Understanding Psychological Safety in Agile Software Development Teams

Lidivânio Monte
Technology Department
State University of Feira de Santana
Feira de Santana, Bahia, Brazil
lidivanio33@gmail.com

Beatriz Santana
Computing Institute
Federal University of Bahia
Salvador, Bahia, Brazil
santanab@ufba.br

Glauco de Figueiredo Carneiro
Computer Science Department
Federal University of Sergipe
São Cristóvão, Sergipe, Brazil
glauco.carneiro@dcomp.ufs.br

Manoel Mendonça
Computing Institute
Federal University of Bahia
Salvador, Bahia, Brazil
manoel.mendonca@ufba.br

Sávio Freire
Teaching Department
Federal Institute of Ceará
Morada Nova, Ceará, Brazil
savio.freire@ifce.edu.br

José Amancio Macedo Santos
Technology Department
State University of Feira de Santana
Feira de Santana, Bahia, Brazil
zeamancio@uefs.br

ABSTRACT

Context: Psychological safety (PS) refers to individuals' perceptions of the interpersonal risks of speaking up or making mistakes in team environments. In agile software development, PS is especially relevant due to its emphasis on collaboration, iteration, and continuous feedback. Despite growing interest in PS, there is still a limited understanding of how specific situations in agile contexts trigger psychological insecurity. **Aims:** This study aims to identify situations that challenge PS in agile teams, explore the causes behind these situations, and analyze their effects on team members and processes. **Method:** We conducted a qualitative content analysis of 54 discussions from Stack Exchange forums, focusing on real-life agile development experiences. **Results:** We identified five categories of challenging situations (e.g., misalignment of agile roles and responsibilities), seven categories of causes (e.g., process conflicts and bureaucracy), and five categories of consequences (e.g., team management and dynamic issues). Besides, we found frequent co-occurrence patterns between causes and consequences, revealing how PS disruptions propagate in agile teams. **Conclusion:** By categorizing real-world PS challenges, this study offers actionable directions for agile practitioners to recognize, anticipate, and mitigate psychologically unsafe conditions. Our taxonomy contributes to a more practical and nuanced understanding of PS in agile environments, supporting both empirical research and organizational interventions.

KEYWORDS

Psychological Safety, Agile Software Development, Grey Literature.

1 Introduction

The concept of psychological safety (PS) refers to individuals' perceptions of the consequences of taking interpersonal risks within a given context, such as in the workplace [9, 10]. In psychologically safe environments, team members feel comfortable sharing ideas, expressing concerns, seeking feedback, and taking initiative without fear of negative consequences [22]. PS has been widely studied across domains and is recognized as a critical factor in enabling team performance, collaboration, learning, and innovation.

In software engineering (SE), where team-based work is the norm and communication is key, PS becomes especially relevant. Agile methodologies, in particular, emphasize collaboration, iterative feedback, and adaptability, all of which depend on a high level of interpersonal trust and open communication. Despite increasing interest in the topic, empirical studies that examine PS within SE — and more specifically in agile contexts — remain limited [1, 5, 17].

Recent research has begun to fill this gap. For instance, Khanna and Wang [15] investigated the influence of digital tools on PS in online retrospectives, Hennel and Rosenkranz [14] analyzed the role of agile practices and social mechanisms in fostering team performance through PS, and Alami et al. [1] explored organizational and individual antecedents of PS in agile environments. However, these studies primarily focus on the benefits of PS or strategies to improve it, leaving a critical question underexplored: what are the concrete situations that challenge PS in agile software teams, and what causes and consequences are associated with them?

This study addresses that gap by analyzing spontaneous, real-world discussions drawn from public question & answer platforms, which offer candid insights from practitioners. Through qualitative content analysis of 54 discussions related to agile development, we identified five categories of situations that challenge PS (e.g., misalignment of agile roles & responsibilities, conflicts in time management & effort estimation), seven categories of underlying causes (e.g., process conflicts and interpersonal tensions), and six categories of consequences (e.g., reduced productivity, demotivation).

Our findings highlight that PS is often compromised by everyday dynamics that agile teams face, being many of which embedded in roles, ceremonies, and organizational expectations. We also observed patterns of co-occurrence between causes and consequences, providing new insights into how tensions emerge and propagate in team settings. These insights offer practical implications for agile practitioners, such as Scrum Masters, team leads, and organizational coaches, who can use this taxonomy to better anticipate risk scenarios, facilitate team communication, and design safer work environments. For researchers, our work expands the theoretical understanding of PS by uncovering how it is disrupted in situ, supporting further empirical and interventionist research.

The remainder of this paper is organized as follows. Section 2 presents relevant background and related work. Section 3 details

the research method. Section 4 describes the results regarding PS situations and its causes and consequences. Section 5 discusses our findings, contrasting them with related work, presenting co-occurrence cause-effect relationships, and addressing the trustworthiness of the study. Lastly, Section 6 concludes this work and offers directions for future research.

2 Background

This section presents concepts related to PS and studies that investigate it in the SE domain.

2.1 Psychological safety

PS refers to the shared belief that the team is a safe space for taking interpersonal risks [9]. In such environments, individuals feel comfortable speaking up, admitting mistakes, asking for help, and challenging ideas without fear of retaliation. Originally conceptualized by Edmondson in organizational teams [9], PS has been shown to promote learning, innovation, and high performance, especially in complex, interdependent work settings.

O'donovan and Mcauliffe [23] highlighted that psychologically safe teams tend to show considerable improvement in essential teamwork processes, such as performance, mutual trust, decision-making, team cohesion, motivation, and conflict resolution. This evolution occurs because in an environment where PS is valued, team members feel more comfortable expressing their ideas and opinions, which can lead to a more open and constructive exchange of information and feedback.

In software development teams, PS is essential due to the diversity of skills involved and the complexity of the systems being developed [8]. The socio-technical nature of SE can give rise to interpersonal conflicts that challenge PS. Such conflicts may stem from factors like changing requirements that jeopardize project outcomes [19]. Teams that adopt agile methodologies may be particularly susceptible to these conflicts, as agile emphasizes close customer collaboration, evolving requirements, and iterative development [6].

2.2 Related work

Some studies have explored PS within the context of agile development. Hennel and Rosenkranz [14] conducted three case studies in two large companies to examine how PS and agile social practices influence team performance. Their findings indicate that PS plays a critical role: teams with high levels of PS are more likely to adopt and benefit from agile social practices, leading to greater participation and fostering a culture of continuous improvement.

In other study, Christensen and Tell [5] combined literature review and case studies to define strategies to promote and maintain SP in agile software development teams. The authors defined a set of practices, such as regular feedback sessions, team reviews and retrospectives, PS trainings & workshops, recognition & validation practices, facilitation of open dialogues, and trust & vulnerability mapping.

