“Frustrating, Stressful, and Overwhelming”: Insights into Software Practitioners’ Productivity from Stack Exchange Discussions

Júlio César Andrade Silva
Technology Department
State University of Feira de Santana
Feira de Santana, Bahia, Brazil
juliocesar.andradee@gmail.com

Lidiany Cerqueira
Computing Institute
Federal University of Bahia
Salvador, Bahia, Brazil
lidiany.cerqueira@ufba.br

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

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

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

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

ABSTRACT

Context: Productivity is a central concern in software engineering, but traditional approaches often overlook the human aspects of daily work. **Aims:** To investigate how software practitioners perceive productivity challenges and how they impact their well-being and their tasks. **Method:** We conducted a thematic analysis of grey literature, drawing from discussions in two Stack Exchange communities (Software Engineering and Project Management). **Results:** We identified 21 factors that affect productivity, emphasizing time management, adopting new standards and technologies, difficulty measuring productivity, and context switching. These challenges reflect technical issues and human concerns such as demotivation and psychological pressure, showing that productivity cannot be understood in isolation. The study also revealed 17 direct impacts on professionals, including stress and perceived fluctuations in productivity, and 8 impacts on tasks, such as delays and rework. **Conclusion:** This study offers a broader understanding of productivity in software engineering by connecting technical and human factors. It highlights the value of grey literature in uncovering underexplored dimensions of practitioners’ experiences.

KEYWORDS

Productivity, Grey Literature, Software engineering, Non-technical factors

1 Introduction

The intense competition in the software industry challenges companies to deliver high-quality products within increasingly shorter deadlines. To remain competitive, development teams must enhance performance and reduce production costs [26]. In this context, productivity is crucial to business success, serving as a key factor in generating products and services [43].

Given the strong influence of human factors in Software Engineering (SE) activities, developers’ productivity is critical for minimizing software production costs [6]. However, achieving good results requires implementing effective practices based on the factors influencing productivity [44]. The challenge lies in the complexity of understanding and identifying these factors [6]. The socio-technical nature of SE further complicates the analysis of productivity-related influences in software projects [15]. This complexity directly impacts the industry, leading to challenges in cost estimation and developers’ satisfaction [21].

Productivity in software development has been primarily researched to facilitate decision-making in organizations [5, 6]. Although existing literature extensively focuses on measuring productivity, there is still a gap in understanding how developers perceive and evaluate their productivity [24]. Studying productivity gains additional relevance as it relates to project success and practitioners’ well-being and satisfaction [15].

Grey literature (GL), which includes unpublished materials not found in traditional channels such as peer-reviewed academic journals [18], constitutes an underexplored data source for understanding productivity-related challenges. Different sources of GL have been used to explore SE topics [7, 8]. Q&A platforms, such as Stack Exchange[1], have proven to be a valuable resource, offering a rich space for sharing experiences and solutions to issues faced by professionals in the SE field [1, 3]. Analyzing these sources can expand the reach of research, enabling the inclusion of updated data and more diverse approaches to emerging problems such as code smells, interpersonal skills, and technical debt [16, 20, 25, 29, 36, 37, 41].

This study focuses on the human factor influencing software practitioners and uses GL as a data source, aiming to capture practitioners’ perspectives on productivity in software development. Including this data connects academic knowledge with practice,

[1]https://stackexchange.com/

filling gaps in understanding the factors influencing practitioner productivity and offering a broader view of how individual and contextual variables impact this process. By prioritizing practitioners' experiences and perceptions, this work seeks to contribute to developing new strategies to optimize productivity and bridge the gap between academia and practice, benefiting the SE community.

This study offers an in-depth analysis of the productivity challenges faced by software practitioners based on 91 discussions of two Stack Exchange communities (Software Engineering and Product Manager). The main goal is to investigate the factors that lead to these challenges, their impacts on practitioners, and the tasks they perform.

The main contributions of this work include (i) identifying 21 factors influencing productivity, (ii) recognizing 17 direct impacts on practitioners and 10 direct impacts on the tasks, (iii) releasing a curated dataset of annotated Stack Exchange discussions to support future research on software practitioners' productivity, and (iv) building a conceptual framework that correlates productivity factors with their impacts on professionals and tasks.

The remainder of this paper is composed of six sections. Section 2 explores the context of productivity in SE. Section 3 presents the adopted research method. Section 4 presents the main findings of the study. Section 5 discusses the results. Section 6 considers the limitations of the research. Section 7 concludes the article, highlighting the main contributions and suggesting directions for future investigations.

