The Journal of Systems and Software 225 (2025) 112341

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

Coping with technostress in the software industry: Coping strategies and factors underlying their selection

Valtteri Siitonen [a,*], Saima Ritonummi [a], Markus Salo [a,b], Henri Pirkkalainen [c], Saija Mauno [d,e]

[a] Faculty of Information Technology, University of Jyvaskyla, PO Box 35, FI-40014, Jyvaskyla, Finland
[b] Oulu Advanced Research on Service and Information Systems, University of Oulu, FI-90570, Oulu, Finland
[c] Faculty of Management and Business, Tampere University, FI-33014, Tampere, Finland
[d] Faculty of Education and Psychology, University of Jyvaskyla, PO Box 35, FI-40014, Jyvaskyla, Finland
[e] Faculty of Social Sciences, Tampere University, FI-33014, Tampere, Finland

ARTICLE INFO

Editor: Dr. Christoph Treude

Keywords:
Technostress
Coping
Coping strategy
Software industry
Software development
Software engineering

ABSTRACT

Background: Software industry employees are subject to the harmful effects of technostress, which is caused by information technology use.
Aim: The aim of the current study is to understand how software industry employees cope with technostress and how their employers can best support these coping efforts.
Method: We employed the critical incident technique using a qualitative approach and collected our data through a questionnaire. In total, we collected and analyzed 715 real-life accounts of coping with critical technostress incidents from employees working in the industry.
Results: We identified 27 strategies for coping with technostress and provided a comprehensive categorization of them. Additionally, we uncovered novel coping strategies and highlighted the contextual (organizational and personal) factors that influence the selection of coping strategies.
Conclusions: Our findings help in explaining the reasons and motives for up-taking specific coping strategies in the software industry. Based on our findings, employees invest considerable effort into coping with technostress, but do not receive enough support for their coping efforts. We offer insightful suggestions for support mechanisms that could be used by software organizations to enhance employees' coping efforts and well-being.

1. Introduction

Stress has been defined as "a relationship between the person and the environment that is appraised by the person as taxing or exceeding his or her resources and endangering his or her wellbeing" (Lazarus and Folkman, 1984, p. 19). Stress is a common occurrence among those working in the software industry (i.e., organizations involved in the development, maintenance, and publication of software), and software work has even been considered as one of the most stressful lines of work (Furuyama et al., 1997; Wong et al., 2023). The stress caused by IT use, which is known as technostress, contributes to the high stress experienced by employees (Nisafani et al., 2020; Siitonen et al., 2022; Tarafdar et al., 2011, 2019). In the software industry, technostress is caused by the high IT use–related demands, including employees having to master the use of many kinds of complex IT (e.g., development tools) (Rajeswari and Anantharaman, 2003), adapting to ever-changing IT (Chilton et al., 2005), technology fatigue (Wong et al., 2023), and being under pressure to perform time-critical tasks (e.g., system testing) that require deep technical know-how (Sonnentag et al., 1994; Venkatesh et al., 2020). Technostress is harmful because it endangers employees' well-being by exposing them to adverse consequences, such as burnout and exhaustion (Reinecke et al., 2017). Experiencing technostress and impaired well-being can also decrease work quality, lead to an increased number of development errors, and lower employees' motivation (Furuyama et al., 1997; Graziotin et al., 2018; Windeler et al., 2017). Thus, to minimize the harmful consequences of technostress, it is crucial to

Abbreviations: Coping strategy (CS), Concrete cognitive, emotional, and behavioral attempts to manage stressors and actualize a coping attempt in practice; Coping family (CF), Core categories of coping that nest together similar CSs (with the same function for coping).

* Corresponding author.
E-mail address: valtteri.m.e.siitonen@jyu.fi (V. Siitonen).

https://doi.org/10.1016/j.jss.2025.112341
Received 7 June 2024; Received in revised form 21 November 2024; Accepted 9 January 2025
Available online 10 January 2025

understand how those working in the industry respond to the ever-increasing technological demands.

Despite the high stress experienced by software industry employees and growing interest in improving their well-being (e.g., Graziotin et al., 2018; Suárez and Vizcaíno, 2024; Trinkenreich et al., 2023; Wong et al., 2023), research on how employees cope with the stress in the industry has received limited attention. The literature on coping originates in psychology, and in this framework, coping is defined as the efforts "to manage external and/or internal demands that are appraised as taxing or exceeding the resources of a person" (Lazarus and Folkman, 1984, p. 141). Indeed, only a few studies have explored the phenomenon of coping, either in specific contexts of software development (e.g., code reviews in Ebert et al., 2019) or through the use of specific coping strategies (CSs) (e.g., the use of psychoactive substances in Newman et al., 2023). Despite these valuable efforts, no *collective* understanding exists regarding how employees in the industry cope with work stress. Furthermore, although IT use is a common source of stress in the industry (Siitonen et al., 2022), employees have been found to struggle with finding ways to cope with the poor mental well-being caused by it (Wong et al., 2023). Understanding coping is key because employees and teams could better evaluate their coping behavior and its possible ineffectiveness, thus making wiser coping decisions and improving well-being. Insights into coping would also be crucial in helping software organizations support the coping efforts of their employees. To address these research gaps, we aimed to answer the following two research questions: *"How do employees cope with technostress in the software industry?" "How do contextual factors of the software industry affect the selection of CSs?"*

We applied the critical incident technique (CIT), a research method for studying human behavior via critical experiences reported by participants (Flanagan, 1954), in a qualitative research setting. We collected our data via a questionnaire and asked those working in the software industry to describe their critical technostress incidents and coping responses in detail. We used content analysis and open coding to identify the specific CSs reported and then categorized them into coping families (CFs) proposed by Skinner et al. (2003), to reveal the most common core types of CSs and why they are used in the industry. Our findings contribute to the software engineering and technostress literature by (1) providing a comprehensive understanding regarding the CSs used by software industry employees battling technostress, (2) explaining the contextual (organizational and personal) underlying factors that support or hinder the CS selection of employees, and (3) suggesting mechanisms (e.g., IT control and autonomy) for software organizations to support the coping efforts of their employees.

The present paper is organized as follows: The next section covers the background literature. This section presents the important concepts in theories on coping, prior literature on coping with technostress, and prior studies on coping in the software industry. The methods and materials section explains the CIT method and our approach to data collection and analysis. The results section presents the findings, which encompass the identification of various CSs utilized by employees coping with technostress, along with the underlying factors influencing the adoption of these strategies. Finally, the discussion goes over the implications and limitations of the study as well as recommendations for future research.

2. Background

Technostress was first defined in the 1980s as "a modern disease of adaptation caused by inability to cope with new computer technologies in a healthy manner" (Brod, 1984, p. 16). Following the early definitions and works on technostress from the 1980s, technostress research has gained momentum, especially during the past two decades (Salazar-Concha et al., 2021). These days, (techno)stress research across various disciplines (e.g., information systems, psychology, and medicine) has often defined stress through transactional stress theory (Lazarus and Folkman, 1984). Using transactional stress theory, technostress is defined as a process in which the conditions of the IT use environment are appraised as demands by the individual (stressors), resulting in attempts made by the individual to manage these demands (coping responses/efforts), which again leads to positive (via successful coping) or negative (via unsuccessful coping) stress outcomes (Lazarus and Folkman, 1984; Tarafdar et al., 2019, p. 8). It is important to note that the feeling of being stressed is a highly subjective experience because individuals react differently to the demands they face (DeLongis et al., 1982).

2.1. (Techno)stress and the software industry

Software engineering research has connected various stressors to the stress experienced by software industry employees. For example, Rajeswari and Anantharaman (2003) suggested 10 stressors that are relevant in the context of software work. Three of these stressors (technical constraints, obsolescence, and individual–team interactions) also include links to IT use through the fear of not knowing how to use the latest IT and constraints imposed by IT. Technological issues have also been connected to the stress and lowered productivity experienced in software development teams, along with other stressors, such as work pressure, lack of trust, and communication issues (Suárez and Vizcaíno, 2024). Other studies have noted the fast pace of technological change, work overload, and role stressors, such as role conflict and role overload, as being connected to the experienced stress (Sonnentag et al., 1994; Windeler et al., 2017). Along with identifying stressors, prior literature has aimed to create methodologies for assessing and measuring stress in the industry (Ostberg et al., 2020) and understanding the antecedents and outcomes of burnout (Tulili et al., 2023).

Furthermore, the concept of technostress has been directly connected to the stress experienced by software industry employees through different technostressors (Siitonen et al., 2022). Some of the most recognized technostressors that are applicable to software work include excessive IT use, the rapid introduction of new IT, and complexity caused by hard-to-use IT (Fischer and Riedl, 2017; Sellberg and Susi, 2014; Tarafdar et al., 2011). Encountering stressful interactions with IT use and being unable to cope with them can also lead to a plethora of harmful outcomes experienced by technostressed individuals. These outcomes include physiological (e.g., increased blood pressure in Riedl et al., 2012), psychological (e.g., burnout in Reinecke et al., 2017), and behavioral outcomes (e.g., lowered productivity in Tarafdar et al., 2010). In organizational contexts that heavily rely on teamwork, such as in the software industry (Suárez and Vizcaíno, 2024), these harmful outcomes experienced by stressed individuals can also have negative consequences on the team climate and motivation.

2.2. Coping and its conceptualization

Coping theories aim to explain and understand the process of coping and identify the cognitive, emotional, and behavioral attempts undertaken by individuals to remove or reduce the impact of stressors, that is, CSs (Biggs et al., 2017; Cheng et al., 2014; Lazarus and Folkman, 1984). In the coping process, a stressed individual evaluates the significance of the incident (primary appraisal), determines whether the incident is a valid threat to their well-being, and, depending on the result of the appraisal, picks a suitable, actionable CS to manage the stressor (secondary appraisal) (Lazarus and Folkman, 1984). Subsequent rounds of appraisals and the up-take of alternative CSs may also occur, depending on whether the chosen coping approach was deemed unsuccessful by the individual. Thus, successful coping is a key process in determining individual stress outcomes in terms of well-being (Biggs et al., 2017; Skinner et al., 2003).

Coping is a complex situational process that is influenced by the context, environment, and resources an individual has available to them when deciding on a suitable CS. Alongside the CSs used by software

industry employees, we are also interested in the underlying contextual factors that influence coping responses; these factors can include personal (e.g., self-efficacy), emotional (e.g., calmness), social (e.g., social contact), and environmental (e.g., information available) resources that are (un)available to the employee (Matheny et al., 1993; Oakland and Oastell, 1996; Van den Brande et al., 2020). These factors vary between different contexts of coping, can affect the individual's appraisal of a given (possibly) stressful situation, and can make specific CSs easier or more difficult to employ (Oakland and Oastell, 1996).

The literature on coping has aimed to create higher-level conceptualizations of coping that include categorizations of individual CSs (Skinner et al., 2003; Skinner and Zimmer-Gembeck, 2016). This has been seen as a crucial endeavor to gain a more comprehensive understanding regarding the types of CSs used by individuals in different contexts, the effectiveness of these CSs, and comparing the CSs identified in different contexts with each other (Skinner and Zimmer-Gembeck, 2016). One such conceptualization is proposed by Skinner et al. (2003) consisting of three distinct levels. The first level refers to the actionable CSs employed by individuals. The second level consists of 12 core CFs, each representing and nesting CSs that serve a similar function for coping. The third and final level includes three adaptive processes that guide the function represented by each of the CFs. These adaptive processes are larger evolutionary processes that guide individuals' basic desires to coordinate their actions, social resources, and available options in possibly threatening situations. The three adaptive processes all include four CFs:

- *Coordinating actions and contingencies in the environment,* including the CFs of problem-solving, information seeking, helplessness, and escape.
- *Coordinating reliance and social resources in the environment,* including the CFs of self-reliance, support seeking, delegation, and isolation.
- *Coordinating preferences with options available in the environment,* including the CFs of accommodation, negotiation, submission, and opposition.

Although the proposed conceptualization and existence of core CFs has received support in both conceptual (Skinner et al., 2003; Skinner and Zimmer-Gembeck, 2007) and empirical analyses (e.g., Duhachek and Oakley, 2007; Webster et al., 2016), we would like to point out that there currently exists no universal agreement regarding the "core categories" of coping that the CFs present. However, the CFs proposed by Skinner et al. (2003) can tackle many of the criticisms that other popular coping categorization systems, such as dividing CSs broadly into problem-focused (addressing the problem directly) and emotion-focused (regulating emotions) CSs, have received. These prevalent criticisms include unclear categorizations, oversimplifying the coping process, and masking critical differences between actionable CSs (Biggs et al., 2017; Skinner et al., 2003; Skinner and Zimmer-Gembeck, 2016). Thus, we decided to adopt the coping conceptualization proposed by Skinner et al. (2003) to (1) gain a more holistic theoretical understanding of the common core types (i.e., CFs) of CSs and why they are used by software industry employees and (2) make our findings easier to compare with the coping literature (i.e., the CFs offer a way of categorizing CSs from different contexts under the same core categories, making contextualized findings easier to compare) (Skinner and Zimmer-Gembeck, 2016). The main concepts relevant to our study are summarized in Table 1.

Table 1
Summary of the main concepts.

