Investigating Enablers, Barriers, and Consequences of Psychological Safety in Software Engineering

Beatriz Santana
Computing Institute
Federal University of Bahia
Salvador, Bahia, Brazil
beatrizsecomp@gmail.com

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

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

Daniela Cruzes
Trondheim, Norway
danielascruzes@gmail.com

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

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

Abstract

Context. Psychological safety (PS) is fundamental in software development teams, whose effectiveness depends on collaboration, innovation, and open communication. Safe environments enable professionals to share ideas, raise concerns, and learn from mistakes without fear of negative repercussions, directly influencing both product quality and individual well-being. Although prior studies have investigated the antecedents and outcomes of PS in software engineering (SE), the results are limited to specific SE practices or software project contexts. Also, little is known about how PS enablers and barriers operate across multiple levels of analysis, including leadership, organizational, team, and individual. **Objective.** This study investigates the enablers, barriers, and consequences of PS in SE. **Method.** We conducted a qualitative study using semi-structured interviews with software professionals, analyzing their experiences across diverse organizational contexts. **Findings.** The results indicate that leadership, organizational practices, team dynamics, and individual characteristics can simultaneously act as enablers or obstacles. When present, PS fostered innovation, product quality, and motivation; when absent, it may lead to rework, delays, financial losses, and mental health issues such as burnout. **Conclusion.** This study positions PS as a strategic factor for efficiency, innovation, and human sustainability in SE, highlighting the importance of organizational practices that promote collaboration and protect professionals' well-being.

Keywords

Psychological Safety, Software Engineering, Team dynamics, Human factors, Well-being

1 Introduction

Psychological safety (PS) has been increasingly recognized as an important factor for both team performance and organizational sustainability across diverse contexts [8]. It refers to the shared belief that individuals can take interpersonal risks, such as expressing opinions, raising concerns, admitting mistakes, or asking questions, without fear of negative consequences, retaliation, or damage to their professional image [7]. When present, PS fosters continuous learning, innovation, and overall well-being in the workplace [9].

Systematic reviews and empirical studies in general organizational contexts have highlighted that PS emerges from the interplay of organizational systems, leadership practices, and individual attributes [16, 17]. For example, Newman et al. [16] find that primary studies have emphasized the role of leadership inclusiveness, peer support, and a supportive organizational culture, while O'donovan and Mcauliffe [17] show that organizational culture, leadership integrity, and individual responsibility collectively reinforce PS in healthcare. These findings illustrate that PS is a multilevel phenomenon, context-dependent and episodic.

Given that software engineering (SE) is inherently collaborative and interdependent, PS plays a key role in how teams function. Activities such as agile ceremonies, code reviews, and cross-functional collaboration expose developers to interpersonal risks, including disagreements with technical decisions, the identification of errors, or the proposal of changes [21, 22]. Such situations constitute social and interpersonal challenges that a psychologically safe environment can mitigate, thereby fostering openness, trust, and constructive interaction among team members [4, 5].

Empirical research in SE has begun to uncover dynamics analogous to those observed in broader organizational contexts. For example, Buvik and Tkalich [5] reported that team autonomy positively influences PS. Hennel and Rosenkranz [11] showed that agile practices, including stand-ups and retrospectives, foster transparency and communication, thereby reinforcing PS. Alami et al. [3] further identified leadership support, collective decision-making, and a no-blame culture as fundamental facilitators of PS, linking it to sustainable code, continuous learning, and improved software quality [2, 4]. Similarly, Ahmad [1] emphasized that leadership behavior, constructive feedback, and open communication are antecedents of PS, whereas its absence may contribute to talent attrition.

Despite these advances, most studies have investigated either antecedents or outcomes in specific SE practices (retrospectives) or software project context (agile methodologies), limiting the complete understanding of them. These studies have also offered limited understanding of how enablers and barriers operate across leadership, organizational, team, and individual levels, and how compromised PS affects innovation, delivery performance, and human sustainability [13, 21].

To address this gap, the present study explores PS in software development by examining the factors that enable or hinder professionals' willingness to express ideas, provide feedback, and acknowledge mistakes. We conducted semi-structured interviews with twelve software professionals across diverse organizations, team sizes, roles, and career stages. Participants shared experiences in which they felt safe or inhibited, allowing us to capture practical enablers and barriers to PS in real world settings.

The study identified patterns at multiple levels: leadership, team, organizational, and individual. Key enablers included `empathetic and technically competent leadership`, `open communication`, `mutual support`, `collaborative culture`, `diversity initiatives`, and `personal traits` such as extroversion and openness to feedback. Barriers included `inadequate leadership behaviors`, `low team openness`, `dismissal of opinions`, and `fear of interpersonal consequences`.

The study also highlights the consequences of PS. Safe environments fostered `motivation`, `engagement`, `innovation`, and `higher quality delivery`, while unsafe contexts led to `project delays`, `rework`, `burnout`, `disengagement`, and `financial and human costs`.

Overall, this research provides insights for cultivating PS in software development. By focusing on enablers and barriers, it offers guidance for leaders, teams, and organizations seeking to create environments that encourage learning, collaboration, and innovation. This study also confirms established findings on PS in SE, extends prior models by incorporating organizational and socio-technical mechanisms grounded in everyday practice, and contributes new empirical evidence on the technical, economic, and human consequences of psychological (un)safety in real-world software development contexts.

The remainder of this paper is organized as follows. Section 2 presents related work on PS in SE. Section 3 describes our research design, including data collection and analysis procedures. Section 4 reports our empirical findings regarding enablers, barriers, and consequences of PS. Section 5 discusses the implications of these findings, situating them within the broader literature and highlighting practical contributions. Section 6 outlines the threats to validity and measures adopted to ensure research rigor. Finally, Section 7 concludes the paper and offers directions for future research.

2 Related Work

PS is the perception that the work environment is safe for interpersonal risk-taking, allowing individuals to speak up, seek help, or admit mistakes without fear of negative consequences or blame [7, 23]. Early research highlighted its role in facilitating adaptation to organizational changes by promoting behavioral flexibility and reducing defensive responses [12]. Within team settings, PS reflects a shared belief that members can engage openly in discussions and express ideas without fear of reprisal, thereby fostering communication, learning, and collaboration [7].

Research across diverse organizational contexts underscores that PS emerges through complex interactions among factors spanning multiple levels [9, 16]. Central elements that contribute to PS include inclusive and transformational leadership, supportive peer relationships, organizational culture, clear role definitions, autonomy, task interdependence, and individual accountability [10, 16]. Complementing these findings, studies in healthcare emphasize the role of leadership integrity, alongside peer support and personal responsibility, in reinforcing PS [17].

PS has been associated with a range of positive outcomes, including increased engagement, information sharing, learning behaviors, creativity, and overall job satisfaction [10]. Nevertheless, PS is context-dependent, meaning its formation, development, and effectiveness vary according to team composition, task characteristics, and organizational setting [9, 26].