2 Background

This section discusses the concept of productivity and its application within software development, followed by an analysis of the factors influencing productivity in this context.

2.1 Productivity

The definition of productivity emerged in 1950 and generally refers to the relationship between the number of goods and services produced and the resources required for their production [17]. Although there is no consensus on a single definition, many scholars consider it the relationship between the products generated and the resources (human or non-human) used during the development process [10]. However, this definition may be simplistic, as productivity can vary depending on the application area. For an engineer, productivity may relate to the number of products produced per work time, while for an accountant, it may be the relationship between profit and investment [31].

In a broader context, many organizations focus on the productivity of their employees, especially in areas such as software development, where productivity is associated with the intellectual work of professionals and not just with machines or other material resources [32].

Productivity in software development is a complex and challenging topic discussed by several scholars over the decades. Boehm [4] highlighted the influence of human, technical, and organizational factors due to the creative and intellectual nature of the work. Traditionally, productivity was defined as the relationship between the amount of software produced and the work required for its production [6]. However, metrics such as lines of code (LOC) written per hour and the number of function points implemented have become insufficient, as they do not adequately capture the value delivered to end users [15].

Instead of focusing on quantity, productivity in SE should consider the functional value delivered to the customer and the impact of the software in the context where it is used [34]. This led to an evolution in the definition of productivity, considering not only the inputs and outputs of the process but also qualitative aspects such as innovation, quality, and product relevance.

Thus, productivity in software development can be defined as the relationship between the functional value of the software produced and the costs or efforts involved in its production [42]. The costs can include personnel and technological resources, and the impacts on the quality of life of the professionals involved [34]. Additionally, traditional measurements such as the number of LOC often prioritize quantity over the quality of the final product, which can lead to a distorted view of productivity [28].

2.2 Factors Influencing Productivity in Software Development

Understanding factors that affect productivity in software development is crucial for organizations. Still, these factors can vary according to the context, characteristics of developers, and the type of project [10]. Identifying and analyzing these factors is not a simple task [35]. Studies such as those by Scacchi [38] and Clincy [10] listed several technological and non-technological factors that impact productivity. Technological factors include the programming language, project complexity, and tools used. Non-technological factors, such as team motivation, communication, and decision-making freedom, also play a significant role [12].

Oliveira et al. [27] defined three categories that can be used to group the factors affecting productivity in software development, as follows:

- **Human Aspects:** Factors related to the characteristics of the team members, such as motivation, communication, professional experience, etc.
- **Aspects Related to Software Products:** Factors related to software characteristics, such as programming language, size, and software complexity, among others.
- **Aspects Related to the Development Process:** Factors related to guidelines and procedures to be followed by the organization, such as software development methodology.

Each organization needs to create its own model for measuring productivity, considering both technological and non-technological aspects, such as the experience of software engineers. Moreover, each development team should carefully evaluate the factors impacting productivity in its environment, avoiding adopting factors used in other contexts without a thorough analysis [30].

3 Research Method

This section outlines the research strategy followed in this study, including research questions, data collection strategies, and analysis techniques. The goal is to ensure transparency and methodological rigor, enabling replication and validation of the results.

3.1 Research Questions

To investigate how various factors affect software practitioners' productivity and understand their perceptions on the topic, we propose the following research questions (RQs):

- **RQ1: What productivity challenges are software practitioners facing?**
  This question aims to identify the main productivity challenges practitioners encounter, exploring their origins and impacts on daily tasks. Analyzing these challenges will provide a deeper understanding of the underlying factors contributing to their occurrence.
- **RQ2: How do productivity challenges affect software practitioners?**
  The goal of this question is to understand how productivity challenges influence software practitioners, considering their impacts on health and well-being.
- **RQ3: How do productivity challenges affect software practitioners' work?**
  This question investigates how these challenges directly impact work activities and outcomes, analyzing the consequences for task execution and delivery quality.

3.2 Data Collection

To answer the research questions, we conducted a thematic synthesis [11] in grey literature [22], specifically discussion forums. We choose GL because, in these environments, software practitioners feel more comfortable expressing their opinions and feelings about problems or everyday situations [2, 20].

