of Service Research*, Vol. 7 No. 1, pp. 65-89, doi: 10.1177/1094670504266138.

Hassenzahl, M. and Tractinsky, N. (2006), "User experience – a research agenda", *Behaviour and Information Technology*, Vol. 25 No. 2, pp. 91-97, doi: 10.1080/01449290500330331.

Janssens, S. and Zaytsev, V. (2022), "Go with the flow", *Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings [Preprint]*, doi: 10.1145/3550356.3559101.

Kalantari, R. and Lethbridge, T.C. (2022), "Preliminary results of measuring flow experience in a software modeling tool", *Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings [Preprint]*, doi: 10.1145/3550356.3559099.

Kaufhold, M.A., Reuter, C. and Ludwig, T. (2019), "Flow experience in software engineering: development and evaluation of design options for eclipse", *Proceedings of the 27th European Conference on Information Systems (ECIS)*, Research-in-Progress Papers.

Kudaravalli, S., Faraj, S. and Johnson, S.L. (2017), "A configural approach to coordinating expertise in software development teams", *MIS Quarterly*, Vol. 41 No. 1, pp. 43-64, doi: 10.25300/misq/2017/41.1.03.

Kuusinen, K., Petrie, H., Fagerholm, F. and Mikkonen, T. (2016), "Flow, intrinsic motivation, and developer experience in software engineering", *Lecture Notes in Business Information Processing*, Springer, Cham, Vol. 251, pp. 104-117, doi: 10.1007/978-3-319-33515-5_9.

Matturro, G., Raschetti, F. and Fontán, C. (2019), "A systematic mapping study on soft skills in software engineering", *Journal of Universal Computer Science*, Vol. 25 No. 1, pp. 16-41.

Meyer, A.N., Barton, E.L., Murphy, G.C., Zimmermann, T. and Fritz, T. (2017), "The work life of developers: activities, switches and perceived productivity", *IEEE Transactions on Software Engineering*, Vol. 43 No. 12, pp. 1178-1193, doi: 10.1109/tse.2017.2656886.

Müller, S.C. and Fritz, T. (2015), "Stuck and frustrated or in flow and happy: Sensing developers' emotions and progress", *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, doi: 10.1109/icse.2015.334.

Myers, M.D. (2020), *Qualitative Research in Business and Management*, 3rd ed., Sage.

Nakamura, J. and Csikszentmihalyi, M. (2002), "The concept of flow", in Snyder, C.R. and Lopez, S.J. (Eds), *Handbook of Positive Psychology*, Oxford University Press, pp. 89-105.

Palomäki, J., Tammi, T., Lehtonen, N., Seittenranta, N., Laakasuo, M., Abuhamdeh, S., Lappi, O. and Cowley, B.U. (2021), "The link between flow and performance is moderated by task experience", *Computers in Human Behavior*, Vol. 124, p. 106891, doi: 10.1016/j.chb.2021.106891.

Peer, E., Brandimarte, L., Samat, S. and Acquisti, A. (2017), "Beyond the Turk: alternative platforms for crowdsourcing behavioral research", *Journal of Experimental Social Psychology*, Vol. 70, pp. 153-163, doi: 10.1016/j.jesp.2017.01.006.

Peters, P., Poutsma, E., Van der Heijden, B., Bakker, A. and De Brujin, T. (2014), "Enjoying new ways to work: an HRM-process approach to study flow", *Human Resource Management*, Vol. 53 No. 2, pp. 271-290, doi: 10.1002/hrm.21588.

Pratt, J.A., Chen, L. and Cole, C. (2016), "The influence of goal clarity, curiosity, and enjoyment on intention to code", *Behaviour and Information Technology*, Vol. 35 No. 12, pp. 1091-1101, doi: 10.1080/0144929x.2016.1171399.

Quinn, R.W. (2005), "Flow in knowledge work: high performance experience in the design of national security technology", *Administrative Science Quarterly*, Vol. 50 No. 4, pp. 610-641, doi: 10.2189/asqu.50.4.610.

Ritonummi, S., Siitonen, V., Salo, M. and Pirkkalainen, H. (2022), "Flow barriers: what prevents software developers from experiencing flow in their work", *Proceedings of the 8th International Workshop on Socio-Technical Perspective in Information Systems Development (STPIS 2022)*.

Salanova, M., Rodríguez-Sánchez, A.M., Schaufeli, W.B. and Cifre, E. (2014), "Flowing together: a longitudinal study of collective efficacy and collective flow among workgroups", *The Journal of Psychology*, Vol. 148 No. 4, pp. 435-455, doi: 10.1080/00223980.2013.806290.

Salo, M. and Frank, L. (2017), "User behaviours after critical mobile application incidents: the relationship with situational context", *Information Systems Journal*, Vol. 27 No. 1, pp. 5-30, doi: 10.1111/isj.12081.

Schlichter, B.R. and Rose, J. (2013), "Trust dynamics in a large system implementation: six theoretical propositions", *European Journal of Information Systems*, Vol. 22 No. 4, pp. 455-474, doi: 10.1057/ejis.2012.24.

Sharafi, P., Hedman, L. and Montgomery, H. (2006), "Using information technology: engagement modes, flow experience, and personality orientations", *Computers in Human Behavior*, Vol. 22 No. 5, pp. 899-916, doi: 10.1016/j.chb.2004.03.022.

Taser, D., Aydin, E., Torgaloz, A.O. and Rofcanin, Y. (2022), "An examination of remote e-working and flow experience: the role of technostress and loneliness", *Computers in Human Behavior*, Vol. 127, p. 107020, doi: 10.1016/j.chb.2021.107020.

Vella, V. and Porter, C. (2022), "Wait a second! assessing the impact of different desktop push notification types on software developers", *33rd European Conference on Cognitive Ergonomics (ECCE'22)*, doi:10.1145/3552327.3552328.

Venkatesh, V., Thong, J.Y.L., Chan, F.K.Y., Hoehle, H. and Spohrer, K. (2020), "How agile software development methods reduce work exhaustion: insights on role perceptions and organizational skills", *Information Systems Journal*, Vol. 30 No. 4, pp. 733-761, doi: 10.1111/isj.12282.

Zubair, A. and Kamal, A. (2015), "Work related flow, psychological capital, and creativity among employees of software houses", *Psychological Studies*, Vol. 60 No. 3, pp. 321-331, doi: 10.1007/s12646-015-0330-x.

Appendix

Questionnaire instructions and open-ended questions:

Please take a moment to recall an outstandingly strong flow experience related to the use of technology or software in your work.

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.

You can take a few minutes to recall. This time is allowed for the duration of the survey:

- Please describe the flow experience in as much detail as possible in your own words:
- What exactly enabled the flow experience?
- What prevents you from experiencing flow in your work more often? (question related to this study)
- How can you enable more flow experiences in your work in the future?

(The questionnaire also included other questions which are not discussed in this paper).

Corresponding author
Saima Ritonummi can be contacted at: saemrito@jyu.fi or saima.e.ritonummi@jyu.fi