In SE, empirical research has begun to uncover similar dynamics. Buvik and Tkalich [5] conducted a quantitative survey of 236 members of 43 agile software teams in Norway, revealing that team autonomy significantly enhances PS, which in turn improves team reflexivity and performance. However, team reflexivity did not mediate the effect of PS on performance, and the study remained focused on team-level constructs.

In addition, Hennel and Rosenkranz [11] reported that agile practices such as retrospectives, stand-ups, and sprint planning foster transparency and communication, thereby strengthening PS. Alami et al. [2] qualitatively explored technical excellence in agile development, characterizing PS as a key enabler within a socio-technical system that supports sustainable code, continuous learning, and teamwork. Their findings emphasize leadership support, customer involvement, and PS as conditions facilitating innovation.

In another related work, Ahmad [1] further examined large-scale agile development and showed that leadership behavior, constructive feedback, and open communication are central to enabling PS, while its absence may lead to brain drain. Alami et al. [3] expanded this discussion through a mixed-methods study, demonstrating that openness, a no-blame culture, collective decision-making, and leadership ownership are key pillars of a psychologically safe workplace. These studies collectively indicate that PS in software development must be actively nurtured through coordinated efforts at the individual, team, and leadership levels. Complementing these findings, Alami et al. [4] used a two-phase mixed methods approach to investigate how PS promotes software quality behaviors in agile teams. Their qualitative phase (20 interviews) identified social enablers such as admitting mistakes and helping others, while a subsequent survey confirmed positive associations between PS and software quality outcomes.

Expanding beyond organizational case studies, Monte et al. [15] analyzed 54 discussions between practitioners on Stack Exchange using qualitative content analysis. They identified five categories of situations that challenge PS, such as misaligned agile roles, time pressure, and lack of autonomy, alongside causes like process conflicts and consequences including reduced motivation. This work underscores the importance of recognizing and mitigating PS challenges in agile teams.

Finally, a systematic review by Santana et al. [21] identified that while most studies emphasize the benefits of PS, such as improved collaboration, communication, and innovation capacity, few offer detailed insights into the concrete enablers that sustain PS, the barriers that undermine it, or the complex nature of its consequences. This gap indicates that, although the value of PS in software development is increasingly acknowledged, it remains unclear whether the mechanisms reported in prior work extend beyond specific SE practices (such as retrospectives, code review, and bug reporting and resolution) and hold across broader software development contexts.

Synthesizing these studies, the SE literature emphasizes antecedents such as leadership inclusiveness, autonomy, and agile practices as key drivers of PS [3, 5, 11]. Outcome-oriented studies further associate PS with improved team performance, product quality, and technical excellence [2, 4]. Despite these insights, there is a lack of integrative research exploring how enablers and barriers operate across organizational levels and the consequences of compromised PS. Addressing this gap is critical to advancing theoretical understanding and providing actionable guidance for practitioners seeking to cultivate sustainable, high-performing software development environments.

3 Research Method

This section presents the research questions and the data collection and analysis procedures we followed to answer them.

3.1 Research Questions

This study investigates the enablers, barriers, and consequences of PS in SE. More specifically, we seek to understand the factors that foster or hinder PS in software development, and to examine the consequences of these conditions for SE practice. To achieve this goal, we defined the following research questions (RQs):

- **RQ1**: *What factors or enabling conditions make psychological safety possible in software engineering teams?* Understanding the factors that enable PS is essential to identify what supports team members in speaking up, sharing ideas, and engaging in collaborative problem-solving. This question aims to capture the enablers that foster trust, openness, and learning within teams, providing insight into the conditions that promote a safe and productive work environment.
- **RQ2**: *What barriers hinder psychological safety in software engineering teams?* Examining the barriers to PS helps reveal the obstacles that prevent team members from expressing concerns, raising questions, or contributing fully. By identifying these obstacles, research can clarify how organizational, leadership, team, and individual factors can inhibit open communication and mutual support, highlighting areas for potential improvement.
- **RQ3**: *What are the consequences of psychological safety (or lack thereof) for software development?* Exploring the consequences of PS, or its absence, provides an understanding of how varying levels of safety impact SE outcomes. This includes effects on individual well-being, team dynamics, and organizational performance. The question motivates an analysis of the practical implications of fostering or neglecting PS in software development contexts.

To answer the RQs, we adopted a qualitative design based on individual, synchronous interviews with open-ended questions, following established guidelines for empirical research in SE [18]. This design enables the capture of diverse experiences while supporting the identification of common patterns, perceived barriers, and contextual factors influencing PS. Interviews were chosen for their ability to elicit in-depth accounts of complex social phenomena, grounded in participants' lived experiences and personal meanings [24].

3.2 Data Collection

Semi-structured interviews were conducted with a protocol organized around five core topics: (1) a situation in which the participant felt hesitant to speak up; (2) barriers to and factors that enabled PS in that situation; (3) perceived consequences of the situation; (4) actions, supports, or practices that could have improved PS; and (5) additional experiences or reflections related to PS. Follow-up questions were included under core topic (1) to probe details about context, participants involved, and specific outcomes. The full interview script is presented in Table 1. This approach enabled consistent data collection while capturing rich, detailed accounts of participants' experiences.

Table 1: Interview script

Interview questions
1. Have you ever encountered a situation within your team in which you felt apprehensive or reluctant to express your opinions or take a stance before your team, leadership, clients, or other departments?
1.1 Based on your experience, particularly in *[explicitly stated specific job-related activities]*, have you ever faced a situation where it was challenging to admit mistakes, disagree with colleagues, propose changes, request assistance, or provide negative feedback?
1.2 If not, in your experience, what factors do you believe enable individuals to feel comfortable admitting mistakes, disagreeing with colleagues, proposing changes, requesting assistance, or providing negative feedback in the workplace?
1.3 Could you describe this situation in greater detail?
2. What factors hindered or enabled your response or actions in that situation?
3. What consequences did this situation have for you personally, for the team, or for the progress of the project?
4. What might have helped you (or the team) manage this situation more effectively?
5. Is there any additional experience or aspect related to psychological safety that you would like to share?

We developed the research protocol based on insights from previous empirical studies and literature on PS, ensuring alignment with the study's RQs. The interview questions (IQs) were iteratively refined and validated through discussions with multiple researchers and a pilot interview, allowing us to confirm clarity, relevance, and completeness. In particular, the first three topics were explicitly mapped to the RQs: enabling factors (RQ1; IQ1, 1.1-1.3), barriers (RQ2; IQ2), and consequences (RQ3; IQ3), while the other topics provided an opportunity to capture supplementary perspectives and experiences not directly covered by the structured questions.