For this study, we selected two communities from Stack Exchange: **Software Engineering**[2] forum and **Project Management**[3] forum. Stack Exchange communities are structured around a central question, followed by answers and comments on both the question and the answers. Additionally, discussions are categorized using tags provided by the question's author and are assigned scores by users to indicate their relevance. We choose these communities as they bring together professionals, academics, and students involved in the software development cycle, providing a rich environment for practical discussions. Additionally, these communities are complementary. For instance, while users on the Software Engineering forum have discussed topics related to software engineering in general, discussions about project management are more commonly found on the Project Management forum.

We collect data using the Stack Exchange Data Explorer (SEDE), which allows querying and extracting information from forum posts, comments, and metadata. To refine the search, we used a string containing the terms *"productivity"* and *"development performance"*, applied to the title, tags, and body of posts. These terms were selected based on preliminary tests and sample analysis, ensuring greater accuracy and relevance in the results.

Inclusion criteria (IC) were defined based on the questions that initiated the discussions to identify the most relevant and high-quality discussions. These criteria (see Table 1) guided the data extraction in SEDE. They were adapted from the study by Gomes et al. [19, 20], with the addition of a temporal criterion regarding the creation date of the questions (IC5 described in Table 1). Discussions that did not meet the inclusion criteria were excluded.

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

Table 1: Inclusion Criteria

ID	Criterion
IC1	The discussion must have at least two posts (a question and an answer).
IC2	The discussion must involve at least two people (the author of the question cannot be the same person who answered it).
IC3	The question's score (upvotes) must be greater than zero.
IC4	The discussion must be related to productivity.
IC5	The question's creation date must be on or after 01/01/2012.

Based on the search string and these criteria, **211 questions** were extracted from the **Software Engineering** forum and **65 questions** from the **Project Management** forum, totaling **275 questions**. These questions then proceeded to the selection process.

3.3 Data Selection

The selection stage was structured to prioritize the most relevant discussions. A preliminary analysis of the 275 questions selected in the data collection phase was conducted to achieve this[4]. The goal was to avoid choosing a random sample, considering time and human resource constraints, as this could compromise the representativeness of the analyzed challenges. Instead, the selection process was designed to allow reviewers to focus on potentially positive cases, increasing analysis efficiency and ensuring a more precise focus on the topics of interest.

One reviewer initially classified each question into one of four categories[5]:

- *False Positive* – irrelevant to the study (e.g., questions not addressing productivity-related factors);
- *Positive* – clearly relevant, directly related to software practitioners' productivity, and included in the analysis;
- *Probably False Positive* – likely irrelevant, but with some uncertainty regarding its content;
- *Probably Positive* – likely relevant, but ambiguous enough to prevent a definitive classification at the time.

A second researcher then reviewed the classifications, adjusting the uncertain cases into either the *Positive* or *False Positive* categories. Questions that did not address software practitioners' productivity issues were ultimately classified as *False Positive* and excluded from the analysis.

As a result of this process, **68 questions** from the **Software Engineering** forum and **33** from the **Project Management** forum were initially selected as positive for analysis. However, during the qualitative analysis phase, **10 questions** from Software Engineering and **10** from Project Management were identified as false

[4]Two reviewers participated in this stage to ensure the quality and consistency of the selection.
[5]The intermediate categories were introduced to manage ambiguous cases where the primary reviewer did not have sufficient confidence to make a clear decision.

positives and excluded by the reviewer pairs, resulting in a final set of **58** and **23 valid discussions**, respectively.

3.4 Qualitative Data Analysis

Before starting the data analysis, we conducted a calibration step to ensure consistency and alignment among the reviewers. In the Software Engineering (SE) forum, four researchers analyzed the same set of three predefined discussions. This step allowed all reviewers to familiarize themselves with the analysis criteria and adjust their interpretations, minimizing potential individual biases and ensuring a more uniform evaluation.

The calibration step was also applied in the Project Management (PM) forum, where only two researchers were responsible for the analysis. The pair reviewed three initial discussions to ensure that both were aligned with the established criteria before proceeding to the complete analysis.