Concept	Description	References
Coping	Individuals' efforts to manage demands appraised as taxing or exceeding their resources.	Lazarus and Folkman, 1984; Skinner et al., 2003
Underlying factor	Contextual and situational factors (e.g., environmental or personal resources) (un)available for an individual that affect the selection and employment of specific CSs.	Matheny et al., 1993; Oakland and Ostell, 1996; Van den Brande et al., 2020
Conceptual levels of coping		
Coping strategy (CS)	Lower-level categories that consist of numerous and specific cognitive, emotional, and behavioral attempts to manage stressors and actualize a coping attempt in practice. ***Example:*** An individual decides to avoid engagement with their development environment (because of constant bugs in the code) by going out for a walk and doing something else actualizing a CS *"behavioral disengagement."*	Skinner et al., 2003
Coping family (CF)	Intermediate level categories that nest together similar CSs that represent the same function for coping. Also referred to as the core categories of coping. ***Example*:** The CF *"escape"* nests all CSs that represent the function of escaping a stressful situation. Other forms of escape from engaging with buggy code could include *refusing to use the development environment, changing jobs*, or *avoiding thinking about the issue*.	Skinner et al., 2003
Adaptive process	Higher-level, sometimes unconscious and involuntary, processes that guide the basic needs and actions of individuals during stressful transactions and help them adapt to their environment. These processes guide the function represented by each CF. ***Example*:** The individual recognizes a threat to their well-being because of the unease and pressure caused by the constant bugs and errors. This makes them adapt to the stressful situation by fleeing from the stressful transaction (fight, flight, or freeze response).	Skinner et al., 2003

2.3. Coping with technostress and coping in the software industry

Regarding prior literature on IT use and coping, the concept of coping has been used to address user adaptation to new IT (Beaudry and Pinsonneault, 2005), information security behavior (D'Arcy et al., 2014), IT use–related interruptions (Galluch et al., 2015), and users' coping responses to stressful IT use incidents in both free time (Salo et al., 2020) and organizational contexts (Pirkkalainen et al., 2019). The research on coping with technostress has often adapted the aforementioned categorization of CSs into problem- and emotion-focused CSs (e.g., Salo et al., 2020; Zhao et al., 2020). Examples of the identified CSs include fixing IT problems, adjusting one's own IT use (problem-focused CSs), downplaying the problem, and blaming the IT (emotion-focused CSs). Other approaches have included the identification of proactive (e.g., IT control) and reactive (e.g., venting) CSs when coping with technostress (Pirkkalainen et al., 2019). Other ways to mitigate the harmful effects of technostress (not directly related to coping) include organizational support mechanisms, such as technical support and the involvement of users in technological decision-making (Fuglseth and

Sørebø, 2014; Ragu-Nathan et al., 2008). Although these prior studies have proposed valuable insights into technostress and coping, it is plausible that critical details regarding the variety of CSs and their main functions when coping with technostress might have been missed. Furthermore, addressing the contextual factors affecting CS selection has received limited attention, a viewpoint on which we focus.

In software engineering research, the concept of coping has been used to investigate the effects of CSs used by software professionals on their well-being and productivity during the COVID-19 pandemic (Russo et al., 2021), the CSs used by software developers to cope with confusion during code reviews (Ebert et al., 2019), the coping mechanisms for dealing with uncertainty in information systems development (Taipalus et al., 2020), and the use of psychoactive substances to overcome stress among software developers (Newman et al., 2023). Examples of popular CSs identified here include changing one's attitude, denial, seeking emotional support, and disengagement. Regarding the organizational mechanisms for stress reduction, the possibilities of organizational culture (belonging, climate for learning, and inclusiveness) (Trinkenreich et al., 2023) and the integration of mental well-being into the design of technologies used by software engineers (Wong et al., 2023) have been investigated as possible ways to mitigate stress. Other potential mechanisms for reducing the stress experienced by software development teams include improving communication with the team, reducing the workload, and improving team spirit (Suárez and Vizcaíno, 2024). Despite software engineering research utilizing the concept of coping and identifying various CSs, there seems to exist no comprehensive view of the different CSs used by software professionals or of the organizational support mechanisms that could best support employees' coping efforts, viewpoints on which we focus.

3. Materials and methods

CIT is a research method whose purpose is to gain insights into a phenomenon of interest by collecting, analyzing, and classifying incidents that have had either a significantly negative or positive effect on an individual (Flanagan, 1954). CIT is widely considered as a flexible research method without rigid rules. However, a general set of phases and recommendations for employing the method have been proposed based on syntheses of prior CIT studies (e.g., Gremler, 2004). These common phases are as follows: First, the problem and aims of the research are determined. This is followed by planning and executing the collection of critical incidents, which can be done in different ways, such as in the form of questionnaires or interviews. The collected incidents are then analyzed (often utilizing the principles of content analysis), and finally, the results are reported. The insights gained from CIT studies include the ways in which individuals respond to a specific phenomenon, the outcomes caused by the phenomenon, and "the frequency and patterns of factors" that affect the phenomenon (Gremler, 2004, p. 66). In the current study, we aimed to gain insights into how individuals respond to the phenomenon of technostress by identifying the CSs used by software industry employees and the contextual underlying factors affecting their coping behavior. Based on the discussion of the CIT method, its advantages, and the aims of our study, we deemed the CIT as an appropriate method for the following reasons:

- First, CIT allows for collecting data from the respondents' points of view in their own words (Edvardsson, 1992). Using CIT allowed the respondents to bring forth their perspectives and actual real-life stories regarding coping with technostress, providing a rich source of data.
- Second, coping is a highly personal and multidimensional process, and individuals can react to stressful situations in different ways (Lazarus and Folkman, 1984; Skinner et al., 2003). Thus, we saw it as important not to bind the respondents to a given framework or terminology or restrict the respondents in their descriptions (Gremler, 2004) regarding technostress and coping.
- Finally, we decided to choose the CIT because it has been proven to be a valuable tool for investigating actual human behavior/responses and enabling novel insights into a previously unmapped context (e.g., Salo et al., 2020; Salo and Frank, 2017; Serenko and Turel, 2010).

3.1. Data collection

In collecting the critical incidents, we utilized a qualitative online questionnaire. The questionnaire included both open- and closed-ended questions with wording adapted from studies using CIT; the wording we used has been found to produce high-quality data related to IT use (e.g., Salo and Frank, 2017; Najmul Islam, 2014). In the context of our study, we defined a critical incident as an exceptionally stressful and/or burdensome IT use experience. Drawing from this definition, we first posed the respondents with the following instructions: "*Recall an exceptionally burdensome/stressful experience related to the use of technology or software in your work*." In the open-ended questions, we asked the respondents to describe 1) "the burdensome/stressful experience in as much detail as possible in your own words," 2) "technology and/or software involved in the experience," and 3) "how did you try to reduce or mitigate the effects of the burdensome/stressful experience." The closed-ended questions included additional information about the experiences: the duration of the experience, when it occurred, and the strain caused by the experience. The questionnaire was first crafted in Finnish and then carefully translated into English with the help of an English language professional. We asked two individuals working as software developers and one individual working as a project manager (leading a software development team) to pretest the questionnaire to make sure that the questions were clear and understandable. The outline of the questionnaire is presented in Appendix A and in an online replication package, which is linked in Section 5.3.

We conducted a pilot phase of data collection by contacting organizations engaged in software development in Finland via email between December 2021 and February 2022. In total, 20 organizations agreed to distribute the questionnaire internally to their employees, either via email or messaging apps (e.g., Slack or Teams). In total, we received 63 responses from employees working as, for example, software developers, system testers, project managers, and product owners. Data quality was deemed sufficient, and no major changes were made to the questionnaire when proceeding to the primary data collection phase.

Primary data collection was carried out between March 2022 and April 2022 using the Prolific online research platform. Prolific and similar online platforms have proven useful for collecting large quantities of high-quality data (see Lowry et al., 2016; Peer et al., 2017). To ensure that we could gather data from individuals working in the software industry, we used the prescreening criteria in Prolific and included the following industries: "software," "information services and data processing," and "video games." We also asked the respondents for their job titles to further clarify the role in which they worked. To ensure the quality of our data, we applied the suggestions made by Prolific by adding two attention check questions. We also made our questionnaire available only to those who had at least 20 prior submissions on the platform and an acceptance rate of 97 % or higher. The respondents received a monetary compensation of approximately two euros for answering the questionnaire according to the Prolific guidelines.

Including the pilot sample, we obtained a total of 852 responses, of which 715 were included in the final sample. The exclusion criteria are elaborated on in the Section 3.2. The most common nationalities reported among those in the final sample were British (11.5 %), Portuguese (11.2 %), South African (9.5 %), and American (7.8 %). In total, 47 nationalities were present, with 10.5 % of the respondents preferring not to disclose their nationalities. Most respondents worked as either developers (44.9 %) or in managerial positions (e.g., product owners and project managers) (13.4 %). Other common job titles included system testers (10.6 %), analysts (6.4 %), software consultants (4.9 %),

and systems designers (3.4 %). Of the respondents, 9.9 % preferred not to disclose their job titles, and 6.4 % worked under miscellaneous job titles, such as 3D modelers in game development, systems architects, and system administrators. The respondents were mainly involved in software development, but many were also responsible for the continuous and ongoing maintenance of software products (e.g., new feature planning and bug fixing). A summary of the demographic information of the final sample is presented in Table 2. For further details, refer to the replication package in Section 5.3.

3.2. Data analysis

Preceding the main phase of the analysis, we established exclusion criteria by adapting the suggestions made for CIT research (Flanagan, 1954; Gremler, 2004). First, we excluded responses that conflicted with the context of our study: *coping with critical stressful incidents caused by IT use at work in the software industry*. Thus, we excluded responses from full-time students (39), responses unrelated to the use of IT (27), and responses reporting the strain caused by the experience to be either "low" or "very low" on a 5-point Likert scale (44) (signaling the non-criticality of the stressful incident). Second, we excluded responses from those who did not provide sufficient detail in their descriptions of the stressful incident or coping effort (25) (e.g., "… I don't remember just work harder") and who failed the attention check questions in Prolific (2). In total, we excluded 137 out of the total 852 responses, which left us with 715 responses to be analyzed in the main analysis phase.

To analyze the data (i.e., identify the CSs used in the software industry), we used the content analysis approach commonly used in CIT research (Gremler, 2004) by following a set of phases and recommendations presented by Gremler (2004). The analysis phases included the following: (1) reading the incidents to familiarize with the data, (2) identifying recurring themes, (3) developing a coding scheme, (4) sorting incidents following the coding scheme, (5) assessing intracoder reliability, (6) having an independent analyst sort incidents, (7) measuring interrater reliability, and (8) testing the coding scheme on a holdout sample. In addition, we have also provided details on how we aimed to ensure consistency throughout the coding process. Next, we describe our adherence to the abovementioned general content analysis phases in CIT research presented in Gremler (2004).

(1) *Read, reread incidents*: The first author familiarized themselves with the data by twice reading through 665 out of the 715 incidents included in the analysis phase. A holdout sample of 50 incidents was left unaddressed at this point to test the coding scheme in a later phase. This familiarization gave us an overview of the types of incidents and how employees coped with them.

(2) *Identify recurring themes*: The first and the second authors read through and discussed an initial sample of 200 incidents. The aim was to identify recurring themes regarding the coping efforts reported by the respondents. In practice, we sought similarities between the coping efforts described by the respondents and grouped these into initial categories (CSs) under descriptive labels utilizing a data-driven approach. We decided to use a data-driven approach in developing the initial categories because of the situational and context-specific nature of coping (Skinner and Zimmer-Gembeck, 2016). In the end, the authors identified 27 potential CSs.

(3) *Develop classification scheme*: We continued by mirroring the initial categories with the literature on coping with technostress and coping with stress in the software industry, here seeking to find similarities between them. In the case of similarities and resemblance, we used the labels and descriptions from prior research to inform our descriptions and labels for each of the identified CSs in our coding scheme. For example, efforts to cope with technostress that involved taking a temporary break from using IT (e.g., development tools or communication technologies) were grouped under "temporary disengagement from IT use" (present in, e.g., Salo et al., 2017). The initial categories that did not include clear links to prior research were labeled and described based on descriptions drawn from the data. Table B1 in Appendix B describes whether each CS was labeled based on data or prior literature. After this, all the authors engaged in a round of discussion regarding the descriptions and labels of the identified CSs. All the identified CSs were discussed, with examples raised from the data. These discussions resulted in small adjustments to the labels and descriptions of the CSs and the creation of a testable coding scheme.

(4) *Sort incidents using classification scheme*: The first author coded 50 incidents (the holdout sample) based on the developed coding scheme.

(5) *Assess intracoder reliability*: The same set of incidents were coded again by the first author with the goal of measuring intrarater reliability. The percent-agreement was measured at 97 % and Cohen's kappa at 0.97.

(6) *Have additional judges/coders sort incidents*: To verify our coding scheme, we asked an independent analyst (not part of our research team) to code the same set of 50 responses. To train the analyst (Syed and Nelson, 2015), we held a meeting with the analyst so that they could ask possible questions regarding individual categories and their descriptions. The analyst also received the coding scheme that included each of the established codes and their descriptions.

(7) *Assess intercoder reliability*: The measured percent-agreement between the raters (author 1 and the independent analyst) was measured at 80 % and Cohen's kappa at 0.77, indicating a substantial level of agreement (Stemler, 2004).