Prior to the interviews, participants read and accepted an informed consent form and completed a brief pre-interview questionnaire in a spreadsheet, which collected demographic and professional information such as role, experience, and team size. This step ensured that participants met the inclusion criteria and facilitated contextualization of the interview data.

Interviews were conducted online, lasting 40–60 minutes, and audio-recorded with participants' consent. Transcriptions were generated using Tactiq and manually reviewed for accuracy. All transcripts were anonymized before analysis. This study was approved by the Research Ethics Committee under the protocol number 7.807.142, registered in Plataforma Brasil with CAAE number 389437125.0.0000.5033. The research follows the ethical guidelines established by the Brazilian National Health Council Resolution, ensuring the protection of participants' rights and confidentiality.

We recruited participants through convenience and snowball sampling, leveraging professional contacts. The inclusion criterion required participants to be currently working in software development teams and actively engaging in collaborative work processes. In total, we interviewed 12 participants.

Saturation was operationally defined as the point at which two consecutive interviews produced no new codes within any of the three RQs. Code emergence was tracked cumulatively across interviews. For RQ1, all 16 enabler codes were identified by interview 8, with four subsequent interviews yielding no new codes. For RQ2, all 9 barrier codes emerged by interview 5, with seven consecutive interviews producing no new codes. For RQ3, all 12 consequence codes were identified by interview 8, with the final four interviews yielding no new codes.

3.3 Data Analysis

We used thematic analysis [6] to analyze the qualitative data, as it provides a flexible yet rigorous approach for identifying, organizing, and interpreting patterns of meaning across a dataset. This approach was considered appropriate given the exploratory nature of the study and its focus on understanding participants' perceptions and experiences regarding PS. We adopted an inductive strategy, allowing themes to emerge directly from the data rather than from pre-existing theoretical frameworks. Analysis was conducted at an explicit (semantic) level, focusing on what participants directly stated, without attempting to infer underlying assumptions.

The unit of analysis was the meaningful excerpt (utterance or clause-level segment), such that each identifiable statement expressing a perception, experience, or judgment related to PS was treated as a discrete codable unit. Transcripts were read multiple times to ensure deep familiarity with the data. Meaningful excerpts were then identified and assigned descriptive labels capturing the explicit semantic content of each statement. This initial coding was performed independently by two researchers. Disagreements were resolved through structured discussion: when the two coders assigned different labels to the same excerpt, they presented their rationale and negotiated a consensus label. In cases where consensus was not immediately reached, a third experienced researcher in qualitative analysis and human factors in SE served as a tiebreaker, reviewing the excerpt and the competing labels before a final code was assigned. No formal inter-rater reliability coefficient (e.g., Cohen's Kappa) was calculated; instead, we relied on a consensus-based approach, which is consistent with established practice in constructivist thematic analysis [6], where the goal is interpretive agreement rather than statistical concordance. Table 2 shows an example of the initial coding of the quotes and labels, while the mapping to subcategories and categories between levels of influence is shown in Table 3.

Table 2: Examples of enablers, barriers, and consequences of psychological safety

RQ	Code	Example Quote
RQ1 Enabler	Humane Leadership	"*I was lucky to have extremely humane bosses. It has a lot to do with the personalities of the people I work with. My boss treats me well, listens to me, and if when say something, he really listens, tries to understand, and if he thinks what I'm saying makes sense, he says 'go for it, it's worth it.' But if not, he doesn't make a big deal out of it, it's fine. (P4)*"
RQ2 Barrier	Fear of relational consequences	"*I also avoid pointing out whose fault it is. The person made a mistake once; I usually correct it myself. But I end up not telling them directly, mainly because I'm afraid of damaging our relationship and making myself look mean. (P6)*"
RQ3 Consequence	Rework	"*We ended up in a situation where we did a lot of work very quickly and later had a lot of rework, because we needed to keep fixing things. We couldn't cover all the validation scenarios because everything had to be done fast. (P1)*"

The resulting codes were subsequently grouped into subcategories and then aggregated into broader categories reflecting recurring patterns within the dataset. This grouping process was also conducted independently by the same two researchers, followed by a consensus meeting to refine subcategories and categories. An experienced researcher reviewed all codes, subcategories, and categories to confirm consistency.

To organize the findings analytically, all codes were mapped onto four levels of influence (individual, team, leadership, and organizational), drawn deductively from established multilevel frameworks of PS [9, 10]:

- **Individual level**: personal traits, skills, and behaviors affecting participation, confidence, and engagement.
- **Team level:** collective norms, interactions, and practices that shape communication, collaboration, and mutual support.
- **Leadership level:** behaviors, management styles, and skills of leaders that foster trust, openness, and learning (or, conversely, that act as barriers).
- **Organizational level:** policies, culture, processes, and institutional practices that create a safe environment (or hinder it) for professional vulnerability.

These levels were introduced as an organizational scaffold during the grouping stage, after initial codes had been inductively derived from the data, rather than used as a priori coding categories. When a code could plausibly be assigned to more than one level, the research team discussed its primary locus of influence and reached consensus before final assignment. For instance, behaviors that originated in individual traits but manifested primarily through team interaction were assigned to the level at which their effect on PS was most directly observed.

"'

To identify **codes related to enablers and barriers**, we relied on participants' descriptions of factors that enhanced or hindered **PS**. Codes summarizing participants' experiences were inductively derived from the data, mapped to subcategories, and subsequently

Table 3: Examples of subcategories and categories associated with psychological safety codes

Category	Subcategory	Code
Leadership	Accessible and empathetic managers	Humane leadership Leader open to new ideas
Individual	Fear of interpersonal consequences	Fear of relational consequences Fear of hindering a colleague's performance
Organization	Rework	Rework

aggregated into their corresponding levels of influence (Individual, Team, Leadership, and Organizational). For instance, enablers such as `humane leadership` were coded under the subcategory `accessible and empathetic managers` and grouped within `Leadership`. Similarly, barriers like `aggressive communication` or `exacerbation of errors` were classified under the subcategory `inadequate leadership behaviors`, also grouped within `Leadership`.

To **derive codes related to consequences**, we considered both the valence and the level of impact emerging directly from participants' reports. Valence was determined based on participants' perceptions: situations associated with insecurity or inhibited participation were categorized as negative, whereas those reflecting safety and confidence were categorized as positive. Consequences were then mapped to their corresponding levels of influence: either **individual-level consequences** (e.g., physical or mental health, engagement, work attitudes) or **organizational-level consequences** (e.g., project delays, rework, financial costs, loss of talent, higher-quality outputs).

4 Results

In this section, we present the findings for each research question.

4.1 Demographic Overview

Table 4 summarizes the participants' demographic data, including company type, years of experience, team size, work modality, gender, and role.