After completing the calibration in both forums, the following steps were carried out:

- **First Complete Reading of Discussions:** An initial reading of the discussions was performed. This step aimed to familiarize the reviewers with the content, immerse them in the data, and identify relevant information to answer the research questions.
- **Text Segment Extraction:** Two independent researchers extracted key text segments from each discussion based on their perceptions without using predefined codes. These segments were selected to address research questions RQ1 to RQ3, ensuring an initial data collection free from pre-established biases.
- **Consensus on Responses:** After analyzing all assigned questions, each pair of researchers compared their responses. Potential divergences were resolved in meetings, during which they presented their arguments and debated their analyses to reach a consensus. In cases where consensus was not possible, a senior third researcher was consulted to mediate the discussion and ensure the resolution of divergences, maintaining the consistency and reliability of the results.
- **Open Coding:** One researcher coded the responses for each RQ, generating descriptive codes that were later reviewed by a second reviewer. The process was iterative: codes were refined and merged until no new relevant elements emerged. We monitored saturation through iterative coding cycles, analyzing additional blocks of discussions until no new relevant codes emerged. This process was documented in a control spreadsheet, and disagreements were resolved by consensus or arbitration. Theoretical saturation was considered achieved when the data ceased to yield new codes, indicating a stable and comprehensive coding scheme. This approach was adopted to ensure that the identified factors and impacts were both comprehensive and representative.
- **Classification:** After refining the codes, the categorization process began. The primary goal of this step was to group the codes into broad and representative categories. Thus promoting an efficient information organization and enabling a more structured data analysis [11]. This wider approach allowed for exploring meanings and relationships among the analyzed elements and facilitated synthesizing the obtained information.

Figure 1 shows an example of the analysis process, illustrating the extraction of text segments to answer the three research questions of this study. From the text segment highlighted in orange, the code *Inadequate work environment* was identified for question RQ1 regarding challenges faced by practitioners. To answer question RQ2 (regarding how practitioners are affected by the challenges), the blue-highlighted segment was extracted, resulting in the codes *Stagnation* and *Cognitive overload*. For question RQ3, regarding how challenges affect the work, the green-highlighted excerpt evidenced the codes *Maintenance difficulty* and *Quality reduction*.

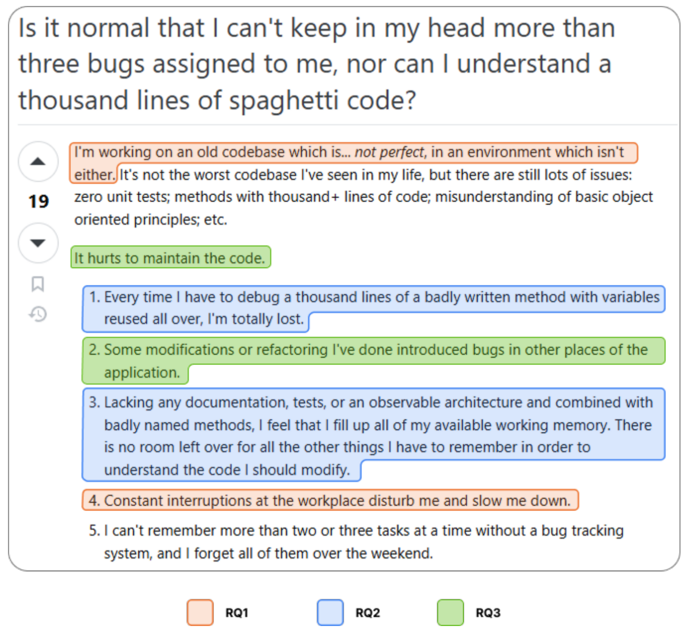

Figure 1: Example of an article analysis

4 Results

This section presents the findings from analyzing 58 discussions from the Software Engineering forum and 23 from the Project Management forum.

4.1 RQ1: What productivity challenges are software practitioners facing?

The analysis revealed 21 factors that impact practitioners' productivity, considering their perception. Multiple factors were often mentioned in a single discussion. We grouped these factors into four categories. Figure 2 illustrates the overall occurrence distribution of these factors into the categories. We detail the categories below. Additionally, we present one factor per category. The complete list with descriptions and examples is available in the supplementary material [40].

- **Organizational factors** refer to how work is structured and managed, including leadership, processes, and the work environment. For instance, **Management issues** is an organizational factor. It refers to leadership issues such as poor task distribution, lack of planning, and arbitrary decisions. As P14 notes: *“Sadly, upper management has a tendency to set deadlines without consulting the programmers, which has led to many an all-nighter.”*