Khanna and Wang [15] examined how the use of online tools influences PS during agile retrospectives conducted in virtual environments. The study focused on an agile software development team within a large multinational company. The findings revealed that participants preferred using video (via avatars or photos), audio, chat, and emojis to facilitate communication during retrospectives. Many opted to keep their cameras off and instead use avatars. Digital tools such as whiteboards and post-it notes were effective in structuring and guiding the discussions. Notably, the option to contribute anonymously enhanced participants' willingness to express emotions and share opinions without fear of judgment.

More recently, Alami et al. [1] conducted a multi-method study to investigate the factors that foster the creation and maintenance of PS in agile teams. Their findings emphasized the importance of a collaborative effort among leadership, team members, and individuals to institutionalize PS within agile development environments, ultimately enhancing team quality and performance. In another study, Alami et al. [2] interviewed agile team members and surveyed software practitioners to examine how PS influences software quality. The results revealed that PS plays a critical role in encouraging behaviors such as admitting mistakes and taking initiative — both essential for continuous learning and improving software quality.

These studies underscore that PS is a key contributor to team performance and software quality. They demonstrate the value of adopting social agile practices, implementing strategies to strengthen PS, and addressing both social and technical factors to create a collaborative and productive work environment. However, none of them has examined the specific situations that pose challenges to PS in agile teams, nor have they analyzed the underlying causes and resulting consequences of such challenges. Exploring these dimensions provides a more comprehensive understanding of the factors that influence PS in agile settings and offers actionable insights for its ongoing enhancement.

Additionally, prior studies have not incorporated grey literature as a data source for investigating PS. Grey literature includes a wide range of informal yet informative materials, such as blogs, technical reports, and community discussions. Different sources of GL have been used to explore SE topics [3, 4]. Question and answer (Q&A) platforms, such as the Stack Exchange[1] network, are a grey literature source that emerged as the most prominent example of a repository for practitioners' knowledge, and several studies have analyzed its discussions regarding topics such as requirements engineering [11], human factors [24–26], and technical debt [12, 13, 27]. We have initiated qualitative investigations that explore PS in SE using discussions from Stack Exchange forums [25, 26]. Although these studies identified interpersonal challenges and indicators of risk to PS, they did not specifically examine the situations, causes, and consequences that challenge PS in agile software development. This study aims to precisely address this gap.

3 Research method

This section outlines the research questions guiding this study, along with the data collection and analysis procedures employed.

3.1 Research questions

The goal of this work is to understand the challenges that affect PS in agile teams by exploring the situations that trigger these

[1] https://stackexchange.com/

challenges, their underlying causes, and their consequences. Thus, we defined the following research questions (RQs):

- **RQ1: What situations create challenges to psychological safety in agile development teams?** This question aims to identify scenarios in which individuals may feel vulnerable or exposed, thereby creating challenges to PS within agile development environments.
- **RQ2: What are the primary causes of situations that challenge psychological safety in agile teams?** This question seeks to analyze the specific dynamics and interactions of agile development that give rise to situations compromising psychological well-being.
- **RQ3: What are the main consequences of situations that challenge psychological safety?** This question explores the outcomes of PS-challenging situations, focusing on their impact on professionals' well-being and team dynamics in agile settings.

3.2 Data collection

To answer the RQs, we used the dataset composed of discussion from *Stack Exchange* collected by Santana et al. [25, 26]. *Stack Exchange* is a Q&A site that offers a wide range of freely accessible content, validated both by moderators and by its active user community. The community comprises professionals, academics, and students from diverse backgrounds, institutions, and regions around the world. *Stack Exchange* consists of several sub-sites, each dedicated to a specific topic. The dataset is composed of discussions from *Stack Exchange Software Engineering*[2] *(SESE)* and *Stack Exchange Project Management*[3] *(SEPM)*, which focus on general aspects of SE and project management, respectively. These two sites offer complementary perspectives that contribute to a broader understanding of the systems development life cycle.

To define the dataset, Santana et al. [25, 26] defined a search string composed of the terms: "*psychological safe*," "*speak up*," "*voice*," and "*silenc*." To ensure that the discussions were related to software teams, they also included the terms "*team*" and "*people*." They applied this search string on SESE and SEPM, retrieving a set of discussions. Each discussion consists of a main question, answers provided by other users, and comments associated with both the question and the answers. To maximize the number of relevant results, they applied the search string on the body, title, and tags of the discussions.

They conducted the search on SESE in December 2022, identifying a total of 410 discussions. For SEPM, the search was carried out in June 2023, resulting in 306 discussions. In total, they collected 716 discussions from the two sites.

3.2.1 Data selection on psychological safety. To identify discussions on PS, two researchers individually reviewed the set of discussions identified in SESE and SEPM [25, 26]. Using a content analysis approach [16], they analyzed the titles and content of each discussion to identify those that addressed specific PS challenges. They included discussions if they described situations with interpersonal risk elements (e.g., fear of speaking up, discomfort in expressing disagreement). Each case was mapped to one of seven interpersonal PS challenges, derived from prior research based on Edmondson [9, 10]. Therefore, they excluded discussions that reflected technical or procedural dysfunctions without interpersonal implications. The results of their analyses were recorded in a spreadsheet, followed by a consensus meeting. A third researcher participated in this meeting to help resolve disagreements. This involvement also helped ensure that the individual analyses were complementary and deepened the understanding of the research topic. As a result of this process, a dataset of 116 discussions containing relevant content on PS in SE was compiled. More details is shown in Santana et al. [25, 26].

3.2.2 Data selection on psychological safety in the agile development environment. As Santana et al. [25, 26]' analysis did not focus specifically on the agile approach, we explored the dataset to verify if there are discussions on PS in agile software development process. This broader analysis revealed the significant presence and relevance of discussions. This insight was essential in defining the scope of the present study, as it underscored the need for a more targeted investigation. Then, we conducted a round of data selection using the original dataset (116 discussions).

The first stage of analysis involved a comprehensive reading of all 116 discussions to identify those related to agile methodologies. Additionally, this stage aimed to detect discussions that described situations posing challenges to PS within the agile context, as well as their causes and consequences for development teams. To guide this process, we established the following inclusion criterion: "the discussion must be explicitly related to agile development." Discussions that indicated aspects of PS but were not situated within the agile development context were marked as false positives and excluded from further analysis. To determine whether a situation was directly connected to agile development, we examined whether the title, body, or tags of the question contained explicit references to agile methodologies and/or their associated practices. As a result of this screening, 62 discussions were discarded, yielding a final dataset of 54 discussions for detailed analysis.

To illustrate the data selection process, Figure 1 presents an example of a discussion that met the inclusion criteria. In this example, agile-related elements are clearly present: (i) the author explicitly identifies their role as a "Scrum Master," indicating a Scrum-based context; and (ii) the body of the post describes typical agile practices, such as "Sprint Retrospectives" and team engagement challenges, grounded in a real-world scenario involving an eight-developer team. These elements confirm the post's relevance to agile methodology, justifying its inclusion in our analysis.