Participants' years of experience ranged from 2 to 35 years. The sample included professionals working in startups (4 participants), medium-sized companies (2), and large companies (6), as self-reported by participants. Team sizes ranged from 4 to 70. Ten participants worked remotely, one worked in a hybrid arrangement, and one worked on-site. In terms of gender, seven participants were male, while five were female.

From a career stage perspective, participants were distributed across three phases: the exploration stage (up to 5 years of experience, 4 individuals), the establishment stage (6 to 15 years of experience, 6 individuals), and the mid- to late-career stage (over 15 years of experience, 2 individuals). Participants held various roles, including back- and front-end developers (5), software engineers (3), product owners (2), one chapter lead, and one software director.

Participants were engaged in projects across different application domains, including financial systems, insurance systems, food delivery systems, and digital platforms, covering a variety of organizational contexts. All participants are from the same country, representing a range of career stages, organizational roles, team sizes, and application domains, providing a broad perspective on software development practices.

4.2 RQ1: Factors that enable Psychological Safety

In response to RQ1, our participants cited 16 factors that enable PS in software development environments. These enablers span four interconnected levels: leadership, team dynamics, organizational context, and individual characteristics. Table 5 presents the enablers organized according to these levels. Below, we explain each factor per level.

Leadership level. Participants (P1, P2, P4-P9, P11, P12) emphasized leadership attributes that supported the expression of new ideas, requests for help, error resolution, and collaboration across roles such as product owner and software developer. `Accessible and empathetic managers` factor was particularly valued, as explained in: "*I had a very easygoing boss, he used to say that no matter what kind of problems I had, if I made a mistake, if I made the company lose millions, or anything like that, I just had to be open. I just had to be able to communicate* (P5)." `Technical expertise` factor also enhanced trust and confidence, as P1 noted: "*because leadership, he has the technical background, he started as a programmer, he understands, he comprehends, so we chat, define the new deadline, and move forward this way.*" Another factor that contributed for PS is leaders who acted as `mediators`, as recalled by P5: "*there were details that could only be resolved if I talked directly with the PO. And then, usually in the meetings I had alone with my Tech Lead, I would tell him that communication with the POs was not always easy, so I asked him for help.*" Finally, `solution-oriented management style` factor avoided blame and promoted collective improvement, as described in: "*The manager never blamed one specific person. He always looked for the solution: 'What is going wrong? How do we solve it?' Afterwards, if there was something to improve, we improved it as a team (P8).*"

Team dynamics level. Participants (P1, P3, P4, P6, P8-P12) identified team-related aspects (`open communication`, `mutual support`, `interpersonal respect`, and `smaller team size`) as factors that enable PS. These elements fostered authentic participation and facilitated integration across roles such as Product Owner, UX, and QA. `Open communication` factor was highlighted by P6: "*the people who bring the requirements, they bring the requirements, but they are very open to conversation, so the conversation with the product person feels like a peer-to-peer exchange. (...) there is room for me to question, to understand what has to be done and why it is being done this way.*" `Mutual support` factor strengthened collective responsibility, as P8 described: "*my experience was always very smooth, like, look, we have the responsibility as a team, everyone will solve it, and it does not matter who pushed the code.*" `Interpersonal respect` in interactions further reinforced confidence, as P4 emphasized: "*the mutual respect from everyone, everyone respects your moment to speak, and there is no such thing as if you say something wrong, you are mocked.*" `Smaller team size` was also mentioned as contributing to safer interactions and decision-making, although this factor was reported by a single participant (P3, team size = 6) and should therefore be interpreted as a preliminary observation rather than a pattern supported across the sample. As P3 explained: "*And*

Table 4: Participants' demographics

Participant (P)	Company Type	Years of Experience	Team size	Work Modality	Gender	Role
P1	Startup	9	5	Remote	Male	Software Engineer
P2	Large	26	50	Remote	Male	Chapter Lead
P3	Startup	3	6	Remote	Male	Software Engineer
P4	Large	3	40	Remote	Female	Software Engineer
P5	Large	9	5	Remote	Male	Developer
P6	Startup	8	6	Remote	Male	Frontend Developer
P7	Medium sized	35	70	Remote	Female	Software Director
P8	Large	4	5	Remote	Female	Backend Developer
P9	Startup	2	4	Remote	Female	Frontend Developer
P10	Large	4	13	Remote	Male	Product Owner
P11	Medium sized	6	5	On-site	Female	Product Owner
P12	Large	6	32	Hybrid	Male	Frontend Developer

Table 5: Enablers of psychological safety across levels (RQ1).

Level	Enablers
Leadership	Accessible and empathetic managers; Technical expertise; Leaders as mediators; Solution-oriented management style.
Team	Open communication; Mutual support; Interpersonal respect; Smaller team size.
Organizational	Culture of collaboration; Specific work process; Open dialogue culture; Well-being and monitoring initiatives; Diversity and inclusion programs.
Individual	Extroversion; Openness to feedback; Negotiation skills.

in this product the team was smaller, it was much easier for us to talk about any strategic or architectural decision."

Organizational context level. Participants (P1, P3-P8, P10, P12) highlighted institutional factors like a `culture of collaboration`, `specific work processes`, a `culture of open dialogue`, institutional initiatives for `well-being and continuous monitoring`, and `diversity and inclusion` as enablers of problem-solving, solution-seeking, and learning behaviors. `Culture of collaboration` factor was particularly relevant, as P7 explained: "*there is this very collaborative culture, you know, if you do not know something, someone else teaches you (...)*". `Specific work processes` factor such as pair programming and agile practices supported transparency and reduced the fear of mistakes, as P4 recalled: "*We practiced pair programming, we were doing things together and that gave me a lot of security, even to make mistakes.*" `Culture of open dialogue` factor was also important, as illustrated by P5: "*the company's posture with us is one of understanding (...) let's deliver with quality. If you think something is not good, let's discuss and change it.*" `Well-being and continuous monitoring` factor (e.g., training, anonymous surveys, and listening channels) reinforced institutional perceptions of support, as P6 described: "*since I joined the company, I have always seen initiatives to check how the teams are doing; every week we have an anonymous form to say how we are feeling about the work.*" `Diversity and inclusion` further strengthened belonging and legitimacy of participation, as P8 shared: "*This whole diversity thing, I can see other women as programmers, I can have an active voice and not be diminished. So I think this really strengthens my sense of safety, you know? I know I'm in a company where I can be heard.*"

At the same time, participants' reflections revealed that the benefits of diversity initiatives become especially salient when contrasted with the challenges of underrepresentation. P4 noted the defensiveness sometimes encountered when women occupy positions of authority: "*We women engineers are in the middle of this, with a lot of men (...) when you are in a position they think you are above. They get kind of defensive.*" P9 described feelings of insecurity and silence when joining an all-male team: "*At first, I felt very insecure because it was actually an all-male team, and that was a particularly difficult time for me. In fact, I stayed quiet for a long time because I was feeling stupid, impostor syndrome, right? I was not confident.*" Similarly, P11 associated her lack of confidence with a leadership style that overemphasized mistakes: "*It made me think about women in technology and how much harder it is for us. I kept asking myself, what am I doing here?*"

These accounts suggest that diversity and inclusion function as facilitators not merely through the presence of formal initiatives, such as women-only groups (P8), but also by counteracting experiences of marginalization and reinforcing a sense of legitimacy, representation, and voice within software teams.