- **Interpersonal factors** involve team dynamics, such as communication, collaboration, and relationships among team members. **Lack of skilled co-workers** emerged as an interpersonal factor. It concerns the difficulty of working with less experienced or technically outdated colleagues. P47 comments: *“How do you handle yourself in a new team where you are the senior most practitioner and most others in the team are junior to you by several years.”*
- **Personal factors** relate to individual aspects such as motivation, well-being, and the practitioner’s ability to maintain focus. **Context switching** is a personal factor. It refers to the difficulty of frequently shifting between tasks or projects, which hinders concentration and reduces productivity. P1 states: *“If I compare this to working on one small library or just some deep backend stuff, I find that my productivity basically non-existent, because I constantly switch around and find bugs in 50 different places, instead of just focusing on one.”*
- **Technical factors** concern technical expertise, skills, and the characteristics of the software being developed. For instance, **Technical debt management** emerged as a technical factor. It relates to the negative impact of accumulated technical debt, often resulting from short-term solutions. As P28 explains: *“I don’t expect zero tech debt. In this post, technical debt problem refers to the severity that has been causing a negative impact, say, productivity.”*

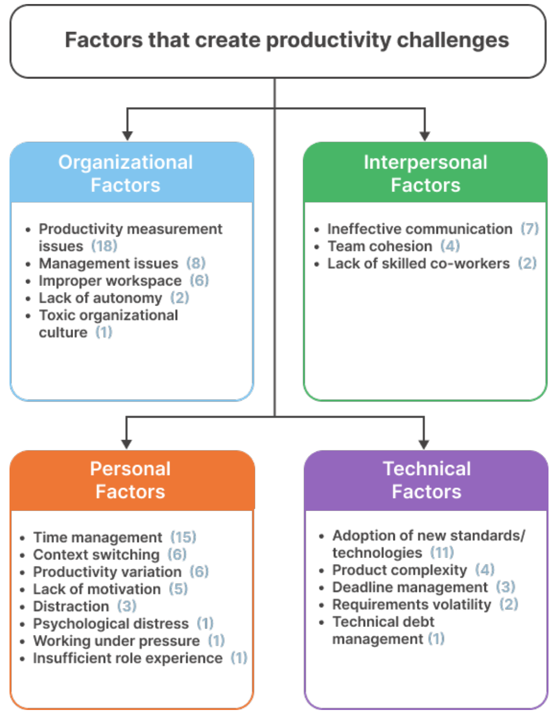

Figure 2: Distribution of productivity factors by category.

4.2 RQ2: How do productivity challenges affect software practitioners?

To address RQ2, we analyzed how practitioners describe their emotional and psychological responses to productivity challenges. Fifteen distinct types of impact were identified and grouped into three categories, as shown in Figure 3. Below, we present an example of one factor per category. Full descriptions and additional examples are available in the supplementary material [40].

- **Mental health issues** reflect emotional and psychological impacts such as frustration, stress, or a sense of unproductivity. For example, **Unhappiness** came up as a mental health challenge. It relates to a feeling of frustration and dissatisfaction with one’s performance. P32 expresses: *“Sometimes I’d spend a good part of the day without writing any code. This is making me unhappy since I don’t feel I’m productive enough.”*
- **Performance issues** include changes in motivation and behavior, such as difficulty concentrating or delaying tasks. **Procrastination** is an example of a performance issue. It is the tendency to delay tasks due to low motivation or disengagement. P23 comments: *“I can’t fix the issue with these people; I can only try to fix myself. I believe that issues come when they make me work on the same thing again and again, and I get demotivated to work, and I procrastinate.”*
- **Physical issues** refer to physical symptoms related to stress or work overload. **Physical pain** is categorized as a physical issue. It refers to the physical discomfort, such as headaches, related to pressure and technical complexity. As reported by P38: *“In the end this led to a bunch of spaghetti code, untestable tasks [...] and a bunch of headaches.”*

4.3 RQ3: How do productivity challenges affect software practitioners’ work?

To address RQ3, we investigated the practical consequences of productivity challenges on practitioners’ work output. Eight distinct types of impact were identified and grouped into operational and technical issues, as shown in Figure 4. Below, we present one factor per category. Detailed descriptions and examples are included in the supplementary material [40].

- **Operational issues** concern task delivery, time management, and workflow organization. **Delivery delays** is a operational issue. It occurs when the practitioner is unable to complete tasks on time. P51 reports: *“The result is often that I don’t deliver. Now I may know at the back of my head what I need to do, but I would appreciate your strategy on how to be the guy who ‘delivers’.”*
- **Technical issues** relate to code quality and product reliability, often affected by haste, multitasking, or lack of testing. **Reduced quality** is an example of a technical issue. It refers to producing lower-quality code prone to bugs, often due to overloaded schedules. P22 describes: *“At any given time I handle around 3-4 projects [...] As a result, many bugs escape*