(8) *Test classification scheme on a holdout (validation) sample*: We used the holdout sample of 50 responses to test our coding scheme (described in the previous phases). The finalized coding scheme used in the coding process is presented in Appendix B.

Coding the rest of the data and ensuring the consistency of the coding process: Following the finalized coding scheme, the first author continued coding the remaining accepted responses. The coding of the new data was constantly compared with the coded data and established codes to ensure that our coding process remained consistent (Berg, 2004). If a respondent described that they had used multiple CSs, the relevant portion of the response was coded to all the relevant strategies (example 2 in Table 3). After all the responses had been coded, unclear data descriptions

Table 2
Demographic information of the final sample ($n = 715$).

Gender	~%	Age	~%	Employment status	~%	Experience	~%
Male	70.5 %	≤ 25	26.6 %	Full time	81.7 %	< 1 year	14.4 %
Female	28.7 %	26–35	41.3 %	Part time	8.9 %	1–2 years	25.3 %
Other	0.5 %	36–45	19.1 %	Freelancer	5.2 %	3–5 years	28.1 %
Not disclosed	0.3 %	≥ 45	10.2 %	Entrepreneur	3.8 %	6–10 years	16.3 %
		Not disclosed	2.8 %	Unemployed	0.3 %	> 10 years	15.8 %
				Not disclosed	0.1 %	Not disclosed	0.1 %

were discussed in a meeting between the authors to ensure that the coding of the incidents remained consistent and coherent to the coding scheme (Syed and Nelson, 2015). The coding of these incidents did not lead to any adjustments regarding the established coding scheme.

With the CSs identified and coded, we moved to classifying them into CFs based on the categorization proposed by Skinner et al. (2003). The aim was to gain a more holistic and easily comparable (Skinner and Zimmer-Gembeck, 2016) understanding of the core types of CSs used by software industry employees and through the functions of the CFs, understand why they are used. To achieve a coherent categorization, we used descriptions of the CFs in the literature to verify the sorting. For example, the CS "temporary disengagement from IT use" was grouped into the CF "escape" based on the following description of its function in Skinner et al. (2003): "(escape) includes (coping) efforts to disengage or stay away from the stressful transaction" (p. 242). Definitions of all the relevant CFs are presented in the results section in Table 4.

As for the contextual underlying factors, we went through the data again, and by utilizing open coding, aimed to identify repeating patterns in the data that led the respondents toward adopting specific CSs. For example, one of the underlying factors for employing the CS "temporary disengagement from IT use" was "temporary exhaustion and awareness of technostress build-up" (e.g., "I closed my laptop … *because I was exhausted and just done for the day*").

Finally, the sorting of the CSs into the CFs and labels of the underlying factors were discussed among the authors, resulting in minor changes to the labels of the underlying factors. Two example chains of evidence that further demonstrate pattern-building and its results are presented in Table 3. For a detailed summary of our data analysis, including the exclusion criteria, coding scheme, analysis results, and closed-ended question results, refer to the online replication package in Section 5.3.

4. Results

Our results include a categorization of 715 descriptions of coping with critical stressful IT use incidents into different CSs, and further, their respectful CFs. These CSs were used when the respondents experienced technostress caused by, for example, learning new IT, complex development tasks, IT-enabled interruptions, and technical malfunctions. The types of IT included both in-house and third-party technologies, such as integrated development environments, programming languages, database management tools, messaging software, bug trackers, and testing tools. The time frame of the critical incidents ranged from acute bursts of stress lasting for a few hours to prolonged stressful periods lasting for multiple weeks. Additional details regarding the date, duration, and strain caused by the incidents can be found in Table C1 in Appendix C.

We connected the CSs to 10 of the 12 CFs introduced by Skinner et al. (2003). Many of the respondents also described how they used multiple CSs when coping with an incident. The following subsections introduce the CSs categorized under each CF in detail. In the subsections, we also go over the different contextual underlying factors identified as affecting the decision to adopt specific CSs. For data extracts demonstrating the identified CSs see Table D1 in Appendix D. For a summary of how the respondents described their coping efforts in the open-ended questions, consisting of the most common types of words associated with each identified CS, see Appendix E. An overview of the main results and how they relate to prior technostress and software engineering literature is presented in Table 4.

4.1. Coordinating actions and contingencies in the it use environment

4.1.1. Escape

CSs connected to the CF of escape received 227 mentions, the most out of any of the CFs. The most prevalent form of escape was to *temporarily disengage from the IT causing stress*. The forms of

Table 3
Chains of evidence demonstrating the results of the analysis process.

Demographics	Contextual underlying factors	CS	CF*
- Male - Age 51 - System specialist	*Need of approval and understanding* "… I needed to fix a system that a consult company had implemented wrong … the choice of technology just didn't work … I felt inadequate, like I wasn't enough, a lack of support…"	*Contact seeking* "…I discussed the experience with other experts, my supervisor, and the project manager. They were mostly understanding."	*Support seeking* Seeking social, emotional, or instrumental support from social resources.
- Female - Age 31 - Design engineer	*Lack of ideas or resources to use other CSs* "I had no idea how to make the code do what I wanted. … I had no experience with the programming language, I couldn't ask for advice, and the bug was critical."	*Emotion control* "I tried to calm myself and breath to relax my body…"	*Self-reliance* Self-control performed by an individual.
		Cognitive avoidance "… in my free time, I tried actively not to think about the experience."	*Escape* Efforts to disengage or escape from a stressful situation.

*Definitions based on Skinner et al. (2003), Skinner and Zimmer-Gembeck (2007), and Duhachek and Oakley (2007).

Table 4
Overview of the results.

Adaptive process	CFs and their definitions* (total number of mentions)	Identified CSs (number of mentions)	Underlying factors affecting the adoption of CSs
Coordinating actions and contingencies in the IT use environment	**Escape** (227) *Efforts to disengage or escape from a stressful situation*	Temporary disengagement from IT use (96) Behavioral disengagement (79) Cognitive disengagement (20) Substance use (8)	- Temporary exhaustion and awareness of technostress build-up - Accepting organizational culture regarding (techno)stress, allowing one to disengage - Freedom to arrange one's work to disengage - Lack of motivation to engage the incident with IT
		IT avoidance (16) Changing jobs (8)	- Prolonged technostress and exhaustion - Lack of or depleted resources to cope with repeating or prolonged technostress - Unbearable IT
	Problem-solving (208) *Adjusting one's own actions to be effective in a stressful environment*	Adjusting IT use routines or habits (47)	- Motivation to change one's harmful IT use routines - Feeling forced into changing one's IT use routines because of, e.g., incompatibility between IT
		Fixing the IT (47) Using workarounds (46) IT switching (23)	- Technical know-how - Confidence, pressure, or frustration motivating one to solve the problem with IT - Freedom to choose one's IT
		Planning and strategizing (45)	- Past experiences and knowledge regarding issues with IT - Desire to boost one's confidence to overcome future incidents with IT
	Information seeking (110) *Taking action to learn more about a stressful situation*	Asking for technical support (59) IT use training (51)	- Unfamiliarity of the issues with the IT - Personal motivation to learn and improve one's IT use skills - Insecurity and doubt caused by insufficient IT use skills and/or training
	Helplessness (16) *Actions (or inactions) caused by a lack of resources to cope with the situation*	Giving up* (16)	- Lack of recourses to cope with the IT use incident - Lack of control over IT - Feeling obliged to continue with harmful IT use practices
Coordinating reliance and social recourses in the IT use environment	**Support seeking** (74) *Seeking social, emotional, or instrumental support from social resources*	Comfort or contact seeking (53) Social escape* (17) Therapy* (4)	- Need of encouragement, approval, compassion, or understanding with IT use - Accepting organizational culture regarding (techno)stress - Lack of personal resources - Prolonged or repeated technostress
	Self-reliance (57) *Self-control performed by an individual to protect their available resources*	Emotion control* (37) Positive thinking (10) Meditation/Mindfulness (10)	- Personality and generally positive outlook on using IT - Panic and/or lack of resources or ideas to come up with other CSs in the moment
	Delegation (5) *Behavior and emotions generally regarded as maladaptive*	Complaining and venting (5)	- Social setting - Shared frustration and incidents with IT
Coordinating preferences and options in the IT use environment	**Negotiation** (55) *Finding new options for fitting together one's personal preferences and constraints imposed by stressful situations*	Compromising with IT* (33) Persuasion to change IT or IT use practices* (22)	- Acute issues with IT - Long-term relief valued over short-term burden - Freedom to influence one's ways of using IT - Limited possibilities to influence IT choices or harmful IT use practices alone
	Accommodation (50) *Adjusting personal preferences to available options (or constraints) in a stressful situation*	Acceptance and toleration of stress-inducing IT* (39)	- Heavy reliance on IT causing stress - Seeing IT-related issues and stress as inevitable
		Cognitive restructuring (11)	- Confidence in one's IT use skills - Motivation to learn from an incident
	Opposition (15) *Externalizing the reasons for stress on others or engaging in aggressive behavior*	Blaming others* (10) Blaming IT (5)	- Frustration and anger caused by the incident - Inability to influence IT choices or harmful IT use practices - Self-image and professional pride

* CS previously unidentified in the software engineering (Ebert et al., 2019; Russo et al., 2021; Taipalus et al., 2020; Weerasekara et al., 2022) or technostress (Fuglseth and Sørebø, 2014; Gaudioso et al., 2017; Ioannou et al., 2022; Pirkkalainen et al., 2019; Salo et al., 2017, 2020; Tarafdar et al., 2020; Zhao et al., 2020) literature.

[a] Definitions based on Skinner et al. (2003), Skinner and Zimmer-Gembeck (2007), and Duhachek and Oakley (2007).

disengagement included, for example, taking short breaks from using troubling IT (e.g., development tools), temporarily switching away from a complex development task, and taking days off. Other strategies for temporary disengagement included *behavioral disengagement* (e.g., "playing the piano and video games at home"), *cognitive disengagement* (e.g., "actively avoiding thinking about the issue"), and *substance use* ("…drinking and smoking"). The most common personal underlying factors for using a form of temporary disengagement were raising stress levels and temporary bursts of exhaustion. The respondents described how they saw disengaging as beneficial to "*come back with a fresh state of mind.*" Organizational factors that helped with disengagement efforts were an accepting atmosphere regarding stress, being allowed or even encouraged to take breaks in case of exhaustion, and having a flexible work schedule. These factors made it easier for the respondents to control their stress levels and avoid prolonged exhaustion. Interestingly, the respondents mentioned that the lack of personal motivation and not feeling invested in the issue also made it easier to disengage rather than, for example, taking a more active approach and trying to solve the stressful issue.

The respondents also described acts of avoidance that exceeded temporal attempts to escape. Long-lasting technostress and exhaustion, unbearable IT tools, and the depletion of resources when dealing with constant stress were described as factors that pushed the respondents toward engaging in more prolonged strategies for escape. For example, one of the respondents described how they found the user experience of a specific debugger so unbearable that, in the end, they refused to use that tool again (*IT avoidance*). Others also decided to *change jobs* to work with tools and technologies that they found more enjoyable or at least less taxing.

4.1.2. Problem-solving

CSs connected to problem-solving were mentioned by 208 respondents. The respondents mentioned direct problem-solving efforts to tackle the problem with the IT by *fixing the IT*, going around the issue by *using workarounds*, and *switching the IT* to avoid the issue. The personal underlying factors for choosing these CSs included technical know-how and confidence to address the problem directly via, for example, fixing bugs or coding workarounds to deal with the issue. Regarding IT switching and using workarounds, it is also important to mention that these strategies required freedom and control over the IT to make the switch to using a different, preferred technology. Those respondents who worked with, for example, client software or had strict limitations regarding the tools they were allowed to use, usually did not have the freedom to make these kinds of switches. Interestingly, it also became evident in the responses that the employees did not always voluntarily engage in problem-solving efforts. Instead, it was described how feeling frustrated and angry about an IT issue or being forced into fixing one "*because no one else was else was there to deal with it*" can also work as factors for choosing problem-solving as a way of coping.

One CS related to problem-solving was to adjust IT use routines or habits. The most prevalent factor for doing so was personal motivation to change harmful IT use routines (e.g., multitasking) because the respondents felt like they had created the stress themselves by adopting these routines. Another factor for changing one's IT use routines was being forced into doing so because of either incompatibility between IT or one's preferred ways of working (e.g., not being able to implement a solution in a preferred way because of legacy issues).

The respondents also described how they took a more proactive approach to problem-solving by coping with *planning and strategizing*. Planning and strategizing included planning the relevant actions or steps for the future use of the IT (e.g., by writing notes and instructions on how to use a specific software or what to do when learning a new tool the next time). The factors for adopting such an approach included the personal desire to gain confidence and be more prepared for dealing with similar incidents in the future. The respondents also described how their expertise and experience with past incidents made it easier to plan and figure out what to do when encountering similar incidents, such as unoptimized code or difficult-to-use software.

4.1.3. Information seeking

In total, 110 respondents mentioned CSs connected to CF information seeking, including engaging in *IT use training* and *asking for dedicated or peer technical support*. The factors underlying the adoption of these CSs included either the respondents feeling forced to use these CSs because they lacked the knowledge to resolve the incident (e.g., working with unfamiliar programming languages) or the respondents' personal motivation to learn and "*improve their skills as developers.*"

The types of training varied from formal training offered by the organizations during working hours to self-training carried out by the respondents themselves using various resources, such as (video) guides and internet forums. Respondents who had to rely on self-training to cope with an incident also described the lack of learning resources and training given by the organizations as a factor for engaging in self-training. These respondents described how they felt that the organization was not interested in training them, which resulted in them deciding to learn the development tools they had to use on their own.