Individual level. Lastly, participants reported personal traits as enablers (P2, P5, P6, P9, P10), such as `extroversion`, `openness to feedback`, and `negotiation skills`. `Extroversion` factor supported relationship building and trust, as P2 explained: "*For me, I am playful, I am talkative, so my personality helps me break the ice and build connections easily.*" `Openness to feedback` factor enhanced receptiveness and constructive exchanges, as emphasized by P5: "*I think it is more about the individuals themselves, each one involved. If people are open, they are more receptive to criticism.*" `Negotiation skills` factor contributed to conflict management and alignment with stakeholders, as highlighted: "*We manage to negotiate well with the client because it is normal for them to bring new items outside the bigger scope we had planned for the year. So we had agreed on, let's say, five features, and then the client says, let's add a sixth. I say no, then we have to remove one. This kind of negotiation made things easier because everything was openly discussed* (P10)."

4.3 RQ2: Barriers to Psychological Safety

In response to RQ2, our participants mentioned 9 barriers that undermine PS in software development environments, as shown in Table 6. Below, we explain each one per level.

Table 6: Barriers that hinder psychological safety across multiple levels (RQ2).

Level	Barriers
Leadership	Inadequate leadership behaviors; Lack of openness to dialogue.
Team	Low team openness; Insufficient support; Dismissal of opinions.
Organizational	Organizational pressures.
Individual	Limited technical confidence; Lack of experience; Fear of interpersonal consequences.

Individual level. Participants (P1-P6, P10, P12) reported that `limited technical confidence`, `lack of experience`, and `fear of interpersonal consequences` often inhibited their expression. In relation to `limited technical confidence`, P4 indicated that "*It is difficult to say something when you do not have 100% mastery. Sometimes it is better to stay quiet.*" This insecurity was intensified when the role demanded broad expertise, making it hard to admit gaps, as P2 noted: "*Given the position I reached, it is not always possible to indicate that you do not know something or that you cannot do it. Because that is exactly what your role requires.*" `Lack of experience` also constrained junior professionals, especially in complex projects, as P3 explained: "*I found it very difficult to suggest anything . . . it was my first project with the tool.*" Finally, `fear of interpersonal consequences` was found in: "*I also avoid pointing out whose mistake it is . . . mostly out of fear the relationship will be damaged* (P6)."

Team level. Barriers at the team level (P2-P5, P7, P9, P10) were linked to `low openness`, `insufficient support`, and `dismissal of opinions`, which hindered collaboration. Regarding `low openness`, P3 described: "*I found it very difficult to suggest anything . . . also because of the lack of openness from the other people, even the technical developers who were on the project.*" `Insufficient support` also emerged as a barrier, as exemplified by P5: "*If there is someone in the meetings that I judge to be more arrogant, for example, I do not ask for help.*" Remote work further compounded this, as P5 added: "*We have assets that make remote work extremely difficult . . . you go on Slack, send a Meet, and the person does not join . . . it is hard to ask for help.*" Finally, `dismissal of opinions` led participants to silence themselves, as P9 recalled: "*Before, I was interrupted a lot and the things I said were not validated. So I stopped talking, I went for a while without giving much opinion. Just accepting.*"

Leadership level. `Inadequate leadership behaviors` (such as inconsistency between leadership discourse and practice, aggressive communication, and public exposure of mistakes) and `lack of openness` are discussed by the participants (P1, P3, P6, P9, P11). P1 shared an example of `inadequate leadership behaviors`: "*A delivery had to be redone, and then he started saying a lot of things in front of everyone. Saying he did not trust anyone anymore, that he did not know what to do, and that he would start charging penalties on everyone.*" Others emphasized `lack of openness` to dialogue, as P11 reported: "*With my leadership, I really didn't feel there was openness to talk to her, to give feedback.*"

Organizational level. Participants (P1, P2, P6, P11) highlighted `organizational pressures` that could be external, such as client or market demands, or internal, including unrealistic promises and aggressive targets. These pressures often generated overload and insecurity within the team. For instance, P2 described external demands leading to `pressure`: "*A sales team goes there and promises something, promises the world, and we have nothing. You cannot tell a paying client, 'I cannot do this.' It is very complex.*" Similarly, P11 illustrated internal *pressures* and lack of tolerance for mistakes, stating: "*There was no room for errors, which I always pointed out in the NPS. No room for validation or real exchange. Everyone was always focused on the result. Where are the deliveries?*"

4.4 RQ3: Consequences of Psychological Safety

The participants reported 12 consequences of PS and its absence, which manifested at organizational, individual, and innovation levels. Table 7 outlines these consequences, distinguishing between positive outcomes that emerge when PS is fostered and the negative effects that result from its absence. We detailed them below.

Table 7: Consequences of psychological safety and its absence across levels (RQ3).

Level	Positive Consequences (Presence of PS)	Negative Consequences (Absence of PS)
Organizational	Faster deliveries; Higher product quality.	Project delays; Rework; Financial costs; Talent loss.
Individual	Increased satisfaction and motivation.	Harmed mental health and well-being; Reduced engagement and motivation.
Innovation	Openness to new ideas and collective innovation.	Repressed creativity; Reluctance to propose improvements.

Organizational impacts. The following consequences are based on participants' retrospective accounts and reflect their perceived associations between psychologically unsafe contexts and project outcomes; other organizational or project factors may have contributed to the situations described. With this in mind, participants attributed `project delays`, `rework`, `talent loss`, and `financial costs` to psychologically unsafe contexts. P1 described: "*Later, when we were already very delayed, we finally spoke up, saying that the deliveries were not properly defined, not properly planned (. . .) And because of that, the second delivery was almost two months late, leading to penalties for the company's leadership.*" P6 recalled a similar case: "*A manager suggested a new internal plan, but it eventually became the official plan. I did not feel safe questioning it, so I stayed quiet. We ended up doing a lot very quickly and then had a lot of rework, fixing things we had rushed through.*". Several participants emphasized that ignoring team members' technical input had severe human, organizational, and financial consequences. P2 recounted: "*I said again: it will not work, down the road we will face a bigger problem. Then when the problem happened, leadership said: you warned us, right? We spent 400,000 reais more because of that coordinator's decision. The team worked 18 extra hours in total. People went to bed at 4 a.m. to start again at 9 a.m.*" P1 similarly highlighted that silencing testing concerns led to extensive delays: "*He himself kept reiterating that we should not do testing so as not to waste time, and in the end, we lost another two months trying to do all the testing.*"