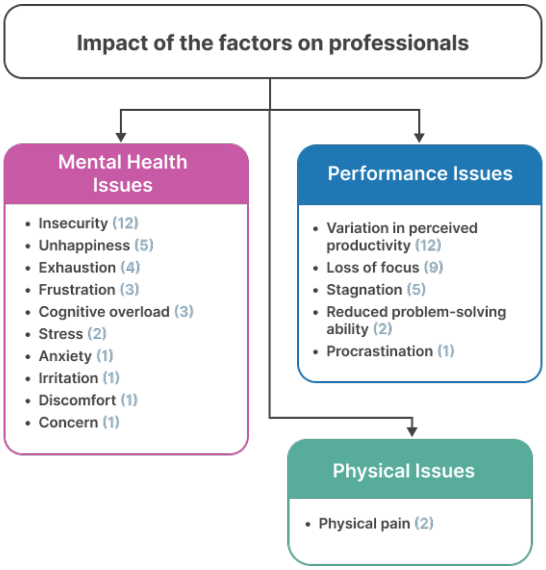

Figure 3: Distribution of impacts on professionals by category.

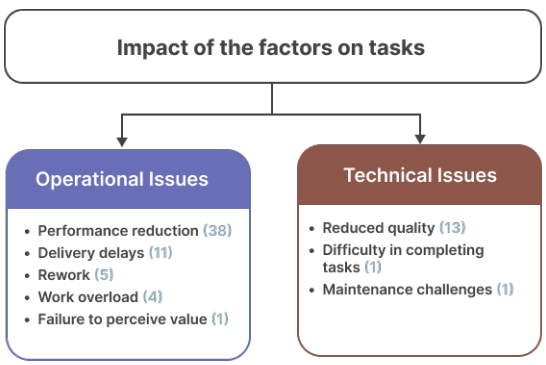

Figure 4: Distribution of impacts on tasks by category.

to production, which I have to fix and, in turn, sets back my other projects."

5 Discussion

In this section, we discuss the main findings of our study in light of the research questions. We present a conceptual map that organizes the relationships between the factors affecting productivity (RQ1), their impacts on professionals (RQ2), and their effects on work tasks (RQ3). We also compare the identified factors with related work and highlight the unique contributions of our study. Finally, we explore the practical and research implications derived from the results.

5.1 RQ1: What productivity challenges are software practitioners facing?

As expected, the productivity challenges faced by software practitioners and project managers involve both common factors and those specific to each role. While both groups deal with issues that impact team performance, discussions in the forums indicate that engineers tend to report challenges related to individual task execution, such as technical difficulties, concentration, and personal organization. In contrast, managers focus more on collective factors such as coordination, communication, and interpersonal conflicts. This combination of shared and distinct challenges reinforces the need for distinguished approaches to support productivity in technical and managerial contexts, respecting the particularities of each professional profile. This finding aligns with a recent survey at a large governmental software organization [9].

Among the factors analyzed, **time management** was the most discussed topic in the SE forum. Practitioners reported difficulty handling multiple tasks, setting priorities, and managing idle time, such as waiting for software tests. Initially, these challenges were grouped under the code *time fragmentation*. Still, refining the analysis led to the creation of two separate categories: **context switching**, referring to productivity loss due to switching between complex tasks, and **time management**, for mentions of inefficient use of available time. This distinction allowed a more accurate understanding of the impacts and underlying causes.

In contrast, the PM forum highlighted **productivity measurement issues** more prominently. This is a recurring issue in knowledge-based work, such as software development, where simplistic metrics like lines of code are widely criticized for failing to capture essential aspects of professional performance [15, 21]. Both managers and engineers expressed dissatisfaction with these metrics and sought more representative alternatives aligned with project realities.

Both communities also emphasized interpersonal factors, especially **team cohesion**, **lack of skilled co-workers**, and **ineffective communication** [13–15, 21]. Well-aligned teams are more productive, while communication failures hinder alignment, increase rework, and limit access to crucial information. Team diversity can enhance problem-solving and reduce burnout risks, provided responsibilities are well distributed in a collaborative environment [15].

Although academic literature has proposed various metrics and models to assess productivity in software development, the interest shown by members of both communities in the practices adopted by other organizations suggests a need for more effective strategies to bridge the gap between academic knowledge and industry application.

5.2 RQ2: How do productivity challenges affect software practitioners?

In the SE forum, practitioners often report how productivity challenges affect them emotionally. Many express **frustrations** when performing tasks perceived as unproductive, such as configuring environments or running tests, revealing a narrow view of productivity focused solely on coding. This perception overlooks the value of essential activities like planning and documentation.