Asking for technical support as a CS was usually related to one-off stressful incidents with IT. The reason for using technical support often described by respondents was that they had no idea how a sudden issue (e.g., a bug in the code or a technical malfunction) could be solved and, thus, decided to rely on technical help to aid with the task. Technical support and different forms of training were also often used together to gain a more holistic understanding of common problems with the IT and gain confidence in solving them to avoid stressful incidents moving forward.

4.1.4. Helplessness

Sixteen respondents mentioned coping by *giving up,* which were connected to CF helplessness. These respondents described how they felt obligated to adapt to stress-inducing IT use practices and saw no way out. Other factors for giving up, as put by the respondents, were the lack of necessary resources (e.g., time, skills, experience) to cope with the situation ("*I didn't have any chance (time) for that (to cope) … I didn't get one hour of sleep*") and having no control over the situation ("*it was something that was beyond my control…and there is nothing I can do about it*"). These factors made the situation feel hopeless and unmanageable, leading to inactivity and unfortunate consequences, such as cognitive exhaustion and self-doubt.

4.2. Coordinating reliance and social resources in the it use environment

4.2.1. Support seeking

Seventy-four respondents mentioned CSs connected the CF of support seeking. The majority of the CSs related to support seeking described *seeking* either *social contact or comfort* from those at the workplace or outside of work, such as family and friends. Seeking social comfort from friends and family usually stemmed from factors related to the need to seek compassion or guidance in a stressful situation (e.g., when personal resources were running out). The most prevalent factor for seeking social contact in the workplace was the need for peer support. For example, many of those struggling with technical problems or troublesome development tools felt like other developers would be the most likely to understand and accept their struggles and encourage them to keep pushing forward toward a solution. An accepting organizational culture regarding stress was also a relevant factor here because it made it easier to talk about one's burden and seek peer support and understanding. Interestingly, social contact was also used as a way of *social escape* by engaging in casual conversations or activities with colleagues to take a step away from the stressful IT use environment.

The respondents also mentioned *seeking aid from a therapist*. These respondents described prolonged stressful incidents in which IT was one of the components causing stress alongside other issues at the

workplace, such as time pressure, high demands, and work overload. These respondents felt that their personal resources were not enough to deal with the demands caused by the incidents, so they turned to a professional to find new ways to manage and cope with the situation.

4.2.2. Self-reliance

Fifty-seven respondents mentioned CSs related to the CF of self-reliance, which included *emotion control* and *positive thinking*. These CSs focused on the individual's personal efforts to either control their negative emotions (e.g., frustration, anger, or nervousness) or focus on positive thoughts and emotions instead (e.g., confidence, hope, or trust). More specific and organized strategies or routines to control one's emotions were also mentioned, such as *mindfulness* and *meditation*. Those who used these CSs described the factors for using them as, for example, how it was in their general outlook on life and personality to stay confident and positive, no matter the situation. These respondents described how negative thoughts and emotions would only blur their thinking, so it would be advantageous to fight off these emotions and not let them take over. However, for some of the respondents, the factors for using these CSs were simply that staying optimistic and calm was the best that they could come up with in that situation to not fall into panic and despair.

4.2.3. Delegation

Five respondents described CSs connected to the CF of delegation. CSs for delegation included *complaining* about the situation and *venting* off emotions with others in a social setting. These respondents described how they would hold meetings (scheduled or informal, such as coffee breaks) with their coworkers to rant about difficult development tasks or software that was not behaving as intended. Interestingly, the main factor for employing these CSs was the possibility to engage with others in a social setting and complain about issues together with peers (compare with social escape in which the goal was to engage with others but not talk about the incident).

4.3. Coordinating preferences and options in the it use environment

4.3.1. Negotiation

Fifty-five respondents mentioned CSs connected to the CF of negotiation, which included *compromising* and *persuasion*. Compromising was used to allocate possible alternate courses of action to alleviate the pressure caused by IT use. In other words, the respondents were compromising with themselves by, for example, finishing a specific development task so that they would not have to stress about it again or using inferior (unpreferred) tools, even if it meant more labor in the short term to fix an IT issue. Another form of compromising included setting boundary conditions to work with stress-inducing IT (e.g., "*I won't work with legacy software unless it has good documentation and support*"). The identified factors for choosing compromising as a CS were the pressure to fix problems that required the respondents' immediate attention (e.g., severe technical problems), valuing long-term relief over short-term burden, and the freedom to influence one's ways of working (e.g., by setting boundary conditions for using certain types of IT).

The other CS connected to negotiation included persuading others by suggesting changes to technologies (e.g., development tools) or general IT use practices (e.g., how many developers review code) that would alleviate the stress caused by these aspects. The main factor for utilizing persuasion as a CS was described as not having control over the aspects of IT use causing stress or not being able to force such changes alone because of company policies or contracts. Persuasion was usually performed at a team level or by communicating one's suggestions to management. Another factor for persuasion was also one's personal motivation to make the issues visible to their peers or higher-ups to have a chance to influence change.

4.3.2. Accommodation

Fifty respondents mentioned CSs connected to the CF of accommodation. The CSs linked with accommodation included *accepting issues with IT* and *cognitive restructuring*. The heavy reliance on IT and frequent IT-related problems made some of the respondents adopt a mindset of acceptance because they saw issues with IT as "*an inevitable consequence of working in these kinds of jobs.*" Basically, they saw IT problems as something that they could not solve and would just have to accept and learn to live with. Another factor for adopting acceptance as a CS was that it was described as easing the long-lasting or continuous frustration caused by, for example, changing and troubling development tools. In case issues with IT could not be avoided or fixed, the respondents thought that it was better to learn to tolerate these issues and push through them rather than to remain in a constant fight against them.

Cognitive restructuring as a CS focused on the respondents altering their own conception and meaning that they had given to the stressful experience caused by IT use. For example, when encountering a difficult or confusing development task, the respondents aimed to think about the experience as a challenge rather than as a stressful problem that they could not overcome. The factors leading to engaging in cognitive restructuring were the confidence that these respondents had in their skills to overcome the problems with the IT or at least their motivation to learn from the experience and improve their skills, even if a solution could not be found.

4.3.3. Opposition

Fifteen respondents mentioned using CSs connected to the CF of opposition when coping with technostress. They sought justification for their stress and issues with IT, such as buggy code or usability issues, by either *blaming others* (e.g., supervisors, colleagues, or IT providers) or *blaming the IT itself*. These respondents described reflecting on the experience and rationalizing the reasons for which the cause for the stress could not be put on them. The factors encouraging the use of these CSs seemed to be strong negative emotions (e.g., anger and frustration) caused by the incident and not being able to control the source of the stress (e.g., dealing with a technical problem that one cannot solve), leading the respondents to seek justification for their stress from elsewhere. Interestingly, another identified factor was that the respondents seemed to feel vulnerable about not being able to solve an issue with IT, which made them defensive, perhaps to protect their self-image and professional pride. The number of mentions received by each CS and the distribution of mentions between CFs are summarized in Fig. 1.

5. Discussion

The goal of our study was to examine how software industry employees cope with technostress in their work. We identified CSs used by employees and categorized them into higher-level CFs (Skinner et al., 2003; Skinner and Zimmer-Gembeck, 2007) to obtain a comprehensive understanding of the different CSs used by the employees. By analyzing the underlying contextual factors affecting CS uptake, we were also able to provide novel insights into the factors guiding CS selection in the industry.

5.1. Research contributions and prior work

First, we contribute to the literature by providing a more comprehensive understanding and categorization of the CSs used by software industry employees. Prior research on coping (with stress) in the software industry has focused on the use of specific CSs (e.g., substance use in Newman et al., 2023), coping with specific demands of the software work (e.g., code reviews in Ebert et al., 2019), or using pre-determined coping scales to identify individual CSs (e.g., Brief COPE scale in Russo et al., 2021). Although these studies have provided valuable insights into the CSs used in the industry, our study provides a more comprehensive understanding of the CSs used via the 27 CSs and grounding

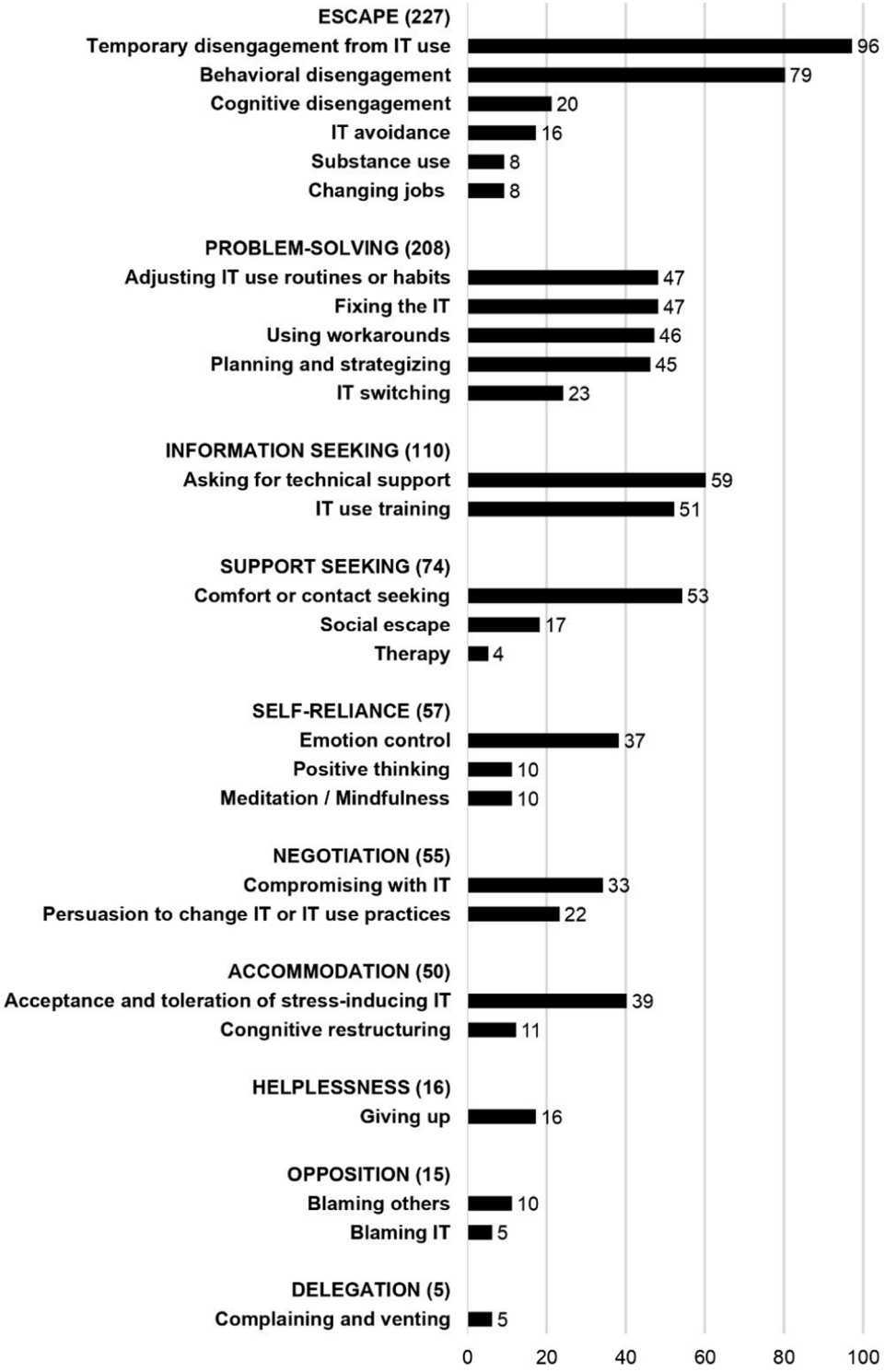

Fig. 1. Distribution of mentions per CS and CF.

these CSs into the actual real-life accounts of those working in the industry. This also enabled us to reveal CSs that have not been identified in prior software engineering or technostress research (e.g., social escape, persuasion, and compromising). By categorizing the CSs into the CFs proposed by Skinner et al. (2003), we were also able identify the most common core types of CSs and their functions used by software industry employees (i.e., escape, problem-solving, and information seeking). Overall, our findings show the great variety of coping efforts that employees must engage in to deal with problems caused by IT use and the resulting technostress. These findings together with prior research explain how employees in the industry are willing to go great lengths when dealing with the stressful experiences that they encounter in their work (e.g., Ebert et al., 2019; Taipalus et al., 2020). Employees are ready to, for example, seek support from their personal contacts, solve IT problems, compromise with how they work, remain positive, and even accept the inevitable stress brought by IT use to work in the industry. On the other hand, as also evidenced in prior works (e.g., Graziotin et al., 2018; Suárez and Vizcaíno, 2024; Wong et al., 2023), demands (in our

case demands caused by IT use) exceeding the limits of employees can become harmful, especially for their well-being and productivity. Our results highlight these aspects by showing that, despite the tremendous efforts put into coping by employees, prolonged and repeating technostress caused by negligence over IT-related problems can also drive employees into giving up, complaining, and blaming others, which are often considered as negative or maladaptive approaches to coping (Skinner and Zimmer-Gembeck, 2016). Although these CSs can be effective for the employee, they might have a negative impact on others. In addition, different forms of disengagement received a surprising number of mentions, escape being the most mentioned CF. Again, disengagement can be a valuable tool for recovering from temporary bursts of exhaustion and lower stress (Salo et al., 2017), but prolonged disengagement or having to constantly avoid stress-inducing IT could hint at the fact that some employees are finding it difficult to find other CSs and adapt to the ever-changing and complex IT use environment of the software industry (Chilton et al., 2005; Wong et al., 2023). Furthermore, by utilizing the CF categorization by Skinner et al. (2003), we were able to provide new insights into the functions of different CSs and why they are used to cope with technostress. For example, we show how CSs often regarded as purely problem focused (e.g., problem-solving or information seeking) are also guided by the emotions (Skinner and Zimmer-Gembeck, 2016) of IT users, such as feeling motivated, confident, or forced to tackle the problem or learn more about it. On the other hand, CSs generally regarded as emotion-focused (e.g., support seeking or positive thinking) can also be guided by one's eventual goal to act and solve an incident (Skinner and Zimmer-Gembeck, 2016) through emotional support, encouragement, or calmness. Thus, our results also widen the popular lens (e.g., Salo et al., 2020; Zhao et al., 2020) for categorizing coping with technostress into rather narrow categories (e.g., problem-focused and emotion-focused coping).