Mid-level leaders were also affected when their warnings were ignored. P3 reported: "*She [the team manager] would often point out* *problems in the process, problems that management chose to ignore, not pay attention to, right? And then the bomb went off, because of the error she pointed out, it exploded in her lap.*" Beyond operational inefficiencies, the disregard for technical voices also had human costs, such as stress and interpersonal conflict. As P2 described: "*Imagine the state this team was in; it started generating fights, stress, people fighting among themselves because of the pressure. (…) with all these overtime hours, things were generated about overtime, and they were wrong, because people were tired and stressed.*"

By contrast, participants reported that psychologically safe environments were associated with `faster deliveries` (P8) and `higher quality` (P3, P5-P8, P11). As P8 explained: "*I had this help, and I can give help to others, even from different teams. So deliveries get faster because people feel more confident next time.*"

Individual impacts. Psychological unsafety directly `harmed mental health and well-being` (P1, P3, P6, P11), leading to burnout, anxiety, and lifestyle changes. P3 described: "*A moment when I stopped talking completely, so I went from my second week until the end of my six months in this experience without saying anything about suggestions to management. (…) It was a time when I slept very little. So, I always slept very little, I stopped doing physical exercise, so the time I had to do physical exercise, I could not use because I had to deliver something by 8 a.m. the next day. Physically, I started to show several symptoms of burnout and such, I started to gain a lot of weight too.*" Unsafety also `reduced engagement and motivation` (P2, P3, P5, P9, P11). As P5 put it: "*I ended up participating less, giving fewer opinions, and just doing the basics because I knew people were not interested.*" Conversely, safe contexts fostered `motivation and satisfaction` (P3, P4, P11), as P3 recalled: "*It was one of the most enjoyable projects to work on because we could prioritize together, discuss what needed to be done first, and it was a great experience.*"

Innovation impacts. Unsafety `suppressed creativity` and `discouraged proposing improvements`. As P5 explained: "*Why would I look for a better solution if no one wants to listen? These things just create friction and obstacles.*" By contrast, safe contexts encouraged `openness to new`, even initially rough, ideas that could evolve collectively into innovations, as pointed out by P4: "*Psychological safety always opens space for people to bring new ideas without fear of exposing their thoughts.*"And P7 added: "*Even when they say something absurd, if it makes sense, they learn from it (…) sometimes an incipient idea materializes.*"

5 Discussion

This section presents how we organized our findings and compares them with the technical literature.

5.1 Organizing the Findings

The complexity of PS in software development calls for a multidimensional perspective. Figure 1 synthesizes our findings by organizing the identified enablers, barriers, and consequences across four interconnected levels: individual, team, leadership, and organization. This framework illustrates how PS emerges from the interaction of personal, social, and organizational factors rather than from isolated conditions.

For practitioners, our findings suggest that fostering PS requires coordinated actions across these levels. Leaders should cultivate empathy, accessibility, and technical credibility to promote trust and open communication, while organizations should reinforce these efforts through structural mechanisms such as anonymous feedback channels, well-being initiatives, and diversity and inclusion programs. Teams can strengthen PS by encouraging open communication, mutual support, and collaborative practices such as pair programming and retrospectives. Recognizing early signs of psychological unsafety, including silence and disengagement, may also help organizations prevent burnout, turnover, and reduced collaboration.

For researchers, our findings highlight the need to further investigate how PS develops across socio-technical systems and organizational contexts. Future studies should examine its longitudinal and cross-cultural dynamics, explore its relationship with human sustainability, and develop SE-specific measurement instruments capable of assessing PS in technical, distributed, and collaborative environments. Another promising direction is to investigate leadership behaviors that foster psychological safety in software teams. Although leadership consistently emerged as a key enabler in our synthesis, little is known about the specific leadership practices that are most effective in software development teams. In addition, future research should investigate how psychological safety manifests within software engineering practices. Collaborative activities such as code reviews, pair programming, architectural discussions, retrospectives, and incident response are fundamental to software development, yet limited evidence exists on how these practices shape, reinforce, or hinder psychological safety in everyday software engineering work.

5.2 Comparison with Related Work

We begin by foregrounding the findings that most clearly extend the current understanding of PS in SE.

First, our findings suggest that participants' experiences of PS are shaped by gendered dynamics in software development environments. Voice was described as being constrained by gender, particularly in interactions involving authority, recognition of technical expertise, and peer evaluation. While prior SE research has acknowledged diversity-related challenges [19], it has not explicitly examined how gendered dynamics influence everyday speaking-up behaviors. Our findings therefore highlight PS as a situated and potentially unequal phenomenon, with important implications for inclusion and participation in software teams.

Second, participants associated low PS with concrete technical, economic, and human consequences, including rework, delivery delays, increased operational costs, anxiety, emotional exhaustion, and burnout. These findings broaden the understanding of PS beyond its established role in learning and innovation, suggesting that software professionals perceive it as closely related to organizational performance and human sustainability, understood here as the preservation of professionals' long-term health, engagement, and capacity to contribute sustainably to software work.

Taken together, these findings portray PS as a context-dependent phenomenon whose unequal distribution and tangible consequences are closely intertwined in everyday software development.

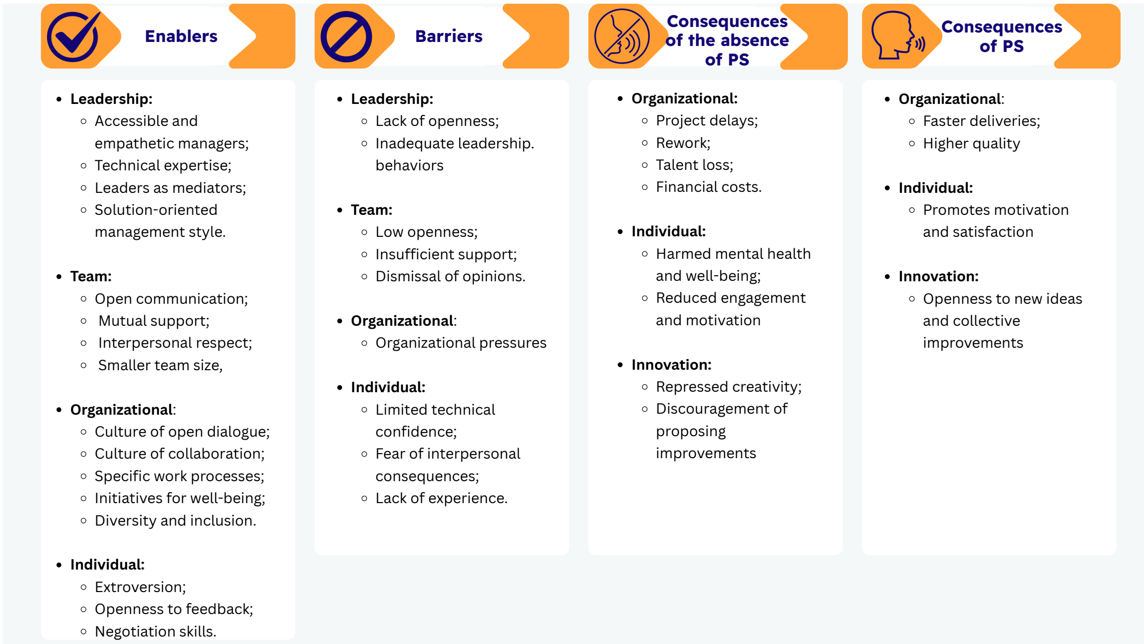

Figure 1: Summary of the main findings regarding psychological safety in software development environments.