Two researchers independently conducted the data selection stage focused on PS in the agile development environment. The first researcher is a Master's student in Computer Science, and the second is a Ph.D. student in the same field. Both have experience in qualitative data analysis and a solid background in PS. After reading the discussions, each researcher added those they considered valid to a spreadsheet for a later consensus stage. Divergences were resolved by a third (senior) researcher. To assess the agreement between the two analyses, we calculated the Cohen's Kappa coefficient (κ)[20], which resulted in $\kappa = 0.86$, indicating a high agreement rate for SE studies ($\kappa \geq 0.79$) [18]. In total, 54 discussions remained.

[2] https://softwareengineering.stackexchange.com/
[3] https://pm.stackexchange.com/

Figure 1: Example of a discussion involving PS in an agile context

3.3 Data analysis

Initially, the same researchers who conducted the data selection phase (PS in agile development) performed a content analysis [7] to extract the data. They individually read the discussions, extracting units of meaning, i.e., the general content of the forum was segmented into central units that provided sufficient information to address the RQs. As a general rule, the units of meaning could consist of sentences or excerpts.

For RQ1, the researchers extracted units describing situations that generate challenges to PS. For RQ2, they extracted segments that identified the causes leading to such situations. Lastly, for RQ3, they extracted segments that presented the consequences of the identified problems. Table 1 shows an example of the data extraction process for each RQ, indicating a unit of meaning for each RQ.

Table 1: Examples of data extraction for each RQ

Research question	Label	Unit of meaning
RQ1	Participation in retrospectives	"*I am of the opinion that only the team should meet to discuss in retrospectives, else all team members will not open up. But one of the managers insist that all the stakeholders should also be included.*"
RQ2	Using an online tool to hold a meeting	"*Perhaps in part because the meeting is held using an online tool, I often times have very long periods of silence after I ask a question.*"
RQ3	Impact on team performance	"*Since 4 sprints now, we clearly see velocity going down, we have 3 developers, so if one of them started to be not engaged, it is impacting the whole team.*"

After the individual analyses, the researchers compiled their units of meaning in an Excel spreadsheet for each RQ to compare and discuss them. When there was agreement between the two analyses, the extracted data proceeded to the coding phase. In cases of disagreement, a third researcher reviewed the discussions and issued a final decision. In total, we identified 54 discussions containing situations, 42 discussions presenting causes, and 36 discussions describing consequences.

After that, the researchers applied manual open coding to label the units of meaning inductively, that is, based on the extracted data [16]. The labels characterized the extracted excerpts and the significant units identified. Each researcher individually coded all units of meaning. An example is shown in Table 1. A consensus meeting was held, and a third researcher resolved any disagreements. Although, we did not limit the number of labels per discussion, we observed that most messages naturally addressed only one PS situation (RQ1).

The researchers also inductively grouped the labels into categories, ensuring that similar labels for each RQ were clustered to form comprehensive categories. This process was conducted collaboratively by the researchers and reviewed by a third researcher with strong experience in qualitative analysis. The same coding and grouping process was applied to all RQs. Table 2 presents an example of this process for RQ1. Based on the labels, we defined an initial coding and grouped the labels into five categories: (1) misalignment of agile roles and responsibilities, (2) conflicts in time management and effort estimation, (3) external interference and reduced autonomy, (4) process misalignment and operational resistance, and (5) participation and communication issues in agile ceremonies.

Table 2: Example of the label grouping process for RQ1

Category	Label
Misalignment of agile roles and responsibilities	Lack of a product owner Disagreement between the product owner and CEO
Conflicts in time management and effort estimation	The team underestimates the time needed to complete tasks Issues filling out the timesheet
External interference and reduced autonomy	Management interferes with the work carried out by the team External interruptions to the project
Process misalignment and operational resistance	Developer does not adapt to new technologies The team does not use continuous integration practices
Participation and communication issues in agile ceremonies	Off-topic discussions during the Daily Scrum Lack of participation from team members in the Sprint Retrospective

4 Results

This section presents the results we found per research question.

4.1 RQ1: What situations create challenges to PS in agile development teams?

PS can play an important role in agile software development, enabling team members to feel safe sharing ideas, raising concerns, and collaborating effectively. However, a variety of workplace situations can pose challenges to this construct, impacting team dynamics and performance. We found five categories of situations that can compromise PS in agile teams, as shown in Figure 2. Below, we detail each category of challenge along with its underlying sources.

Misalignment of agile roles and responsibilities: Defining and executing roles and responsibilities within agile teams can create challenges to PS when there is a lack of clarity about responsibilities, difficulties in communication, and imbalances in power dynamics. Issues such as the absence of a well-defined *Product Owner*, disputes over authority, and lack of active participation

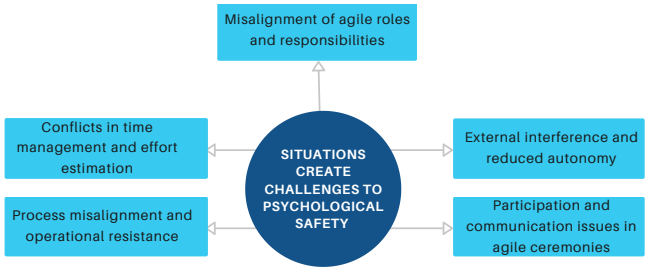

Figure 2: Situations that challenge psychological safety in agile software development teams

from team members can lead to insecurity, conflict, and poor collaboration.

These challenges emerge in a variety of situations, such as when a *Product Owner* overly centralizes decisions or, conversely, when the organization is unable to assign someone to this role, overloading the team with uncertainties. For example, in one of the cases analyzed, "*the organization has trouble appointing a product owner, because they 'can't find anybody who has enough knowledge and time to take on the job.*" When team members perceive this problem but do not feel comfortable discussing it openly – whether for fear of retaliation, judgment, or simply because they believe their concerns will not be taken into account – PS is compromised. Other factors include the difficulty in dealing with colleagues who do not collaborate or who impose their opinions, and the perception of low performance on the part of some members.

Conflicts in time management and effort estimation: Time management and effort estimation in agile development can create challenges to PS when there are inflexible expectations, lack of consensus on estimates, and excessive pressure to meet deadlines. Issues such as discrepancies in effort estimates, resistance to certain planning practices, and strict demands on productivity can lead to frustration, insecurity, and demotivation among the team.

These challenges arise in a variety of situations, such as when there is disagreement about the assignment of *story points* or resistance to the use of techniques such as *planning poker*. For example, in one of the cases analyzed, a team member reported: "*I've been here for a year and we have a very persistent problem: almost everybody can't finish planned work in almost every sprint. We have huge carryovers in every sprint. I tend to vote for larger estimates, but my teammates almost never learn from their past mistakes and persistently vote low in these estimates.*" When problems like these cannot be discussed openly, whether for fear of criticism, fear of appearing incompetent, or pressure to maintain idealized metrics, the PS of the team is compromised.

Other challenges include dissatisfaction with time tracking systems, the perception that the team is only following processes to meet management metrics, and excessive demands for granularity in tasks. When these issues are not addressed collaboratively, the team may feel undervalued, restricting their autonomy and compromising transparency in planning.