Second, our study reveals the contextual organizational and personal underlying factors that affect employees' CS selection (Matheny et al., 1993; Oakland and Oastell, 1996; Van den Brande et al., 2020). These results are valuable in understanding the types of resources that employees need in their personal coping efforts, how they can be supported, and how the lack of these resources might affect employees' coping decisions. The identified organizational factors that make personal coping efforts (e.g., information seeking, problem-solving, compromising, disengagement, and support seeking) easier to employ include the availability of peer support, technical support, IT training, accepting organizational atmosphere regarding stress, autonomy to arrange one's work, and control over IT. Many of these (or similar) factors, such as social contact (Russo et al., 2021), IT control (Pirkkalainen et al., 2019), IT support (Ragu-Nathan et al., 2008), and inclusiveness (Trinkenreich et al., 2023), have also been noted in prior research to improve employees' well-being. Regarding coping, these factors would help employees by making sure that they have more CSs available, and they do not have to (solely) rely on their personal resources when coping. On the other hand, our findings also reveal the organizational factors that hinder or limit employees' CS selection. Examples of these factors include having no control over harmful IT use practices, not being able to change one's IT, unfamiliar, acute, or repeated problems with IT, and insufficient peer or technical support structures. These factors force employees to rely more on their personal (emotional and creative) capacity to manage technostress and cope by, for example, driving for change in the IT use practices, coming up with workarounds, changing one's perspective over the incident, remaining positive, or accepting the situation. However, if the situational demands become too high and personal resources are drained, it could lead to employees coping by, for example, changing jobs, taking extended breaks from work, blaming others, and suffering from harmful outcomes, such as exhaustion and hopelessness. These findings are alarming because prolonged stress and ineffective coping can lead to a plethora of harmful outcomes (e.g., burnout, depression, and employee turnover), as evidenced by prior research on technostress (Reinecke et al., 2017; Tarafdar et al., 2010, 2019) and stress in the industry (Graziotin et al., 2018; Suárez and Vizcaíno, 2024; Trinkenreich et al., 2023). Thus, it is imperative that employees' coping efforts are effectively supported. Personal factors that make CSs easier to employ include, for example, technical know-how, expertise, motivation, confidence, social structures, and a positive outlook on difficulties. Personal factors, such as strong emotional reactions to stress (e.g., anger and frustration) and lack of knowledge, confidence, or motivation, can hinder the coping efforts. Mirroring these findings with prior research, the belief and confidence that one's coping will be effective (coping self-efficacy) is an important aspect of coping (Cheng et al., 2014; Salo et al., 2020). Thus, although these factors are personal, it would still be important and beneficial to strengthen the know-how and stress management skills of employees to boost their confidence and self-esteem. Possible interventions to accomplish these aspects could include training, endorsing a positive inclusive atmosphere, and encouragement to continuous learning (e.g., Suárez and Vizcaíno, 2024; Trinkenreich et al., 2023). These viewpoints on coping with technostress contribute to prior research by painting a more detailed picture of the selection of CSs when coping with technostress.

Finally, a detailed understanding of the CSs and underlying factors provides a needed baseline for designing organizational support mechanisms (discussed further in practical implications) for supporting the coping efforts of those working in the software industry. Hence, we contribute to the current software engineering (e.g., Trinkenreich et al., 2023) and technostress research (e.g., Fuglseth and Sørebø, 2014) by extending the mechanisms suggested in prior research and aligning our suggestions to the needs of those working in the industry. Our results also show that commonly suggested organizational support mechanisms are not utilized optimally. For example, mechanisms such as IT support and training (Fuglseth and Sørebø, 2014; Ragu-Nathan et al., 2008; Tarafdar et al., 2015) might be in place, but it seems that employees are often not given the time to engage with these mechanisms or that these mechanisms are not available because of overly long response times. Furthermore, many of the respondents said that they did not have any of these organizational support mechanisms available if they encountered IT-related issues. This is alarming because, as demonstrated, the combination of limited organizational support for coping (e.g., training, control, and autonomy) and depleted personal resources can hinder the coping efforts of employees and lead to employees feeling helpless and exhausted. The main contributions and results of our study are

Table 5
Summary of the main contributions and results.

Contribution 1: A comprehensive categorization of CSs	- Identifying and explaining 27 CSs used by software industry employees to cope with stressful IT use incidents. - Identifying eight novel CSs for coping with technostress. - Identifying and explaining the core CSs used in the software industry (per the CFs by Skinner et al. (2003)) and why they are used.
Contribution 2: Contextual factors affecting CS selection	- Identifying and explaining the personal and organizational contextual underlying factors that affect CS selection in the software industry. - Explaining how these underlying factors can either support or hinder the selection of CSs.
Contribution 3: Organizational support mechanisms for coping	- Identifying six potential organizational mechanisms to support the coping efforts of employees: IT control, autonomy, social support, IT support, IT training, and (techno) stress education. - Integrating the potential organizational support mechanisms with the coping needs of software industry employees to remove hindrances from CS selection.

summarized in Table 5.

5.2. Practical implications

Drawing from the identified CSs and underlying factors, we offer suggestions for interventions that software organizations could use to support the coping efforts (and well-being) of their employees. First, software organizations should ensure that their employees feel **a sense of control over IT** (Pirkkalainen et al., 2019) and how IT is being used. Employees and development teams should be given the opportunity (when possible) to decide on the tools (e.g., programming languages or development frameworks) and IT use practices that best fit their ways of working and know-how. Our suggestion is also backed up by previous studies showing how developers feeling a sense of control during development can boost their productivity and self-confidence (e.g., Graziotin et al., 2015; Mäntylä et al., 2016). In practice, organizations could arrange feedback sessions or collect feedback through questionnaires or interviews regarding the possible problems with the IT or regarding employees' wishes toward IT tools that they would like to use. Based on our results, control over IT would give employees more possibilities for coping by encouraging them to engage in problem-solving (taking action), information seeking (learning more), and bringing forth possible issues with IT (finding new options) to move away from harmful IT use practices. However, in many cases, giving more control might not be feasible (e.g., the monetary investment is too high, or a client requires the use of specific software or programming language). In this case, it would be important to make the situation as temporary as possible and clearly communicate the reasons for the lack of control to the employees in advance. Employees should also be given enough time and ongoing support to learn the IT being used (e.g., introduce dedicated or peer technical support and IT use training). These interventions could make it easier for employees to manage limited control over IT by making sure that they have the know-how and expertise to use the technology, boosting confidence and perceived control and limiting uncertainty and complexity, which are prominent causes for technostress (Tarafdar et al., 2019). By limiting uncertainty and increasing control, employees might also be more willing to restructure their thinking (e.g., face the situation as a challenge or as a learning experience) and adopt a positive mindset to avoid stress escalation (Pirkkalainen et al., 2019).

Second, employees should be given more **autonomy and flexibility** to arrange and plan their work to take time for both reactive and proactive coping efforts. Regarding reactive coping, autonomy is important because technostress is often caused by unexpected or surprising incidents, such as systems going down, encountering bugs, and having to solve acute IT-related problems (Siitonen et al., 2022). Many of the respondents described it as crucial that they had the flexibility to, for example, take minibreaks (disengage), ease their frustration (control their emotions), or rearrange their schedule (compromise) to react to and manage an acute stressful incident. Regarding proactive coping, autonomy is important because it gives employees the possibility to learn from past incidents (seek information), plan their coping behavior (plan and strategize), and recover from prolonged stressful incidents (e.g., by disengaging or utilizing social support) to complement their resources to cope with future stressful incidents. Thus, supporting the coping efforts of employees is not only about the concrete mechanisms that organizations can offer (e.g., IT use training or technical support), but also about trusting the employees and giving them the autonomy to plan their work tasks and coping behavior to avoid accumulating stress and prolonged negative outcomes. Allowing for autonomy has also been found to be beneficial for the long-term productivity and well-being of employees in the industry (Russo et al., 2021).

Third, software industry organizations should ensure that their employees have the necessary **social (peer) support structures** in the workplace. This is crucial because social interactions and resources are vital assets for coping with technostress in the industry. Based on our results, CSs related to social support (e.g., comfort-seeking, social escape, and venting in a group environment) also commonly accompany other CSs. Peer support in the workplace seems to be an especially important factor because other social connections, such as friends and family, most likely do not share the same daily problems with IT that those working in the software industry do. The social support structures could include mentor or peer support groups, either at the workplace or in a virtual environment for those working remotely. Simultaneously, these support groups could foster a sense of inclusiveness and belonging, which have been found to be important factors for reducing burnout in the industry (Trinkenreich et al., 2023; Wong et al., 2023). Via these organizational structures, employees would have a legitimate place to discuss, for example, IT-related problems and challenging work tasks as well as offer peer support and solution suggestions to help with problem-solving and information seeking. Having these structures could also help in building an environment that supports open dialogue and bringing up stressful issues, making stress a more accepted phenomenon in the organization. In addition to helping with coping efforts, a positive atmosphere and social contacts in the workplace are important aspects in improving the employees' well-being and productivity in the software industry (Graziotin et al., 2018; Russo et al., 2021; Suárez and Vizcaíno, 2024).

Fourth, it would be important that concrete support mechanisms (**IT support and training**) would be available for employees and that they have the necessary resources to utilize these mechanisms. Despite the suggestions for these interventions (Fuglseth and Sørebø, 2014; Ragu--Nathan et al., 2008; Tarafdar et al., 2011), not having the necessary training or knowledge to use IT still seems to be a common issue among software professionals, causing stress, frustration, self-doubt, blame, and anger (Graziotin et al., 2018; Siitonen et al., 2022). Our results also support these findings by showing how being dissatisfied with the organizational training and support for IT use seems to affect employees' motivation to engage in personal coping efforts, such as fixing IT problems and self-learning, instead encouraging coping efforts connected to blaming others, helplessness, and acceptance. Proper IT support and training (especially when introducing new IT) would help employees cope with IT by helping them identify and understand common problems, work around these problems, improve their IT use skills, and avoid stress escalation and feelings of helplessness.

Fifth, it seems that not enough emphasis has been placed on the importance of **educational interventions** regarding the training of staff involving themes related to technostress and coping. There were no mentions of this kind of training in the questionnaire responses, despite some respondents expressing a desire for it, and how it has been found valuable in other occupational contexts (e.g., Paudel et al., 2022). Stress training could include mentoring and educating employees about what (techno)stress is, how it emerges in the software industry, and introducing possible CSs and organizational support mechanisms to support employees in their coping efforts. This type of training could help employees recognize CSs, identify potentially harmful IT use practices (e.g., multitasking), and engage in proactive coping efforts (e.g., planning, strategizing, and adjusting one's personal IT use practices) to avoid technostress escalation.

Finally, individual employees and teams can find our results valuable in improving their coping effectiveness. Becoming better aware of one's own coping efforts can help employees realize possible limitations in their coping behavior and make changes that could improve their coping effectiveness (Cheng et al., 2014). For example, in the software industry context, one could be more proactive in their coping by being prepared for encountering problems from past incidents, scheduling more time for stress recovery, making an effort to change harmful IT use practices (e.g., multitasking or checking work email outside of work hours), and considering adopting techniques that would help with managing emotions (e.g., mindfulness) when encountering IT problems. We hope that showcasing the variety of CSs used by those employed in the software industry can help employees reflect on their coping behavior, identify

the reasons for up-taking certain strategies over others, and find effective alternatives or additional CSs that they had not considered before. Simultaneously, we hope that our recommendations for organizational support mechanisms can help individual employees take the initiative to ask organizations for specific types of support and make arrangements inside development teams (Suárez and Vizcaíno, 2024) that would aid in the coping efforts of team members (e.g., improve peer support inside teams or make a shared conscious decision to give up a stressful IT use practice).