Reports also frequently mention difficulties in maintaining consistent performance, referencing exhaustion and a sense of inefficacy. The expectation of continuous productivity leads to job dissatisfaction. These narratives expose human vulnerabilities, making the Software Engineering forum a technical space and an environment for emotional support and shared experiences. As P7 reflects: *"Sometimes frustrating, stressful & overwhelming. In an ideal world it wouldn't be this way but it appears to be very common. When you are in a setup where it can't be avoided, how do you make the best of it?"*

In contrast, the PM forum discusses impacts more objectively and is results-driven. Only two impact types emerged: **mental health** and **performance issues**, with emphasis on stagnation (linked to a sense of lack of progress) and concern about uneven team performance. We presume that the scarcity of emotional accounts may reflect a professional culture where expressing vulnerability is seen as less appropriate.

5.3 RQ3: How do productivity challenges affect software practitioners' work?

Both forums' most commonly reported work-related impact was **performance reduction**. This terminology was adopted due to a central finding of the analysis: the lack of consensus among professionals on what truly defines "productivity." Participants often referred to "productivity loss" when describing efficiency, effectiveness, or profitability.

To address this ambiguity, the concept of *performance* proposed by Brumby et al. [5] was adopted, offering a more integrated and comprehensive view of software development work. In this context, *efficiency* refers to the optimized use of resources such as time, memory, and computational effort; *effectiveness* relates to delivering functionalities as expected by stakeholders; *profitability* involves the balance between development costs and the value generated, and *productivity* relates to the volume of work delivered in a given period. Therefore, *performance* encompasses all these elements, including the ability to sustain results over time.

Figure 5 illustrates the relationship between these concepts, offering a broader understanding of productivity in software engineering. Adopting this perspective allows for a straightforward interpretation of the everyday challenges faced by professionals and supports the development of more effective strategies to address them.

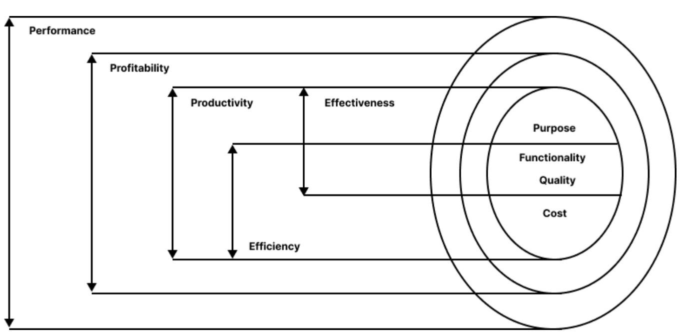

Figure 5: Integrated definition of software productivity. Adapted from Brumby et al. [5].

5.4 Comparison with related work

To broaden the understanding of productivity factors, this study was compared to the work of Wagner and Murphy-Hill [44], which compiles findings from peer-reviewed scientific literature. Unlike their study, this work used GL and data from online forums, environments where professionals feel more comfortable sharing personal experiences. This approach enabled identifying factors and understanding how these challenges emotionally affect professionals and directly impact their work.

The comparison revealed several correspondences between the two studies, even if not always direct, such as in the cases of technology adoption, inadequate work environment, team cohesion, technical complexity, inefficient communication, and context switching. There were also similarities in factors related to psychological well-being and managerial experience.

However, some factors identified in our study were not found in Wagner and Murphy-Hill [44], including *lack of motivation*, *productivity measurement issues*, *management issues*, *lack of autonomy*, *time management*, *lack of skilled co-workers*, *technical debt management*, *working under pressure*, and *productivity variation*. This highlights the value of GL in expanding the perspective on productivity in real-world contexts.

Table 2 presents the equivalence between the factors identified in both studies.

5.5 Conceptual map: linking factors and productivity impacts

To integrate the responses to RQ1, RQ2, and RQ3, we built a conceptual map representing the relationships between the factors influencing productivity, their impacts on professionals, and their effects on tasks. These relationships are organized into two figures: Figure 6 links factors to impacts on professionals, while Figure 7 shows their effects on tasks.

Four-factor categories comprise the figures: technical, organizational, personal, and interpersonal. Arrows indicate the connections identified in the qualitative analysis. Although the frequency of each occurrence is not shown, every relationship represented was reported at least once in the dataset.