External interference and reduced autonomy: The influence of external interference and the reduced autonomy of the development team can create challenges to PS, especially when there are abrupt changes in priorities, imposition of decisions without consulting the team, and lack of autonomy to carry out the work. The constant presence of management in internal meetings, restrictions imposed without prior dialogue, and external pressures to change the planning impact the team's trust, making the environment less safe for open communication and collaboration.

These challenges emerge in several situations, such as when senior management imposes changes without considering the impact on the team or restricts the participation of developers in certain activities. For example, in one of the cases analyzed, a *Product Owner* reported: "*I'm a product owner on a 10 people Scrum team (too big of a team already) and last week Management came to us stating that developers can no longer participate in testing from the next sprint on, and that they need to focus on development activities or automation only (which they have never done before).*" When teams feel they cannot openly discuss these issues, whether due to fear of retaliation, lack of space for dialogue, or the perception that their concerns will not be taken seriously, PS is compromised.

Other challenges include managers being present at meetings such as dailies and retrospectives, creating an environment where team members may feel monitored or uncomfortable expressing concerns freely. In addition, pressure from stakeholders to constantly include new demands can destabilize the workflow. When these issues are not addressed transparently and collaboratively, teams may feel demotivated, lose a sense of control over their work, and avoid raising legitimate concerns.

Process misalignment and operational resistance: process misalignment and operational resistance arise in the context of agile development when there are difficulties in adapting to established practices, failures in process integration, and resistance to change. These issues can generate tension in the team, frustration, and resistance to following fundamental processes, compromising PS by limiting the team's ability to collaborate effectively and build an environment of trust.

These challenges arise in several situations, such as when there is a developer who performs work outside of normal sprint hours, creating difficulties in integrating with the team's work. An example of this occurred when a team member reported: "*Recently we have run into a problem of trying to integrate/handle the work of an overachieving developer doing out of band work (choosing to work outside of the normal working hours/sprint).*" The absence of a clear integration process can generate frustration among team members, especially when there is no alignment regarding expectations regarding deadlines and deliverables.

Other examples include team members becoming disengaged and demotivated during Scrum events, when one of them verbally expresses boredom, and resistance to adopting practices such as vertical slicing of features, even in the face of attempts to change. Difficulties in implementing essential development practices, such as automated testing and continuous integration, can also hinder team efficiency. In addition, a lack of cohesion in sub-teams within Scrum can negatively affect team collaboration and morale, making the work environment more challenging and, in many cases, undermining confidence in the effectiveness of the agile process. When these issues are not adequately addressed, PS is compromised, and the team may feel undervalued and unable to overcome operational obstacles.

Participation and communication issues in agile ceremonies: Agile processes and ceremonies are essential to ensuring team structure and alignment, but when they are not well managed or do not meet the needs of the team, they can create challenges to PS. Problems related to lack of participation, ineffective communication, and mismatched expectations can create an environment where team members feel uncomfortable or insecure about expressing themselves, directly impacting collaboration and performance.

These challenges manifest in various situations, such as difficulty encouraging participation in *Scrum meetings*, particularly during the *Sprint Retrospective*, where a lack of engagement is often observed. For example, one situation reported was: "*How to encourage more participation in web scrum meetings? This has only been a problem for me in one way, the Sprint Retrospective.*" When the team does not feel comfortable or does not see value in ceremonies, communication becomes limited and constructive feedback is lost. Furthermore, when a Scrum Master finds himself forced to interrupt team members during the *Daily Scrum* or when meetings become ineffective due to lack of participation, this can generate frustration and insecurity.

Other issues include difficulties in conducting effective retrospectives, especially in a remote context where physical distancing can hinder interaction. These issues can affect trust, making members less likely to actively engage or share issues openly.

In addition, the constant pressure to improve the effectiveness of ceremonies such as *Standups* and *Retrospectives*, when poorly conducted, can create discomfort and insecurity, especially when there is disagreement about stakeholder participation or how to document feedback. When members feel that their feelings or opinions are not taken into account, PS is compromised, resulting in less collaboration and a less productive work environment.

> **Finding #1**: The situations that challenge PS in agile software development can arise from the agile process itself—including its functions, roles, processes, and ceremonies — as well as from issues related to time management and planning, operational and procedural problems, and interferences or external factors.

4.2 RQ2: What are the primary causes of situations that challenge PS in agile teams?

The second research question explored the underlying causes of the PS challenges experienced by software developers. Analysis of the posts revealed six main categories of causes, listed below in descending order of frequency.

Process conflicts and bureaucracy arise when organizational practices conflict with the team's routine or impose requirements perceived as unnecessary. These tensions can reduce autonomy and hinder open communication, often leading to frustration. Common examples include mandatory time tracking or sudden backlog changes without prior discussion. The lack of consensus in estimations and the use of unclear metrics further erode trust in the process, as exemplified in "*Why is the team not burning hours properly in JIRA. Who in the team is not logging hours? Why the team has to be reminded each day to log hours in JIRA?*"

Difficulties in adopting scrum emerge when teams or organizations are not adequately prepared to implement agile principles coherently and consistently. Lack of experience, inconsistent or skeptical leadership, and superficial adoption of practices create confusion and demotivation. Some members find no value in ceremonies like *retrospectives*, while others report decisions based on incorrect assumptions. Misunderstandings about *Scrum roles and events* undermine collective learning, as we can see in "*Because of this change in Scrum Masters and their way of running the show, it has left my team numb to the idea of Scrum because the principles haven't been enforced consistently and one of the Scrum Masters was a person who do not believe in agile development …*"

Disagreements over participation in agile ceremonies can arise from hierarchical dynamics and external interference. For example, when a senior manager attended a *retrospective*, junior developers felt intimidated, leading to limited participation, as shown in "*Then last Sprint my boss attended the retrospective and the junior devs just shut down. He always asked everybody's opinion but every time he opened his mouth he killed any chance that 7 out of 8 team members would say anything.*" Additionally, an agile coach's presence, intended for guidance, sometimes disrupted team discussions. Another issue was the prioritization of personal matters over technical topics, affecting the flow of meetings. Finally, disagreements over stakeholder involvement and input from multiple departments complicated participation and focus.

Interpersonal conflicts and dynamics impact PS when aggressive behaviors, ineffective communication, or lack of collaboration arise. Ambiguous hierarchies, constant criticism, and difficulties in managing remote work intensify tensions. The absence of clear acceptance criteria and rigid role boundaries also contribute to frustration, ultimately reducing mutual trust and team engagement. For example, "*She is very rude and has an explosive personality if she does not agree with the Team.*"

Time management challenges and deadline pressure occur when estimations are inaccurate, meetings are excessive, or work pace is inconsistent, as illustrated in "*During the daily standup, he gets visibly upset when a subtask sits in the 'In Development' column for longer than a day or two. He claims that developers should create subtasks that are equivalent to 4–8 hours of development time.*" Delivery pressure and accumulating demands hinder collaboration and generate frustration, especially when productivity is judged harshly. External requests can also disrupt focus, diminishing commitment to the sprint and reducing the safety to express difficulties or suggest changes.