To further situate our contribution, it is important to clarify how our findings differ from prior conceptualizations of PS in SE. Existing studies have predominantly framed PS in terms of distal antecedents, such as leadership support or team climate, which describe general conditions associated with the emergence of PS [1, 3]. Systematic reviews confirm this pattern: Newman et al. [16] found that primary studies emphasize leadership inclusiveness, peer support, and organizational culture as key antecedents, while Santana et al. [20] noted that few studies offer detailed insights into the concrete enablers that sustain PS or the barriers that undermine it in everyday SE practice. In contrast, our findings enable the identification of proximal mechanisms, that is, concrete and observable interactions through which participants described PS as being enacted, sustained, or inhibited in everyday practice. For instance, while "supportive leadership" is often described as an antecedent, participants consistently described how specific leader behaviors, such as responding constructively to mistakes, acknowledging contributions, or demonstrating technical competence, shaped whether they felt safe to speak up. This distinction advances a more fine-grained understanding of how PS materializes in socio-technical work settings.

These contributions are reinforced and refined by prior work that has identified leadership as a central factor influencing PS [1, 3, 7]. Our findings align with this body of research in showing that accessible and non-punitive leaders foster open communication. However, we extend this perspective by demonstrating that, in SE contexts, recognized technical expertise emerged as an important enabling condition. This suggests that inclusiveness alone may be insufficient in highly technical environments, where credibility appears to play a central role in enabling voice, complementing Edmondson and Lei [9]'s notion of inclusive leadership.

At the organizational level, prior studies have emphasized the importance of supportive climates and organizational culture [1]. Our findings build on this perspective by explicitly identifying institutional mechanisms, such as anonymous feedback channels, well-being initiatives, and diversity and inclusion programs, as structural enablers of PS. Participants also described diversity-related initiatives as protective spaces that fostered voice and belonging for underrepresented groups, reinforcing the role of organizational arrangements in mitigating unequal participation.

At the team level, our results corroborate prior evidence that trust-based relationships, open communication, and mutual respect are essential for fostering PS [1, 3]. However, rather than treating these elements as static conditions, our findings suggest that they are continuously negotiated through everyday interactions, such as collaborative problem-solving, peer feedback, and shared decision-making practices. This reinforces the view of PS as an emergent and dynamic property of team interactions.

At the individual level, existing literature has acknowledged the role of personal traits in shaping PS [10]. Our findings support this view, showing that characteristics such as extroversion and openness to feedback influence individuals' willingness to speak up. Importantly, however, our results indicate that individual agency operates within structural and relational constraints, suggesting that personal disposition alone is insufficient to ensure voice in environments where PS is compromised.

In addition to enablers, our study contributes by explicitly identifying barriers that actively undermine PS in software development environments. Our findings regarding barriers to PS converge with prior evidence from healthcare education, where hierarchical dynamics, fear of negative evaluation, and limited opportunities for supportive dialogue have been shown to inhibit speaking up [14].

Participants similarly reported that inadequate leadership behaviors, low team openness, and concerns about interpersonal consequences constrained voice in software teams. These findings are consistent with evidence from other high-stakes domains, such as healthcare and aviation, where fear of speaking up and hierarchical constraints inhibit communication [8, 14]. Furthermore, our findings suggest that, in the participants' accounts, such barriers are not confined to training or simulation contexts, but also emerge within continuous, real-world development processes, positioning PS as a structural feature of everyday collaboration rather than merely a situational phenomenon.

Finally, our findings expand the understanding of the perceived consequences of PS in SE. While prior research has primarily framed PS as a precondition for learning, experimentation, and innovation [20], participants described its influence as broader and more immediate. They associated psychologically safe environments with greater technical efficiency, motivation, and collaboration, while describing psychologically unsafe environments as contributing to rework, delivery delays, financial losses, reduced motivation, and harm to professionals' health and well-being. These findings extend existing frameworks by providing empirical evidence of how software professionals perceive the technical, economic, and human implications of PS in their everyday work.

6 Threats to Validity

We designed this study to align with recognized qualitative research quality criteria, namely **credibility**, **resonance**, **usefulness**, and **transferability**, as outlined in the *Empirical Standards for SE Research* [18].

To enhance **credibility**, two researchers independently coded the data and refined the coding through iterative discussions, while an experienced qualitative researcher reviewed all codes and categories. Researcher reflexivity was supported through reflective journals, peer debriefing, and the complementary expertise of the first author in SE and PS and the second author in human aspects. We also provide direct quotations and a replication package containing the interview protocol and coding scheme (Section 7).

Potential biases include interviewer influence, socially desirable responses, self-reporting, and the online interview setting. Interviews were conducted in the participants' native language to reduce linguistic and cultural bias. Although IQ1 elicited situations in which participants hesitated to speak up, IQ1.2 and IQ4 explicitly invited participants to describe enabling conditions and practices that could foster PS, partially balancing this emphasis on compromised PS.

Snowball sampling may have introduced homogeneity because referrals often originate from similar professional networks. To mitigate this risk, recruitment started from nine independent seeds spanning different companies, roles, and career stages. Because the study was exploratory, we intentionally investigated participants' experiences rather than measuring PS levels, and our findings should therefore be interpreted as participants' perceptions rather than comparisons between high- and low-PS environments.

As the interviews relied on retrospective accounts, they are subject to **recall bias**. Participants often referred to different projects and situations across the interview, reflecting the episodic and context-dependent nature of PS rather than inconsistencies in their accounts.

To strengthen **transferability**, participants were intentionally diverse in role, experience, and organization type. Nevertheless, all participants were from the same country, and 83% worked remotely. Although P11 and P12 (on-site and hybrid) were interviewed after theoretical saturation had been reached, whether PS operates similarly across work modalities remains an open question [25]. While the sample size ($N = 12$) does not support statistical generalization, theoretical saturation supports an in-depth understanding of the phenomenon and provides practically useful insights across multiple software development contexts.