5.3. Limitations and threats to validity

As with any study, our study has its limitations. First, our study might be limited by the practical significance of our findings (Tracy, 2010). Coping is a process that manifests in the continuous interactions between the individual and the environment (Biggs et al., 2017; Cheng et al., 2014; Lazarus and Folkman, 1984; Skinner et al., 2003). In other words, coping is an active process during which different CSs could be activated and ineffective ones abandoned, making coping a difficult phenomenon to study in cross-sectional designs (Skinner et al., 2003). The current study has focused on software industry employees' initial CSs when dealing with technostress. Thus, we could not further investigate the subsequent appraisals of the incidents and whether the identified coping efforts were successful in the long term when coping with technostress, possibly limiting the applicability of our findings.

Second, our study might be limited by the resonance and transferability of our results (Tracy, 2010). Despite employing a rather large sample size of 715 respondents, coping is still a highly situational and subjective process as numerous CSs have been identified in the field of psychology (Lazarus and Folkman, 1984; Skinner et al., 2003). Thus, replicating the results of our study might be challenging in different contexts or with different samples because the CSs might change depending on the circumstances. However, our findings are supported by the fact that they reflect many CSs that have been recognized in the literature on coping (Skinner et al., 2003). In addition, sorting the CSs into the CFs makes our findings easier to compare, as the CFs provide a way to categorize contextualized CSs under the same core categories (Skinner and Zimmer-Gembeck, 2016). We have aimed to ensure the replicability of our study by giving adequate details for future research regarding the data collection and analysis performed in our study via, for example, providing our data collection instrument (Appendix A) and coding scheme (Appendix B) used in the analysis. We have also provided an online replication package that includes our data collection instrument, final sample demographics, data exclusion criteria, coding scheme (e.g., code descriptions and examples), and analysis results to further enhance the replicability of our results (see https://doi.org/10.5281/zenodo.14180103).

Third, our study might be limited by the credibility of our data (Tracy, 2010). Reliance on self-reported data could introduce biases, such as recall bias or social desirability bias. However, regarding recall bias, CSs are difficult to study other than retrospectively because individuals typically have better abilities to reflect on their coping behaviors *after* a stressful experience (Biggs et al., 2017; Lazarus and Folkman, 1984). Still, our data might be subject to inaccuracies caused by recall bias, such as respondents misremembering details regarding their coping efforts or reinterpreting the past incident as more or less stressful than it was at the time of the incident. We have tried to counter this limitation by instructing the respondents to take their time when answering the questionnaire and reporting on their incidents in as much detail as possible. Regarding social desirability bias, stressful experiences can evoke strong emotions, such as guilt and embarrassment, and can be difficult to report on. The respondents might also be wary of the possible negative consequences of reporting on negative aspects or experiences about their work, compromising the authenticity of our data. We have aimed to counter this limitation by ensuring the full anonymity of the respondents in the instructions to our questionnaire and in the privacy notice obtained by the respondents.

Finally, our study might be limited by the rigor of our data collection process (Tracy, 2010). The use of online research platforms such as Prolific could expose our study to the so-called "super users" who pay little attention to the tasks on the platform (Lowry et al., 2016). We have aimed to counter this limitation by following the suggestions given by Prolific to obtain high-quality data (using attention check questions and limiting the pool of respondents based on their acceptance rate (97 %)). We also only included responses from respondents who had at least 20 prior responses on the platform and provided sufficient detail in their responses.

5.4. Future research

Our insights into coping with technostress in the industry also open interesting avenues for future research. First, our study does not directly address the effectiveness of the different CSs (Oakland and Ostell, 1996). Thus, future research should investigate whether there are differences between the effectiveness of CSs when encountering technostress and whether employees find the utilized CSs beneficial in terms of adaptation and well-being (typical criteria for coping efficacy). Hence, we need more longitudinal and intervention studies on this topic. Furthermore, coping self-efficacy (a person's belief that coping efforts are effective) is a critical factor in assessing the effectiveness of coping (Cheng et al., 2014), which deserves more attention in the technostress literature.

Second, it is important to determine whether the core CSs remain similar when employees cope with other stressful aspects of their work (e.g., time constraints, steep learning curves, teamwork issues, or changing requirements) (Skinner and Zimmer-Gembeck, 2016). The CFs provide a plausible framework for this task (Skinner et al., 2003). This is crucial because differences between common coping efforts between contexts could suggest that organizational support measures should be adjusted accordingly. Therefore, we recommend that future research investigate whether our findings are replicable in other coping contexts within the software industry.

Third, we suggest that future research investigate the effectiveness of the presented organizational support mechanisms in supporting employees' coping efforts. These efforts might also reveal possible support mechanisms left unaddressed in our study. As of now, it seems that employees might not receive the support and resources that they would like to have in employing their CSs, often making the coping process largely based on one's personal resources.

Finally, with the swift proliferation of development tools utilizing artificial intelligence and language models, we ponder whether these tools could help combat the harmful effects of technostress (Wong et al., 2023). We suggest that future research should investigate whether these technologies could be utilized to support the coping efforts of employees by, for example, reminding them to take breaks from using IT, giving quick and effective support on problem-solving efforts, helping with information seeking, or suggesting possible ways to cope with an incident.

6. Conclusion

Technostress is an unfortunate consequence of the IT-heavy work environment in the software industry. It is crucial to continue to conduct research on how the coping efforts of individual employees and organizational support mechanisms could work together to mitigate this harmful phenomenon. Our study provides a wide view of the CSs used by software industry employees, along with how the different underlying factors guide the adoption of the CSs that employees use. We hope that raising awareness of the harmful effects of technostress, strategies for coping with them, and the current shortcomings of organizational support regarding coping efforts can help improve the well-being, work enjoyment, and productivity of those working in the rapidly changing environment of the software industry.

Funding Sources

This work was supported by the Finnish Foundation for Economic Education; the Research Council of Finland [grant number 341359, 346479]; the Emil Aaltonen Foundation. The funding sources had no involvement in the design, collection, analysis or interpretation of data, writing of the report, or decision to submit the article for publication.

CRediT authorship contribution statement

Valtteri Siitonen: Conceptualization, Methodology, Formal analysis, Investigation, Writing – original draft. **Saima Ritonummi:** Conceptualization, Methodology, Formal analysis, Investigation, Writing – review & editing. **Markus Salo:** Conceptualization, Methodology, Investigation, Writing – review & editing, Project administration, Funding acquisition. **Henri Pirkkalainen:** Conceptualization, Writing – review & editing. **Saija Mauno:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We would like to thank Iida-Maria Tiitinen for her help with writing and proofreading, as well as the area editor and the three anonymous reviewers for their valuable and constructive comments.

Appendix A. Questionnaire Outline Regarding the Questions about the Critical Incidents and Coping [with the authors' additional notes]

[The outline of the questionnaire is presented regarding the parts relevant in the current study. The questionnaire also included other questions, which are not presented here.]

Questionnaire outline:

In this questionnaire, we ask you to describe one exceptionally burdensome/stressful experience related to the use of technology or software in your work. Experiences can be related to, for example, using a work computer, phone, or software in your work tasks.

The questionnaire takes about 15 min to complete.

The answers will be treated confidentially. You can read the privacy policy here *[link]*.

Thank you for your participation!

Background information [*the background information of the respondents included their Age, Nationality, Gender, Employment Status, Job title, Work Experience in Current Role, and other questions not relevant here* e.g.*, their Prolific ID. All questions included an option: "I prefer not to disclose"]*

Burdensome/stressful experience

Please take a moment to recall an exceptionally burdensome/stressful experience related to the use of technology or software in your work. You can take a few minutes to recall. This time is allowed for in the duration of the questionnaire.

Technology and/or software that was involved in the experience: *[open question]*

Please describe the burdensome/stressful experience in as much detail as possible in your own words: *[open question]*

For how long did the experience last?

- Less than one hour
- 1–12 h
- 13–24 h
- 1–7 days
- 1–4 weeks
- Over one month
- I don't know

How long ago did the experience occur?

- Less than two weeks ago
- 2–4 weeks ago
- 1–3 months ago
- 4–12 months ago
- Over one year ago
- I don't know

How would you evaluate the strain caused by the experience?

- Very low
- Fairly low
- Moderate
- Fairly high
- Very high
- I don't know

How did you try to reduce or mitigate the effects of the burdensome/stressful experience? *[open question]*

[This is an attention check question] What is this writing task about? Please choose one of the following options:

- stressful experiences at work
- cooking
- studying abroad
- the meaning of life

Thank you for participating in the questionnaire!

Confirm your submission by using the link below:

[Using the link, the respondents were able to confirm in Prolific that they had answered the questionnaire to receive their monetary reward.]

Appendix B. Finalized Coding Scheme

Table B1
Finalized coding scheme and code descriptions.

CS (code label)	Code description	Code (label or description) informed by prior technostress or software engineering literature
Temporary disengagement from IT use	IT users taking short breaks and temporarily stepping away from stressful IT use incidents.	Label and description (Salo et al., 2017)
Behavioral disengagement (from IT use)	IT users engaging in other activities, such as playing sports or video games to disengage from stressful IT use incidents.	Label (Gaudioso et al., 2017; Russo et al., 2021)
Cognitive disengagement (from IT use)	IT users' attempts of actively avoiding thinking about or trying to forget stressful incidents caused by IT use.	Description (Salo et al., 2017)
Substance use	IT users consuming substances, such as alcohol or drugs to escape or manage the discomfort caused by stressful IT use incidents.	Label (Newman et al., 2023; Russo et al., 2021; Weerasekara et al., 2022)

(*continued on next page*)

Table B1 (*continued*)

CS (code label)	Code description	Code (label or description) informed by prior technostress or software engineering literature
IT avoidance	IT users actively avoiding and refusing to use IT that cause stressful incidents.	Description (Zhao et al., 2020)
Changing jobs	IT users changing jobs to avoid stressful IT use incidents.	Description (Fuglseth and Sørebø, 2014)
Adjusting IT use routines or habits	IT users modifying or changing IT use practices that they find are causing stressful IT use incidents.	Label and description (Salo et al., 2017, 2020)
Fixing the IT	IT users attempts to repair or fix the problem(s) with the IT causing stressful IT use incidents.	Label and description (Salo et al., 2020)
Using workarounds	IT users coming up with alternate ways and solutions to getting around the IT problem causing stressful IT use incidents.	Description (Salo et al., 2020)
IT switching	IT users shifting to a different, more efficient or less stressful, IT and replacing the current one with a substitute.	Label and description (Salo et al., 2020)
Planning and strategizing	IT users preparing and planning their courses of actions based on their prior knowledge, expertise, and experiences when faced with potentially stressful incidents with IT use.	Label (Gaudioso et al., 2017; Russo et al., 2021)
Asking for technical support	IT users requesting technical support to understand and/or fix the IT problem causing stressful IT use incidents.	Description (Gaudioso et al., 2017; Zhao et al., 2020; Ebert et al., 2019)
IT use training	IT users engaging in acquiring knowledge and skills through self-learning or formal training to solve stressful IT use incidents.	Description (Ebert et al., 2019; Tarafdar et al., 2020)
Giving up	IT users engaging in hopeless or helpless thinking and behavior and running out of resources when facing stressful IT use incidents.	
Comfort or contact seeking	IT users seeking social and emotional support from others (e.g., peers, friends, or family) to manage and discuss stressful IT use incidents.	Label (Russo et al., 2021; Weerasekara et al., 2022)
Social escape	IT users disengaging from stressful IT use incidents by engaging in unrelated activities and discussions with others.	
Therapy	IT users seeking help from mental health professionals to discuss the stressful IT use incidents and find ways for technostress mitigation and coping.	
Emotion control	IT users managing negative emotions to stay stable and calm when experiencing stressful IT use incidents.	
Positive thinking	IT users maintaining an optimistic mindset to focus on solving the IT problem and avoiding self-doubt when experiencing stressful IT use incidents.	Label and description (Tarafdar et al., 2020)
Meditation/ Mindfulness	IT users using the techniques of mindfulness and meditation to address the emotions, thoughts, and feelings caused by stressful IT use incidents.	Label and description (Ioannou et al., 2022)
Complaining and venting	IT users expressing their emotions, thoughts, and grievances regarding stressful IT use incidents either alone or with others.	Label and description (Pirkkalainen et al., 2019; Salo et al., 2020; Tarafdar et al., 2020)
Compromising with IT	IT users making trade-offs regarding their desired ways of using IT or the IT they use to solve or manage stressful IT incidents and achieve stress relief in the long-term.	
Persuasion to change IT or IT use practices	IT users advocating for modifications, improvements, or changes regarding the IT used or IT use practices causing stressful IT use incidents.	
Acceptance and toleration of stress-inducing IT	IT users acknowledging and coming to terms with the IT or IT use practices causing stressful IT use incidents.	
Cognitive restructuring	IT users changing their perspective or way of viewing stressful IT use incidents.	Description (Pirkkalainen et al., 2019; Taipalus et al., 2020)
Blaming others	IT users blaming others for causing stressful IT use incidents.	
Blaming IT	IT users blaming the IT for causing stressful IT use incidents.	Label and description (Salo et al., 2020)

Appendix C. Results of the Closed-Ended Questions

Table C1
Results of the closed-ended questions.