Lack of participation is observed when team members feel uncomfortable or unmotivated to actively engage in discussions and decisions. Dominance by authority figures and a lack of encouragement for divergent opinions foster passivity. Limited interaction weakens collaboration and hinders the construction of shared solutions. For example, "*Their PO is essentially like a well respected team leader. He speaks almost all the time and they rarely disagree with him. They are mostly silent and in agreement with everything he says.*"

Disagreements on technical decisions stem from misalignment regarding development practices or technology choices, which can lead to frustration and team fragmentation, as demonstrated in "*They just say that it's impossible and so they essentially work*

in horizontal slices with each team member working from his/her own backlog of technical tasks." Resistance to change — such as adopting new tools or frameworks — undermines effective Scrum implementation. When each member follows a personal technical logic, collaboration and shared goals are weakened.

> **Finding #2**: The challenges in PS arise from process-related issues (conflicts, bureaucracy, and difficulties adopting Scrum), team collaboration problems (interpersonal conflicts, lack of participation, and disagreements on agile ceremony involvement), time pressures (management challenges and deadlines), and technical misalignments (disagreements on technical decisions).

4.3 RQ3: What are the main consequences of situations that challenge PS?

In total, we found six consequences that arise from situations that affect PS in software development teams. The analysis revealed several recurring negative consequences, the most prominent of which concerns disruptions in team management and dynamics.

This study found that challenges to PS in software teams often lead to **team management and dynamic issues**, such as low morale, poor collaboration, and disrupted workflows. Issues such as unclear roles, dominance in discussions, and exclusion from decisions weaken cohesion. Team members may feel unmotivated, miss meetings, or experience frequent conflict. As we can see in "*Other developers complain about not being involved in design decisions related to these stories, since the work was done out of band.*"

Negative impact on productivity is a frequent result when PS challenges hinder communication, collaboration, and clarity of roles within software teams. These conditions reduce team speed and lead to rework, frustration, and lower overall performance. An illustrative situation involves the absence of a product owner, which disrupts planning and slows development: *As there is no prior analysis of tasks, we waste too much time in priority discussion and estimations. As these meetings are run by 'manager' we still plan everything based on task-people mapping rather than a general estimate.*"

Damage to organizational climate arises when PS challenges create an environment of distrust, reluctance to speak up, and disengagement. In such situations, team members may feel pressured to comply with practices they don't believe in, which leads to frustration and a decrease in morale. This not only affects trust and leadership but also hinders collaboration and transparency. One telling example comes from a retrospective where junior developers became silent when their boss attended: "*Last Sprint my boss attended the retrospective and the junior devs just shut down. I am against this because it erodes trust that I am trying to build with my team. It also reduces psychological safety within my team.*"

Increased task execution time occurs when decisions are based on incorrect assumptions or unnecessary adjustments, which leads to delays and inefficiencies. Time is wasted when tasks are revisited or re-prioritized without clear justification, causing frustration among team members. An example of this can be seen in the following: "*I notice that their decision is often based on wrong assumptions and false hypotheses; leading to a sorting readjustment, therefore wasting anybody's time.*"

Decreased motivation arises when there is a lack of engagement during key team activities, such as sprint planning or decision-making discussions. This can manifest as prolonged periods of silence, reduced effectiveness, and a general disconnection among team members. An example is: "*I often times have very long periods of silence after I ask a question.*"

> **Finding #3**: The consequences of challenges to PS include team-related issues (such as team dynamics and management), impacts on productivity and quality (including reduced productivity, and longer task execution times), and motivational and organizational effects (such as decreased motivation and negative organizational climate).

5 Discussion

This section discusses the findings obtained for each research question and presents the cause–consequence co-occurrence relationships identified in PS-related situations. It also addresses the trustworthiness of the study.

5.1 Revisiting the research questions

This study aims to identify situations that pose challenges to PS in agile development teams, the underlying causes of these situations, and the consequences for the team. The main findings of each research question are discussed below.

The analysis of situations that compromise PS in agile development teams (**RQ1**) reveals the complexity of the work environment, characterized by the interaction of organizational, technical, and interpersonal factors. Identifying these situations is important to developing strategies that promote a safer, more transparent, and collaborative work environment. This aligns with Edmondson's concept of PS as a shared belief that the team is safe for interpersonal risk-taking [9], and supports prior research showing that agile environments are particularly sensitive to contextual and interpersonal dynamics [14]. In our study, we found that certain agile roles and functions, such as *Scrum Master*, *Product Owner*, and *development team members*, are strongly linked to situations that pose risks to PS. Lack of clarity in responsibilities and task overload contribute significantly to insecurity in the workplace. Alami et al. [1] reported similar findings, highlighting that role ambiguity and poorly distributed workload are key antecedents of reduced PS in software teams. In addition, issues such as time management and planning, external factors and interference, operational issues, and decisions about agile ceremonies can contribute to this scenario.

The identified causes (**RQ2**), such as process conflicts and bureaucracy, reveal a disconnect between organizational practices and the needs of the team. The imposition of requirements, such as daily hour logging in JIRA without prior discussion, reduces the team's autonomy and hinders open communication. These tensions generate frustration and erode trust in the processes, contributing to the perception of an ineffective structure. Such tensions align with the findings of Newman et al. [22], who argued that PS is compromised in environments where team members lack control over

their work context and perceive top-down decisions as unjustified or non-transparent.

The impact of Scrum adoption emerged as another cause, particularly when the implementation is inconsistent or not properly understood. A lack of aligned leadership and resistance to agile practices compromise team engagement, creating an environment where ceremonies and processes are viewed as mandatory tasks with little real value. This can lead to a cycle of demotivation and frustration among team members. Such findings resonate with Christensen and Tell [5], who highlighted that superficial or rigid implementations of agile methods — disconnected from team values — can erode PS and result in disengagement.

Additionally, interpersonal dynamics is part of the PS of teams. Aggressive behaviors, ineffective communication, and lack of collaboration negatively affect mutual trust and collective engagement. These aspects are extensively covered in the literature, where PS is considered essential for enabling open dialogue, constructive conflict, and mutual support in teams [23]. The absence of clarity in roles and responsibilities, coupled with resistance to constructive feedback, creates an environment where team members do not feel safe to express opinions or admit mistakes, an issue also discussed in Alami et al. [2], who emphasize the role of PS in enabling learning behaviors such as speaking up and admitting errors.

The consequences of these challenges (**RQ3**) include a range of negative impacts on productivity and organizational climate. Increased task execution time, for example, often results from poor communication between the *Product Owner* and *development team*, leading to cluttered and overly complex user interfaces. The lack of clear requirements and insufficient communication with stakeholders make it difficult to deliver a product that meets end-user expectations. These results echo the findings of Khanna and Wang [15], who highlighted that communication breakdowns in low-PS environments are detrimental to alignment and product value delivery. Furthermore, decreased motivation and increased task execution time highlight the frustration of team members, further hindering collective performance and timely value delivery. These interdependent consequences highlight the need for a holistic approach to addressing PS-related issues.