7 Conclusion

This study investigates the enablers, barriers, and consequences of PS in SE. Our findings show that PS in software development is shaped by leadership, organizational practices, team dynamics, and individual characteristics, which can operate both as enablers and barriers. Psychologically safe environments foster innovation and well-being, whereas unsafe ones often lead to rework, delays, financial losses, and health issues.

Future research may examine PS across varied contexts (e.g., startups vs. large enterprises, distributed vs. co-located teams), investigate the role of specific roles and functions (such as product owners or technical leads), and employ longitudinal designs to capture how PS evolves throughout project lifecycles. Cross-cultural and diversity-oriented studies are also necessary to broaden understanding within global software development settings.

In conclusion, our findings suggest that PS functions as an important condition for the sustainability of SE work. Ensuring that professionals feel safe to contribute their expertise openly appears essential not only for improving product quality and organizational performance but also for protecting the well-being of those who build the software systems upon which modern society increasingly depends.

Artifact Availability

The interview files, the coding schema, and the complete data set are available on Zenodo, archived under DOI 10.5281/zenodo.21068327.

Acknowledgements

ChatGPT was used to support language editing and improve clarity. All content and interpretations remain the authors' responsibility. We thank all participants for sharing their experiences. 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] Muhammad Ovais Ahmad. 2023. Psychological safety, leadership and non-technical debt in large-scale agile software development. In *2023 18th Conference on Computer Science and Intelligence Systems (FedCSIS)*. IEEE, 327–334.
[2] Adam Alami, Oliver Krancher, and Maria Paasivaara. 2022. The journey to technical excellence in agile software development. *Information and Software Technology* 150 (2022), 106959.

[3] Adam Alami, Mansooreh Zahedi, and Oliver Krancher. 2023. Antecedents of psychological safety in agile software development teams. *Information and Software Technology* 162 (2023), 107267.

[4] 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.

[5] Marte Pettersen Buvik and Anastasiia Tkalich. 2022. Psychological Safety in Agile Software Development Teams: Work Design Antecedents and Performance Consequences. In *Proceedings of the 55th Hawaii International Conference on System Sciences (HICSS)*. Hawaii, USA, 10. doi:10.24251/HICSS.2022.880

[6] Victoria Clarke and Virginia Braun. 2013. Teaching thematic analysis: Overcoming challenges and developing strategies for effective learning. *The psychologist* 26, 2 (2013).

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

[8] Amy C Edmondson. 2018. *The fearless organization: Creating psychological safety in the workplace for learning, innovation, and growth.* John Wiley & Sons.

[9] 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.

[10] M Lance Frazier, Stav Fainshmidt, Ryan L Klinger, Amir Pezeshkan, and Veselina Vracheva. 2017. Psychological safety: A meta-analytic review and extension. *Personnel psychology* 70, 1 (2017), 113–165.

[11] 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.

[12] William A Kahn. 1990. Psychological conditions of personal engagement and disengagement at work. *Academy of management journal* 33, 4 (1990), 692–724.

[13] André Kaiser, Eliezer Dutra, and Gleison Santos. 2025. Investigating How Psychological Safety Can Be Fostered in Agile Teams. In *Anais do XXXIX Simpósio Brasileiro de Engenharia de Software* (Recife/PE). SBC, Porto Alegre, RS, Brasil, 348–359. doi:10.5753/sbes.2025.9953

[14] Kelly Lackie, Kathryn Hayward, Caitlyn Ayn, Peter Stilwell, Jennifer Lane, Cynthia Andrews, Tanya Dutton, Doug Ferkol, Jonathan Harris, Shauna Houk, et al. 2023. Creating psychological safety in interprofessional simulation for health professional learners: a scoping review of the barriers and enablers. *Journal of Interprofessional Care* 37, 2 (2023), 187–202.

[15] Lidivânio Monte, Beatriz Santana, Glauco Carneiro, Manoel Mendonça, Sávio Freire, and José Santos. 2025. Understanding Psychological Safety in Agile Software Development Teams. In *Anais do XXXIX Simpósio Brasileiro de Engenharia de Software* (Recife/PE). SBC, Porto Alegre, RS, Brasil, 416–426. doi:10.5753/sbes.2025.9974

[16] 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.

[17] 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.

[18] Paul Ralph, Nauman bin Ali, Sebastian Baltes, Domenico Bianculli, Jessica Diaz, Yvonne Dittrich, Neil Ernst, Michael Felderer, Robert Feldt, Antonio Filieri, Breno Bernard Nicolau de França, Carlo Alberto Furia, Greg Gay, Nicolas Gold, Daniel Graziotin, Pinjia He, Rashina Hoda, Natalia Juristo, Barbara Kitchenham, Valentina Lenarduzzi, Jorge Martínez, Jorge Melegati, Daniel Mendez, Tim Menzies, Jefferson Molleri, Dietmar Pfahl, Romain Robbes, Daniel Russo, Nyyti Saarimäki, Federica Sarro, Davide Taibi, Janet Siegmund, Diomidis Spinellis, Miroslaw Staron, Klaas Stol, Margaret-Anne Storey, Davide Taibi, Damian Tamburri, Marco Torchiano, Christoph Treude, Burak Turhan, Xiaofeng Wang, and Sira Vegas. 2020. Empirical Standards for Software Engineering Research. arXiv:2010.03525 [cs.SE] https://arxiv.org/abs/2010.03525

[19] Gema Rodríguez-Pérez, Reza Nadri, and Meiyappan Nagappan. 2021. Perceived diversity in software engineering: a systematic literature review. *Empirical Software Engineering* 26, 5 (2021), 102.

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

[21] Beatriz Santana, Lidivânio Monte, Bianca Santana de Araújo Silva, Glauco Carneiro, Sávio Freire, José Amancio Macedo Santos, and Manoel Mendonça. 2025. Psychological safety in software workplaces: A systematic literature review. *Information and Software Technology* 187 (2025), 107838. doi:10.1016/j.infsof.2025.107838

[22] 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.

[23] Edgar H Schein and Warren G Bennis. 1965. *Personal and organizational change through group methods: The laboratory approach.* Wiley New York.

[24] Irving Seidman. 2006. *Interviewing as qualitative research: A guide for researchers in education and the social sciences.* Teachers college press.

[25] Anastasiia Tkalich, Darja Šmite, Nina Haugland Andersen, and Nils Brede Moe. 2024. What Happens to Psychological Safety When Going Remote? *IEEE Software* 41, 1 (2024), 113–122. doi:10.1109/MS.2022.3225579

[26] Stewart A Vella, Elizabeth Mayland, Matthew J Schweickle, Jordan T Sutcliffe, Desmond McEwan, and Christian Swann. 2024. Psychological safety in sport: A systematic review and concept analysis. *International Review of Sport and Exercise Psychology* 17, 1 (2024), 516–539.