For how long did the experience last?	~% (n)	How long ago did the experience occur?	~% (n)	How would you evaluate the strain caused by the experience?	~% (n)
Less than one hour	6.2 % (44)	Less than two weeks ago	16.6 % (119)	Very high	38.7 % (277)
1–12 h	26.7 % (191)	2–4 weeks ago	14.1 % (101)	Fairly high	39.7 % (284)
13–24 h	8.7 % (62)	1–3 months ago	20.6 % (147)	Moderate	21.5 % (154)
1–7 days	22.7 % (162)	4–12 months ago	21.5 % (154)	Fairly low (EXCLUDED)	4.8 % (34)
1–4 weeks	15.8 % (113)	Over one year ago	26.3 % (188)	Very low (EXCLUDED)	1.1 % (8)
Over one month	18.3 % (131)	I don't know	0.8 % (6)	I don't know (EXCLUDED)	0.3 % (2)
I don't know	1.7 % (12)				

Appendix D. Examples from Data Demonstrating the Identified CSs

Table D1
Data examples demonstrating the identified CSs.

CS	Quote from data
Temporary disengagement from IT use	"… Internal development related to the (analytics software) interface should be done, and at the same time, support requests from the service desk software are coming in to be resolved. … It's a stressful situation when I need to advance 4–5 different things, all of which are urgent. … **I took a break and a moment outside.** After that, my thoughts were clearer." [Senior Developer]
Behavioral disengagement (from IT use)	"It felt like I was trying to save a sinking ship. Many others had already left the company, which caused technical debt and problems. I had to start fixing these issues under time pressure and with minimal prior experience regarding the tools (I had to use). … **I just spent time with my friends and played video games.** I want to stay away from projects that don't interest me." [Senior Software Developer]
Cognitive disengagement (from IT use)	"I had no idea how to make the code do what I wanted. … I had no experience of the programming language, I couldn't ask for advice, and the bug was critical. … **On my free time, I tried actively not to think about the experience.**" [Design Engineer]
Substance use	"I was supposed to develop a (specified) system for the staff in a month. … The actual task was a bit complex but manageable. The problem was having to use dated (development) frameworks that were not fit for the task. … **I literally sucked it up and drank a bit more on weekends just to forget."** [Intermediate Software developer]
IT avoidance	"The database had been accidentally wiped by an erroneous piece of software. The moment it was discovered it was all hands-on deck to mitigate the fallout of this database deletion. … **We banned that piece of software altogether."** [Software Engineer]
Changing jobs	"Recently, a client introduced me to (a messaging software) and I hated every minute of it. Different groups and chat rooms, employees having fun, I've never felt like such an outsider in my life. … **In the end, I left the agency and found a new client.** That agency, to this day, is still the only one I've worked for that's used that (messaging software)." [Job title not disclosed]
Adjusting IT use routines or habits	"… **Be mindful of ad hoc calls on communication software, prepare myself for work hours and not multitask, and de-clutter my PC and close all windows/tabs in browser.**" [Software Developer]
Fixing the IT	"We had an issue where this internal software developed a bug where it wouldn't update what co-workers were uploading or what they had done throughout the day. … Nobody had any clue what had been done. This led to some confusion because I was testing the same area as my co-worker … and you are wasting your time doing something that has already been done. **I helped figure out what the problem was and figure out a way to get it fixed."** [Software Test Engineer]
Using workarounds	"(We were) changing over to this CRM new system. The whole thing was implemented without any input from the teams and people who used it. When the switch was made, it was overnight without any training. Nobody knew how it worked. … **I implemented my own manual system until the new one was amended and up to scratch.**" [Project Manager]
IT switching	"I was trying to create a chatbot and the software allows you to do so almost entirely through a GUI. … Using that software was a terrible experience. … As of today, I still wonder if I just didn't know how to use it or if the things I wanted to implement simply couldn't be done because they are not supported. On top of that, the software has a bad UX, and it was hinting at some user actions (e.g., drag and drop) that weren't implemented. … **We simply ditched that software and moved to its SDK version that allowed for a more 'classical' programming experience."** [ML Engineer]
Planning and strategizing	"The (error reporting) software has several bugs. I have to update templates twice or it won't update the record. … Also, because of caching, sometimes I'd think that my update had no effect. … **I wrote a sheet of paper with the steps I have to take into account when working with that software. Every time I have to work on it, I read the paper before to have these concerns 'fresh' in my mind.**" [IT Consultant]
Asking for technical support	"Due to client restrictions, I was asked to perform a task using a certain software (I never use) rather than writing code in Python. … It took me three times longer and the final result was worse than I could have achieved with other methods. … **I asked colleagues for help** … **to master my way around the interface.**" [Senior AI Consultant]
IT use training	"When I first came to work in this company, I did not have the specific experience to address these (ERP) upgrades since I did not have the technology skills or background. My company did not bother to train me, so … **I had to learn all the procedures … on my own so that I wouldn't experience the same feeling again.**" [Job title not disclosed]
Giving up	"The software was very old with limited support and no peer help on the internet. A black box for almost everybody and I was admitted recently to the team. … **I didn't have any chance (time) for that (to cope)** … I didn't get one hour of sleep." [Job title not disclosed]
Comfort or contact seeking	"A customer requested to perform a data migration from an old CRM-system to a new one. … However, the data wasn't easily machine-readable and making the scripts took over six weeks. … **I discussed the issues themselves with peers, but also engaged in informal conversations.**" [Software Developer]
Social escape	"Our systems were struggling, and our servers were reaching 100 % of CPU usage. The app that we offer to our clients was unusable. The situation became much more stressful as my boss insisted that this is a database server configuration issue, and it should be resolved immediately. I'm the only Devops engineer in my team, and it was me who was supposed to actually fix stuff. … It's always something with networking, server configuration or whatever. It's never the badly written code! … **I was constantly chatting with my coworkers - mainly backend developers who were trying to help and we were sending some memes so that made the experience a little bit less stressful."** [Devops Engineer]

(*continued on next page*)

Table D1 (*continued*)

CS	Quote from data
Therapy	"I couldn't reach a solution to develop a webpage for a client. I had to work without a template and the design they gave me was difficult to replicate with the frameworks I was using. … **I'm currently going to therapy once a month. My therapist taught me some tools to help me relax and reduce the anxiety.**" [QA Tester]
Emotion control	"We were preparing for a rollout for more than six months. We had done our testing, resolved all bugs we could find, and examined different test cases. … Monday morning was a disaster. Nothing was going as expected. No orders were being processed, and around noon management decided we had to rollback the changes. … **I did try to stay calm, and calm my colleagues down,** but having to roll back work that took the better part of a year to implement within a day was nerve racking." [Senior Developer]
Positive thinking	"I had to implement a new technology in the game I was developing, the problem is that the performance on the game's target platform was terrible. … There is no documentation or help on the web for what I was doing. … I don't think I did anything in particular; **I usually try to stay positive**." [Developer]
Meditation/Mindfulness	"(Specific) tests were failing a lot. I was doing an investigation of the issues, and if possible, also fixes. But the tests were very poorly written, difficult to understand, and even more difficult to fix. They even were not isolated, so one test was affecting another. … **I meditate daily."** [Front-End Developer]
Complaining and venting	"New IT system did not come with all tools required, and with restricted user rights. … **I joined a session of fellow affected users and moaned about it until we reached catharsis.**" [Quality Manager]
Compromising with IT	"I accidentally deleted an important database property and had to rollback changes, but the change logging in the database system was very rudimentary. I had to piece together several information sources to fix my mistake. … **I worked on fixing the mistake ASAP. I found it out on a Sunday night but just worked on it because then I wouldn't have to stress about it (on Monday).**" [Product Manager]
Persuasion to change IT or IT use practices	"The challenge was integrating different pieces of code, worked on by multiple developers into a working final piece. … **I suggested that such work be shared among two developers who regularly integrate their code instead of doing it as one process toward the end.**" [Software Developer]
Acceptance and toleration of stress-inducing IT	"**Accepted, I couldn't do much (to a server crash) and just tried to keep on top of what I could** and ensure when we were back up, I was able to crack on with what I needed to do." [Administrator]
Cognitive restructuring	"The software didn't have much documentation, and some of it was outdated or incorrect. I tried to create a new data replication entity with this tool. … I thought to myself that nobody in the team knew how to do this either, so that relieved some pressure off my shoulders. …**I tried to look at it as a challenge, like a puzzle or a problem, instead of an anxious and complex task.**" [Developer]
Blaming others	"Meet deadlines, often without me doing the requirements assessment, and later, without help or little indication of how to solve the problems. … **Started thinking it wasn't my problem if seniors weren't interested in helping me.**" [Developer]
Blaming IT	"The operating system that we used in development decided to make changes to its settings on its own. … It was really difficult to figure out what had changed. … After looking into the software problem, **I decided that it's not the lack of my own expertise but that the operating system is not intended for software development. I continued working on the problem, but it could clearly be stated that the fault was in the tools, not in me.** It's also completely unreasonable that the organization I work for doesn't let us use tools that actually work." [Software Developer]

Appendix E. Tree Map of the Most Common Keywords Used by the Respondents

The tree map figure below displays the five most common words used by the respondents to describe their coping efforts, categorized by CS. To create the word listings, we grouped together all the data segments coded to represent each CS and analyzed each group for word frequencies. Common words with minimal semantic value, such as articles and conjunctions, were excluded from the final listings. Notably, only three words are listed under the CSs "therapy" and "complaining and venting" due to a lack of unique or frequently recurring words (Fig. E1).

Fig. E1. The five most mentioned keywords related to coping in the data categorized per CS.

Data availability

In the privacy notice of the study, the respondents were promised that only a specific group (the authors of the paper and specified individuals) would be able to access the data and written responses that the respondents provided. This was done to ensure the privacy of the respondents, because it would be difficult to guarantee complete anonymity for such written responses. Possible names of organizations and specific technologies mentioned by the respondents in the incident descriptions have also been removed from the article and its direct quotations to ensure the privacy of the respondents.

References

Beaudry, A., Pinsonneault, A., 2005. Understanding user responses to information technology: a coping model of user adaptation. MIS Q. 29 (3), 493–524. https://doi.org/10.2307/25148693.

Berg, B.L., 2004. Qualitative Research Methods For the Social Sciences, 5th ed. Pearson Education, Boston.

Biggs, A., Brough, P., Drummond, S., 2017. Lazarus and Folkman's psychological stress and coping theory. In: Cooper, C.L., Quick, J.C. (Eds.), The Handbook of Stress and Health. Wiley, New York, pp. 351–365. https://doi.org/10.1002/9781118993811.ch21.

Brod, C., 1984. Technostress: The Human Cost of the Computer Revolution. Addison Wesley Publishing Company, USA.

Cheng, C., Lau, H.P., Chan, M.P., 2014. Coping flexibility and psychological adjustment to stressful life changes: a meta-analytic review. Psychol. Bull. 140 (6), 1582–1607. https://doi.org/10.1037/a0037913.

Chilton, M.A., Hardgrave, B.C., Armstrong, D.J., 2005. Person-job cognitive style fit for software developers: the effect on strain and performance. J. Manag. Inf. Syst. 22 (2), 193–226. https://doi.org/10.1080/07421222.2005.11045849.

D'Arcy, J., Herath, T., Shoss, M.K., 2014. Understanding employee responses to stressful information security requirements: a coping perspective. J. Manag. Inf. Syst. 31 (2), 285–318. https://doi.org/10.2753/mis0742-1222310210.

DeLongis, A., Coyne, J.C., Dakof, G., Folkman, S., Lazarus, R.S., 1982. Relationship of daily hassles, uplifts, and major life events to health status. Health Psychol. 1 (2), 119–136. https://doi.org/10.1037/0278-6133.1.2.119.

Duhachek, A., Oakley, J.L., 2007. Mapping the hierarchical structure of coping: unifying empirical and theoretical perspectives. J. Consum. Psychol. 17 (3), 218–233. https://doi.org/10.1016/s1057-7408(07)70030-x.

Ebert, F., Castor, F., Novielli, N., Serebrenik, A., 2019. Confusion in code reviews: reasons, impacts, and coping strategies. In: 2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER). Hangzhou, China, pp. 49–60. https://doi.org/10.1109/SANER.2019.8668024. February 24–27, 2019.

Edvardsson, B., 1992. Service breakdowns: a study of critical incidents in an airline. Int. J. Serv. Ind. Manag. 3 (4), 17–29. https://doi.org/10.1108/09564239210019450.

Fischer, T., Riedl, R., 2017. Technostress research: a nurturing ground for measurement pluralism? Commun. Assoc. Inf. Syst. 40, 375–401. https://doi.org/10.17705/1cais.04017.

Flanagan, J.C., 1954. The critical incident technique. Psychol. Bull. 51 (4), 327–358. https://doi.org/10.1037/h0061470.

Fuglseth, A.M., Sørebø, Ø., 2014. The effects of technostress within the context of employee use of ICT. Comput. Hum. Behav. 40, 161–170. https://doi.org/10.1016/j.chb.2014.07.040.

Furuyama, T., Arai, Y., Iio, K., 1997. Analysis of fault generation caused by stress during software development. J. Syst. Softw. 38 (1), 13–25. https://doi.org/10.1016/s0164-1212(97)00064-2.

Galluch, P., Grover, V., Thatcher, J., 2015. Interrupting the workplace: examining stressors in an information technology context. J. Assoc. Inf. Syst. 16 (1), 1–47. https://doi.org/10.17705/1jais.00387.