The agile model, when implemented inconsistently or without adequate organizational support, can become a source of stress rather than a solution. Teams, particularly in transitions to agile practices, need a safe space to learn, fail, and adapt their practices according to emerging needs. This is consistent with the view of PS as a foundation for experimentation, team learning, and process improvement [9, 22].

Therefore, it is important for software organizations to create an environment that fosters genuine collaboration, where communication is clear, and expectations are realistically managed. PS is not only a necessary condition for the well-being of team members, but also a key factor for the continuous improvement of organizational performance and the quality of the final product [2, 14].

5.2 Analysis of cause–consequence co-occurrences in PS situations

Based on the discussions, we identified the cause–consequence co-occurrence relationships for each situation that challenges PS. We define cause–consequence co-occurrence as the presence of both elements within the same unit of meaning, i.e., within a single discussion describing a psychologically unsafe situation. These links were manually coded and cross-validated to ensure reliability. It is important to note that these links do not imply causality but rather indicate recurring contextual associations that may guide future research. These relationships are illustrated in a Sankey diagram (Figure 3). This visualization allows software practitioners and researchers to better understand the factors that give rise to PS-challenging situations, as well as their resulting effects. Below, we provide an example of how to interpret the diagram.

In the diagram, the *conflicts in time management and effort estimation* situation stands out as one of the most frequently discussed and represents a critical concern for PS in agile contexts. This category includes challenges related to defining deadlines, estimating effort, coping with productivity pressure, and inconsistencies in planning processes.

The diagram shows four distinct causes associated with this situation. The first relates to interpersonal conflicts and dynamics, where dysfunctional interactions — such as authoritarian or judgmental behavior — discourage team members from expressing concerns about estimates. The second involves difficulties in adopting Scrum, particularly when it is applied superficially or inconsistently, without alignment between leadership and team members, thereby weakening collaborative planning. The third cause is time management challenges and deadline pressure, characterized by rigid delivery expectations that often disregard the team's actual capacity, making members reluctant to communicate delays or propose adjustments. Lastly, process conflicts and bureaucracy stem from organizational constraints such as strict time-tracking policies or unplanned backlog changes, which reduce team autonomy and limit open discussion during planning.

This situation is linked to three consequences. First, it can cause team management and dynamic issues, including reduced cohesion, internal tension, and impaired collaboration. Second, it leads to decreased motivation, as team members feel their planning efforts are undervalued or ignored, diminishing their engagement. Third, it negatively impacts productivity, evident in misaligned expectations, rework, and suboptimal deliveries.

This structure of situation–cause–consequence co-occurrences also emerges clearly along a distinct path in the diagram. For instance, the situation identified as **"Participation and communication issues in agile ceremonies"** is illustrated by the question: "Under what circumstances should outsiders attend a scrum team's ceremonies?" This highlights tensions related to the involvement of external agents in agile rituals. The underlying cause is characterized as **"Interpersonal conflicts and dynamics,"** exemplified by a coach unilaterally imposing attendance without prior agreement: "The agile coach wants to 'observe' all the ceremonies of my team." Although intended as support, the lack of clear expectations and mutual consent created interpersonal tension. The poster reports adverse consequences: "It erodes trust that I am trying to build with my team. It also reduces psychological safety within my team and makes it harder to maintain leadership." Consequently, this leads to **"Damage to organizational climate,"** illustrating how well-intentioned but imposed practices can undermine trust, disrupt leadership, and hinder open communication—key elements for PS.

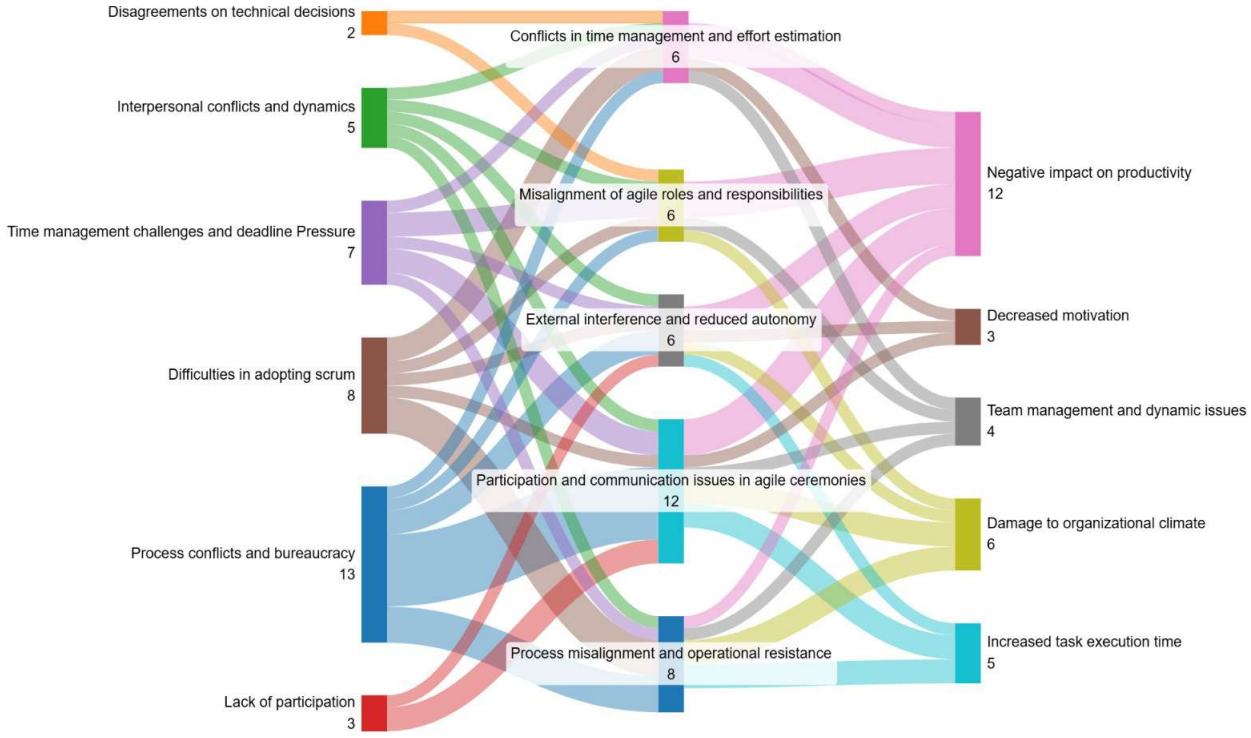

Figure 3: Sankey diagram linking categories of PS-challenging situations (center) to their causes (left) and consequences (right)

In summary, the cause–consequence co-occurrence relationships for *conflicts in time and effort estimation* go beyond operational hurdles, as they are shaped by relational and structural conditions that directly influence PS. In the absence of PS, professionals are less inclined to voice uncertainties, acknowledge limitations, or propose changes, ultimately undermining the effectiveness of agile practices.

We also identified co-occurrences[4] between the causes and consequences of PS challenges. These co-occurrences are represented in a heat map, as illustrated in Figure 4. The heat map reveals patterns that provide insight into the interplay between the causes and consequences of PS challenges in general:

- The *process conflicts and bureaucracy* cause was related to all effects found in this study, suggesting that excessive process complexity, rigid structures, or procedural inefficiencies can hinder team efficiency and contribute to a decline in the overall work atmosphere and collaboration.
- The *difficulties in adopting Scrum* cause was related to almost all effects (except for *team management and dynamic issues*). We found this cause strongly related to *negative impacts on productivity*, highlighting the operational and emotional challenges that teams often face when transitioning to or implementing agile methodologies, especially when adequate training, support, or alignment is lacking.
- The *interpersonal conflicts and dynamics* cause was mainly associated with *negative impacts on productivity* effects, emphasizing the importance of fostering a psychologically safe environment where open communication and constructive interactions can thrive.
- The *time management challenges and deadline pressure* cause was primarily linked to both *negative impacts on productivity* and *increased task execution time* effects, suggesting that high pressure related to deadlines may paradoxically result in slower task completion and reduced performance, likely due to stress and rushed decision-making.
- The *lack of participation* cause was related to *decreased motivation* and *negative impact on productivity* effects, showing that a psychologically safe environment is required to awaken the desire to carry out activities.
- The *negative impact on productivity* effect was related to all identified causes, revealing that PS affects software development teams' productivity in agile contexts.

5.3 Trustworthiness of the study

To ensure the trustworthiness of our qualitative study, we adopted strategies based on the criteria of credibility, transferability, dependability, and confirmability [7].

To guarantee **credibility**, all discussions were independently reviewed and analyzed by two researchers experienced in qualitative analysis. In cases of disagreement, a third senior researcher with domain expertise provided an independent assessment to ensure accuracy and depth of interpretation. Additionally, we used

[4]Co-occurrence relationships indicate the number of times a specific cause and a specific effect appeared together in the same discussion. However, it is important to note that these relationships do not imply causality.

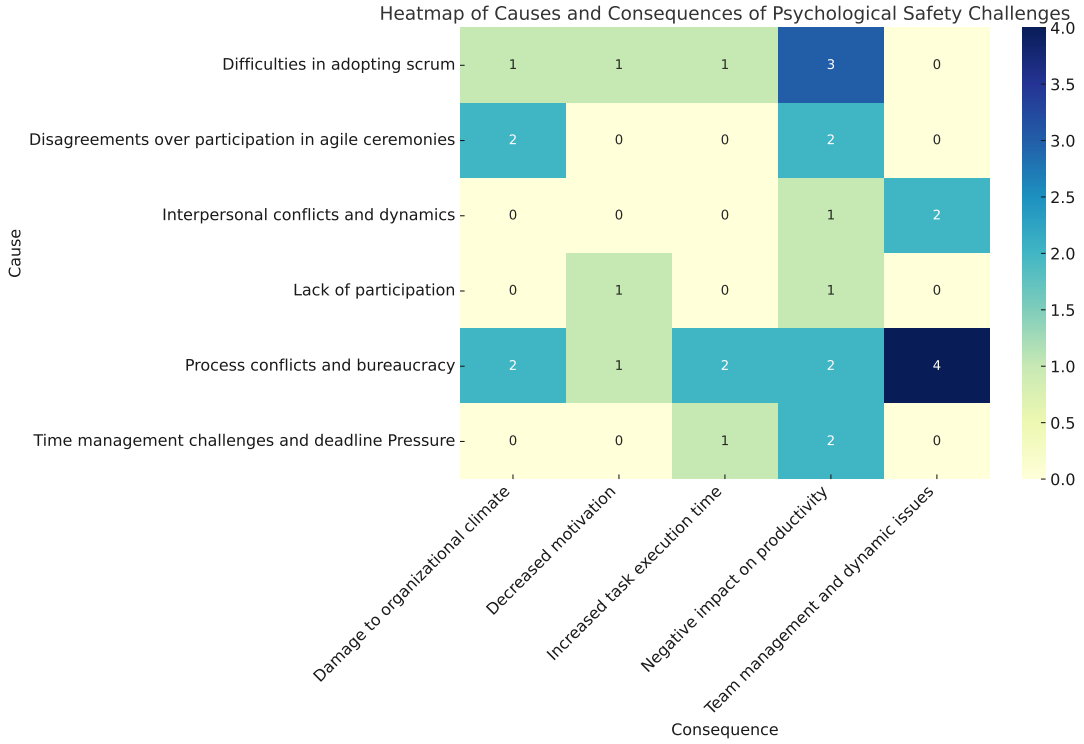

Figure 4: Heatmap of causes and consequences of Psychological Safety challenges

direct quotes from discussions to ground the interpretations in the participants' own words, reinforcing the link between data and analysis. Although our search string already draws on terminology introduced in Edmondson's work [9, 10], additional keywords could be incorporated to widen the range of discussions captured.

We addressed **transferability** by providing rich contextual information about the data source and the selection process. We detail the characteristics of the forums used (Stack Exchange Software Engineering and Project Management), the search strings applied, the inclusion criteria for identifying relevant discussions, and the nature of the selected dataset. This transparency enables readers to assess the applicability of our findings to similar agile environments, even though those findings are limited to the perspectives of platform users.

To enhance **dependability** we documented the research process in a consistent and traceable manner. We maintained detailed records of each phase, including data collection, filtering criteria, labeling decisions, and coding iterations. The peer-review procedures and Cohen's Kappa agreement score (0.86) further demonstrate the consistency and reliability of our selection process.

Lastly, we ensured **confirmability** by adopting reflexive practices and minimizing researcher bias. All interpretations were discussed collaboratively and validated against the original textual data. By relying on naturally occurring discourse rather than interviews or self-reports, we reduced the influence of social desirability or researcher intervention on the participants' responses.

6 Conclusion

This study proposes a taxonomy of psychologically unsafe situations based on practitioner narratives from Stack Exchange. By identifying patterns of causes and consequences, we highlight how interpersonal risks manifest in agile environments and threaten team collaboration, motivation, and learning. Our findings emphasize that PS is affected not only by individual behaviors but also by systemic issues such as misaligned roles, external interference, and flawed processes. The taxonomy provides a structured lens for agile practitioners to reflect on team dynamics and design targeted interventions. Future work can integrate this taxonomy into assessment tools or explore quantitative methods to test the observed co-occurrences at scale.

ARTIFACT AVAILABILITY

The complete data set is available in a Zenodo repository [21].

ACKNOWLEDGMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior Brasil (CAPES) - Finance Code 001, by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), and by the Fundação de Amparo a Pesquisa do Estado da Bahia (FAPESB).

REFERENCES

[1] Adam Alami, Mansooreh Zahedi, and Oliver Krancher. 2023. Antecedents of psychological safety in agile software development teams. *Information and*

Software Technology 162 (2023), 107267.