Gaudioso, F., Turel, O., Galimberti, C., 2017. The mediating roles of strain facets and coping strategies in translating techno-stressors into adverse job outcomes. Comp. Hum. Behav. 69, 189–196. https://doi.org/10.1016/j.chb.2016.12.041.

Graziotin, D., Fagerholm, F., Wang, X., Abrahamsson, P., 2018. What happens when software developers are (un) happy. J. Syst. Soft. 140, 32–47. https://doi.org/10.1016/j.jss.2018.02.041.

Graziotin, D., Wang, X., Abrahamsson, P., 2015. Do feelings matter? On the correlation of affects and the self-assessed productivity in software engineering. J. Softw. Evol. and Proc. 27, 467–487. https://doi.org/10.1002/smr.1673.

Gremler, D.D., 2004. The critical incident technique in service research. J. Serv. Res. 7 (1), 65–89. https://doi.org/10.1177/1094670504266138.

Ioannou, A., Lycett, M., Marshan, A., 2022. The role of mindfulness in mitigating the negative consequences of technostress. Inf. Syst. Front. 1, 27. https://doi.org/10.1007/s10796-021-10239-0.

Lazarus, R.S., Folkman, S., 1984. Stress, Appraisal, and Coping. Springer, New York.

Lowry, P.B., D'Arcy, J., Hammer, B., Moody, G.D., 2016. "Cargo Cult" science in traditional organization and information systems survey research: a case for using nontraditional methods of data collection, including Mechanical Turk and online panels. J. Strateg. Inf. Syst. 25 (3), 232–240. https://doi.org/10.1016/j.jsis.2016.06.002.

Mäntylä, M., Adams, B., Destefanis, G., Graziotin, D., Ortu, M., 2016. Mining valence, arousal, and dominance: possibilities for detecting burnout and productivity?. In: Proceedings of the 13th International Conference on Mining Software Repositories (MSR '16). NY, USA. Association for Computing Machinery, pp. 247–258. https://doi.org/10.1145/2901739.2901752.

Matheny, K., Aycock, D., Curlette, W., Junker, G., 1993. The coping resources inventory for stress: a measure of perceived resourcefulness. J. Clin. Psychol. 49, 815–830. https://doi.org/10.1002/jclp.10219.

Najmul Islam, A.K.M., 2014. Sources of satisfaction and dissatisfaction with a learning management system in post-adoption stage: a critical incident technique approach. Comput. Hum. Behav. 30, 249–261. https://doi.org/10.1016/j.chb.2013.09.010.

Newman, K., Endres, M., Weimer, W., Johnson, B., 2023. From organizations to individuals: psychoactive substance use by professional programmers. In: 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE). Melbourne, Australia, pp. 665–677. https://doi.org/10.1109/ICSE48619.2023.00065. May 14–20, 2023.

Nisafani, A.S., Kiely, G., Mahony, C., 2020. Workers' technostress: a review of its causes, strains, inhibitors, and impacts. J. Decis. Syst. 29 (1), 243–258. https://doi.org/10.1080/12460125.2020.1796286.

Oakland, S., Ostell, A., 1996. Measuring coping: a review and critique. Hum. Rel. 49, 133–155. https://doi.org/10.1177/001872679604900201.

Ostberg, J.P., Graziotin, D., Wagner, S., Derntl, B., 2020. A methodology for psychobiological assessment of stress in software engineering. PeerJ Comput. Sci. 6, e286. https://doi.org/10.7717/peerj-cs.286.

Paudel, N.R., Adhikari, B.A., Prakash, K.C., Kyrönlahti, S., Nygård, C.H., Neupane, S., 2022. Effectiveness of interventions on the stress management of schoolteachers: a systematic review and meta-analysis. Occup. Environ. Med. 79 (7), 477–485. https://doi.org/10.1136/oemed-2021-108019.

Peer, E., Brandimarte, L., Samat, S., Acquisti, A., 2017. Beyond the Turk: alternative platforms for crowdsourcing behavioral research. J. Exp. Soc. Psychol. 70, 153–163. https://doi.org/10.1016/j.jesp.2017.01.006.

Pirkkalainen, H., Salo, M., Tarafdar, M., Makkonen, M., 2019. Deliberate or instinctive? Proactive and reactive coping for technostress. J. Manag. Inf. Syst. 36 (4), 1179–1212. https://doi.org/10.1080/07421222.2019.1661092.

Ragu-Nathan, T.S., Tarafdar, M., Ragu-Nathan, B.S., Tu, Q., 2008. The consequences of technostress for end users in organizations: conceptual development and empirical validation. Inf. Syst. Res. 19 (4), 417–433. https://doi.org/10.1287/isre.1070.0165.

Rajeswari, K.S., Anantharaman, R.N., 2003. Development of an instrument to measure stress among software professionals. In: Proceedings of the 2003 SIGMIS Conference on Computer Personnel Research (SIGMIS CPR'2003). Philadelphia, Pennsylvania, USA, pp. 33–43. https://doi.org/10.1145/761849.761855. April 10–12, 2003.

Reinecke, L., Aufenanger, S., Beutel, M.E., Dreier, M., Quiring, O., Stark, B., Wölfling, K., Müller, K.W., 2017. Digital stress over the life span: the effects of communication load and internet multitasking on perceived stress and psychological health impairments in a German probability sample. Media Psychol. 20 (1), 90–115. https://doi.org/10.1080/15213269.2015.1121832.

Riedl, R., Kindermann, H., Auinger, A., Javor, A., 2012. Technostress from a neurobiological perspective. Bus. Inf. Syst. Eng. 4 (2), 61–69. https://doi.org/10.1007/s12599-012-0207-7.

Russo, D., Hanel, P.H.P., Altnickel, S., van Berkel, N., 2021. Predictors of well-being and productivity among software professionals during the COVID-19 pandemic – a longitudinal study. Empir. Softw. Eng. 26, 62. https://doi.org/10.1007/s10664-021-09945-9.

Salazar-Concha, C., Ficapal-Cusí, P., Boada-Grau, J., Camacho, L.J., 2021. Analyzing the evolution of technostress: a science mapping approach. Heliyon 7 (4), e06726. https://doi.org/10.1016/j.heliyon.2021.e06726.

Salo, M., Frank, L., 2017. User behaviours after critical mobile application incidents: the relationship with situational context. Inf. Syst. J. 27 (1), 5–30. https://doi.org/10.1111/isj.12081.

Salo, M., Makkonen, M., Hekkala, R., 2020. The interplay of IT users' coping strategies: uncovering momentary emotional load, routes, and sequences. MIS Q. 44 (3), 1143–1175. https://doi.org/10.25300/misq/2020/15610.

Salo, M., Pirkkalainen, H., Chua, C., Koskelainen, T., 2017. Explaining information technology users' ways of mitigating technostress. In: Proceedings of the 25th European Conference on Information Systems (ECIS'2017). Guimarães, Portugal, pp. 2460–2476. June 5–10, 2017.

Sellberg, C., Susi, T., 2014. Technostress in the office: a distributed cognition perspective on human–technology interaction. Cogn. Technol. Work. 16 (2), 187–201. https://doi.org/10.1007/s10111-013-0256-9.

Serenko, A., Turel, O., 2010. Rigor and relevance: the application of the critical incident technique to investigate email usage. J. Organ. Comput. Electron. Commer. 20 (2), 182–207. https://doi.org/10.1080/10919391003711050.

Siitonen, V., Ritonummi, S., Salo, M., Pirkkalainen, H., 2022. The emergence of technostress in software development work: technostressors and underlying factors. In: Proceedings of the 8th International Workshop on Socio-Technical Perspectives in Information Systems Development (STPIS'2022). Reykjavik, Iceland. CEUR Workshop Proceedings, p. 3239. August 19–21, 2022.

Skinner, E.A., Zimmer-Gembeck, M.J., 2007. The development of coping. Annu. Rev. Psychol. 58 (1), 119–144. https://doi.org/10.1146/annurev.psych.58.110405.085705.

Skinner, E.A., Zimmer-Gembeck, M.J., 2016. Ways and families of coping as adaptive processes. In: Skinner, E.A., Zimmer-Gembeck, M.J. (Eds.), The Development of Coping. Springer, Cham, pp. 27–49. https://doi.org/10.1007/978-3-319-41740-0_2.

Skinner, E.S., Edger, K., Altman, J., Sherwood, H., 2003. Searching for the structure of coping: a review and critique of category systems for classifying ways of coping. Psychol. Bull. 129 (2), 216–269. https://doi.org/10.1037/0033-2909.129.2.216.

Sonnentag, S., Brodbeck, F.C., Heinbokel, T., Stolte, W., 1994. Stressor-burnout relationship in software development teams. J. Occup. Organ. Psychol. 67 (4), 327–341. https://doi.org/10.1111/j.2044-8325.1994.tb00571.x.

Stemler, S.E., 2004. A Comparison of consensus, consistency, and measurement approaches to estimating interrater reliability. Pract. Assess. Res. Eval. 9, 4. https://doi.org/10.7275/96jp-xz07.

Suárez, J., Vizcaíno, A., 2024. Stress, motivation, and performance in global software engineering. J. Softw. Evol. Proc. 36 (5), e2600. https://doi.org/10.1002/smr.2600.

Syed, M., Nelson, S.C., 2015. Guidelines for establishing reliability when coding narrative data. Emerg. Adulthood. 3 (6), 375–387. https://doi.org/10.1177/2167696815587648.

Taipalus, T., Seppänen, V., Pirhonen, M., 2020. Uncertainty in information system development: causes, effects, and coping mechanisms. J. Syst. Soft. 168, 11. https://doi.org/10.1016/j.jss.2020.110655.

Tarafdar, M., Bolman Pullins, E., Ragu-Nathan, T.S., 2015. Technostress: negative effect on performance and possible mitigations. Inf. Syst. J. 25 (2), 103–132. https://doi.org/10.1111/isj.12042.

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. Inf. Syst, J. 29 (1), 6–42. https://doi.org/10.1111/isj.12169.

Tarafdar, M., Pirkkalainen, H., Salo, M., Makkonen, M., 2020. Taking on the "dark side" —Coping with technostress. IT Prof. 22 (6), 82–89. https://doi.org/10.1109/mitp.2020.2977343.

Tarafdar, M., Tu, Q., Ragu-Nathan, T.S., 2010. Impact of technostress on end-user satisfaction and performance. J. Manag. Inf. Syst. 27 (3), 303–334. https://doi.org/10.2753/mis0742-1222270311.

Tarafdar, M., Tu, Q., Ragu-Nathan, T.S., Ragu-Nathan, B.S., 2011. Crossing to the dark side: examining creators, outcomes, and inhibitors of technostress. Commun. ACM 54 (9), 113–120. https://doi.org/10.1145/1995376.1995403.

Tracy, S.J., 2010. Qualitative quality: eight "big-tent" criteria for excellent qualitative research. Qual. Inq. 16 (10), 837–851. https://doi.org/10.1177/1077800410383121.

Trinkenreich, B., Stol, K.J., Steinmacher, I., Gerosa, M.A., Sarma, A., Lara, M., Feathers, M., Ross, N., Bishop, K., 2023. A model for understanding and reducing developer burnout. In: 2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP). Melbourne, Australia, pp. 48–60. https://doi.org/10.1109/ICSE-SEIP58684.2023.00010. May 14–20, 2023.

Tulili, T.R., Capiluppi, A., Rastogi, A., 2023. Burnout in software engineering: a systematic mapping study. Inf. Softw. Technol. 155, 107116. https://doi.org/10.1016/j.infsof.2022.107116.

Van den Brande, W., Baillien, E., Vander Elst, T., De Witte, H., Godderis, L., 2020. Coping styles and coping resources in the work stressors–workplace bullying relationship: a two-wave study. Work Stress 34, 323–341. https://doi.org/10.1080/02678373.2019.1666433.

Venkatesh, V., Thong, J.Y.L., Chan, F.K.Y., Hoehle, H., Spohrer, K., 2020. How agile software development methods reduce work exhaustion: insights on role perceptions and organizational skills. Inf. Syst. J. 30, 733–761. https://doi.org/10.1111/isj.12282.

Webster, V., Brough, P., Daly, K., 2016. Fight, flight or freeze: common responses for follower coping with toxic leadership. Stress Health 32 (4), 346–354. https://doi.org/10.1002/smi.2626.

Weerasekara, M., Smedberg, Å.B., Karunathilaka, G., Sandmark, H., 2022. User needs gathering for the design of information and communications technology-supported occupational stress management intervention: a quantitative study. Digit. Health 8, 1–15. https://doi.org/10.1177/20552076221127778.

Windeler, J.B., Maruping, L., Venkatesh, V., 2017. Technical systems development risk factors: the role of empowering leadership in lowering developers' stress. Inf. Syst. Res. 28 (4), 775–796. https://doi.org/10.1287/isre.2017.0716.

Wong, N., Jackson, V., Van Der Hoek, A., Ahmed, I., Schueller, S.M., Reddy, M., 2023. Mental wellbeing at work: perspectives of software engineers. In: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems. Hamburg, Germany, 858. https://doi.org/10.1145/3544548.3581528. April 23–28.

Zhao, X., Xia, Q., Huang, W., 2020. Impact of technostress on productivity from the theoretical perspective of appraisal and coping processes. Inf. Manag. 57 (8), 1–11. https://doi.org/10.1016/j.im.2020.103265.