[2] Adam Alami, Mansooreh Zahedi, and Oliver Krancher. 2024. The role of psychological safety in promoting software quality in agile teams. *Empirical Software Engineering* 29, 5 (2024), 1–50.

[3] Lidiany Cerqueira, Sávio Freire, João Bastos, Rodrigo Spínola, Manoel Mendonça, and José Santos. 2023. A Thematic Synthesis on Empathy in Software Engineering based on the Practitioners' Perspective. In *Proceedings of the XXXVII Brazilian Symposium on Software Engineering* (Campo Grande, Brazil) *(SBES '23)*. Association for Computing Machinery, New York, NY, USA, 332–341. doi:10.1145/3613372.3613407

[4] Lidiany Cerqueira, Sávio Freire, Danilo Ferreira Neves, João Pedro Silva Bastos, Beatriz Santana, Rodrigo Spínola, Manoel Mendonça, and José Amancio Macedo Santos. 2024. Empathy and Its Effects on Software Practitioners' Well-Being and Mental Health. *IEEE Software* 41, 4 (2024), 95–104. doi:10.1109/MS.2024.3377897

[5] Mikkel Agerlin Christensen and Paolo Tell. 2022. Building a Toolbox for Working with Psychological Safety in Agile Software Teams. In *Agile Processes in Software Engineering and Extreme Programming: 23rd International Conference on Agile Software Development, XP 2022, Copenhagen, Denmark, June 13–17, 2022, Proceedings*. Springer, 82–98.

[6] A. Cockburn. 2002. "Agile software development joins the 'would-be' crowd". *Cutter ITJournal* 1, 15 (2002), 6–12.

[7] John W Creswell and J David Creswell. 2017. *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications.

[8] R. S. Dubey, V. Tewari, and B. Pandiya. 2017. A soft approach towards gaining employability in IT professionals. In *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. 299–303. doi:10.1109/IEEM.2017.8289900

[9] Amy Edmondson. 1999. Psychological safety and learning behavior in work teams. *Administrative science quarterly* 44, 2 (1999), 350–383.

[10] Amy C Edmondson and Zhike Lei. 2014. Psychological safety: The history, renaissance, and future of an interpersonal construct. *Annual review of organizational psychology and organizational behavior* 1, 1 (2014), 23–43.

[11] Sávio Freire, Felipe Gomes, Larissa Barbosa, Thiago Souto Mendes, Galdir Reges, Rita SP Maciel, Manoel Mendonça, and Rodrigo Spínola. 2023. Requirements Engineering Issues Experienced by Software Practitioners: A Study on Stack Exchange. In *International Working Conference on Requirements Engineering: Foundation for Software Quality*. Springer, 3–20.

[12] Felipe Gomes, Eder Santos, Sávio Freire, Thiago Souto Mendes, Manoel Mendonça, and Rodrigo Spínola. 2023. Investigating the Point of View of Project Management Practitioners on Technical Debt - A Study on Stack Exchange. *Journal of Software Engineering Research and Development* 11, 1 (Sep. 2023), 12:1 – 12:15. doi:10.5753/jserd.2023.3191

[13] Felipe Gomes, Eder Pereira dos Santos, Sávio Freire, Manoel Mendonça, Thiago Souto Mendes, and Rodrigo Spínola. 2022. Investigating the point of view of project management practitioners on technical debt: a preliminary study on stack exchange. In *Proceedings of the International Conference on Technical Debt*. 31–40.

[14] Phil Hennel and Christoph Rosenkranz. 2021. Investigating the "Socio" in Sociotechnical development: The case for psychological safety in agile information systems development. *Project Management Journal* 52, 1 (2021), 11–30.

[15] Dron Khanna and Xiaofeng Wang. 2022. Are Your Online Agile Retrospectives Psychologically Safe? the Usage of Online Tools. In *Agile Processes in Software Engineering and Extreme Programming: 23rd International Conference on Agile Software Development, XP 2022, Copenhagen, Denmark, June 13–17, 2022, Proceedings*. Springer, 35–51.

[16] Klaus Krippendorff. 2018. *Content analysis: An introduction to its methodology*. Sage publications.

[17] Per Lenberg and Robert Feldt. 2018. Psychological safety and norm clarity in software engineering teams. In *Proceedings of the 11th international workshop on cooperative and human aspects of software engineering*. 79–86.

[18] Jiakun Liu, Qiao Huang, Xin Xia, Emad Shihab, David Lo, and Shanping Li. 2020. Is Using Deep Learning Frameworks Free? Characterizing Technical Debt in Deep Learning Frameworks. In *2020 IEEE/ACM 42nd International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*. 1–10.

[19] Julie Yu-Chih Liu, Hun-Gee Chen, Charlie C Chen, and Tsong Shin Sheu. 2011. Relationships among interpersonal conflict, requirements uncertainty, and software project performance. *International Journal of Project Management* 29, 5 (2011), 547–556.

[20] Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochem Med (Zagreb)* 22, 3 (2012), 276–282.

[21] Lidivânio Monte, Beatriz Santana, Glauco Carneiro, Manoel Mendonça, Sávio Freire, and José Amâncio Macedo Santos. 2025. Understanding Psychological Safety in Agile Software Development Teams - Complementary Material. doi:10.5281/zenodo.16705151

[22] Alexander Newman, Ross Donohue, and Nathan Eva. 2017. Psychological safety: A systematic review of the literature. *Human resource management review* 27, 3 (2017), 521–535.

[23] Róisín O'donovan and Eilish Mcauliffe. 2020. A systematic review of factors that enable psychological safety in healthcare teams. *International journal for quality in health care* 32, 4 (2020), 240–250.

[24] Antônio Carlos M. de Paula, Carlos Frederico J. Muakad, Sávio Freire, Rodrigo Spínola, and Manoel Mendonça. 2024. Burnout in Software Projects: An Analysis of Stack Exchange Discussions. In *Proceedings of the XXIII Brazilian Symposium on Software Quality (SBQS '24)*. Association for Computing Machinery, New York, NY, USA, 321–330. doi:10.1145/3701625.3701670

[25] Beatriz Santana, Sávio Freire, José Amancio Santos, and Manoel Mendonça. 2024. Psychological Safety in the Software Work Environment. *IEEE Software* (2024).

[26] Beatriz Silva de Santana, Sávio Freire, Leandro Cruz, Lidivânio Monte, Manoel Mendonca, and José Amancio Macedo Santos. 2023. Exploring psychological safety in software engineering: Insights from stack exchange. In *Proceedings of the XXXVII Brazilian Symposium on Software Engineering*. 503–513.

[27] Eder Pereira Santos, Felipe Gomes, Sávio Freire, Manoel Mendonça, Thiago Souto Mendes, and Rodrigo Spínola. 2022. Technical debt on agile projects: Managers' point of view at stack exchange. In *Proceedings of the XXI Brazilian Symposium on Software Quality*. 1–9.