The analysis revealed relevant patterns. Technical factors affect tasks directly (e.g., delays or rework), while personal factors such as time management and context switching more strongly impact professionals, leading to exhaustion, stress, and insecurity. Organizational and interpersonal factors showed significant effects at both levels, influencing both individual well-being and the quality of task execution.

This visual mapping supports a clearer understanding of the interactions between factors and their effects. It can help researchers and practitioners identify priority areas for intervention. It also serves as a foundation for future research and practical strategies to improve productivity in software development environments.

For example, a software team leader could use the conceptual map to identify how recurring context switching (a personal factor) reported by developers may lead to cognitive overload and loss of focus. By recognizing this pattern, the leader might restructure

Table 2: Comparison between factors identified in this study and those reported by Wagner and Murphy-Hill [44]

Factors identified in this study	Factors identified by Wagner and Murphy-Hill [44]
Adoption of new standards/technologies	**Platform experience**: "The familiarity with the hardware and software platforms"
	Application domain experience: "The familiarity with the application domain"
Improper workspace	**Proper workplace**: "The suitability of the workplace to do creative work"
Team cohesion	**Team cohesion**: "The cooperativeness of the stakeholders"
Product complexity	**Product complexity**: "The complexity of the function and structure of the software"
Ineffective communication	**Communication**: "The degree and efficiency with which information flows in the team"
	Credibility: "Open communication and competent organization"
Deadline management	**Schedule**: "The appropriateness of the schedule for the development task"
Context switching	**Time fragmentation**: "The amount of necessary 'context switches' of a person"
Psychological distress	**Psychological safety**: "The atmosphere is safe for risk-taking"
Insufficient role experience	**Manager application domain experience**: "The familiarity of the manager with the application"
Requirements volatility	**Requirements stability**: "The number of requirements changes"

work routines to minimize interruptions, reduce parallel task assignments, or schedule more focused work sessions—ultimately aiming to improve both developer well-being and delivery outcomes.

5.6 Implications of the study

This study provides meaningful contributions for researchers, practitioners, and educators by analyzing the factors affecting productivity and their impacts on professionals, their tasks, and the strategies discussed by the communities.

For **researchers**, the findings highlight the importance of exploring underrepresented factors in the academic literature, especially those related to well-being and emotional dynamics in the workplace. The study provides evidence of GL as a valuable data source for uncovering additional dimensions and identifying new research opportunities in industrial and complex settings. We offer a publicly available dataset and a set of visual artifacts that synthesize key insights and can serve as foundations for comparative studies or tool development.

For **professionals**, the study offers insights to support more effective management practices by emphasizing how communication, team cohesion, and time organization influence productivity and well-being. For instance, a manager can use our conceptual map to identify recurring challenges developers face, identify a pattern, and seek focused improvement for developers' well-being and delivery outcomes.

For **educators**, the results provide a foundation to better prepare students for real-world challenges, promoting the development of skills such as time management, teamwork, and resilience under organizational pressure.

Overall, by addressing the factors and their practical effects, this study advances knowledge, professional practice, and educational preparation in software engineering.

6 Limitations

This section exposes some of the limitations of this study and the actions we took to mitigate the issues through our research design. We adopted the Total Quality Framework (TQF) approach for qualitative research [33]. TQF derives from experiences in qualitative research and is more appropriate to this work than the typical internal/external/construct frame often applied in statistical studies [23]. Below, we present some limitations in terms of the subdomains' credibility (concerning data collection), analyzability (data analysis), transparency (reporting), and usefulness (results).

Regarding **credibility**, we sought to reinforce the completeness and accuracy of data collection. The scope of the study was limited to a representative sample of texts collected from the Software Engineering (SE) and Project Management (PM) forums. As described in Section 3.2, we applied a temporal filter, excluding discussions created before January 1, 2012, in order to focus on more current practices and challenges in software development. However, we acknowledge that this temporal cutoff may influence the results, as older discussions could reflect different perceptions, organizational contexts, and technological trends. Analyzing other platforms could also yield different answers to the research questions. Therefore, this study does not aim to be exhaustive, and the findings are specific to the sample of texts obtained from these communities and within the selected time frame.

To minimize the inclusion of participants without professional experience, we selected only discussions in which the authors explicitly or implicitly demonstrated real-world experience in software development. Indicators of professional background included references to corporate environments, organizational practices, professional tools, team management, and responsibilities typically associated with workplace settings. While it is not possible to fully verify each author's background, these criteria served as filters to enhance the credibility of the analyzed contributions.

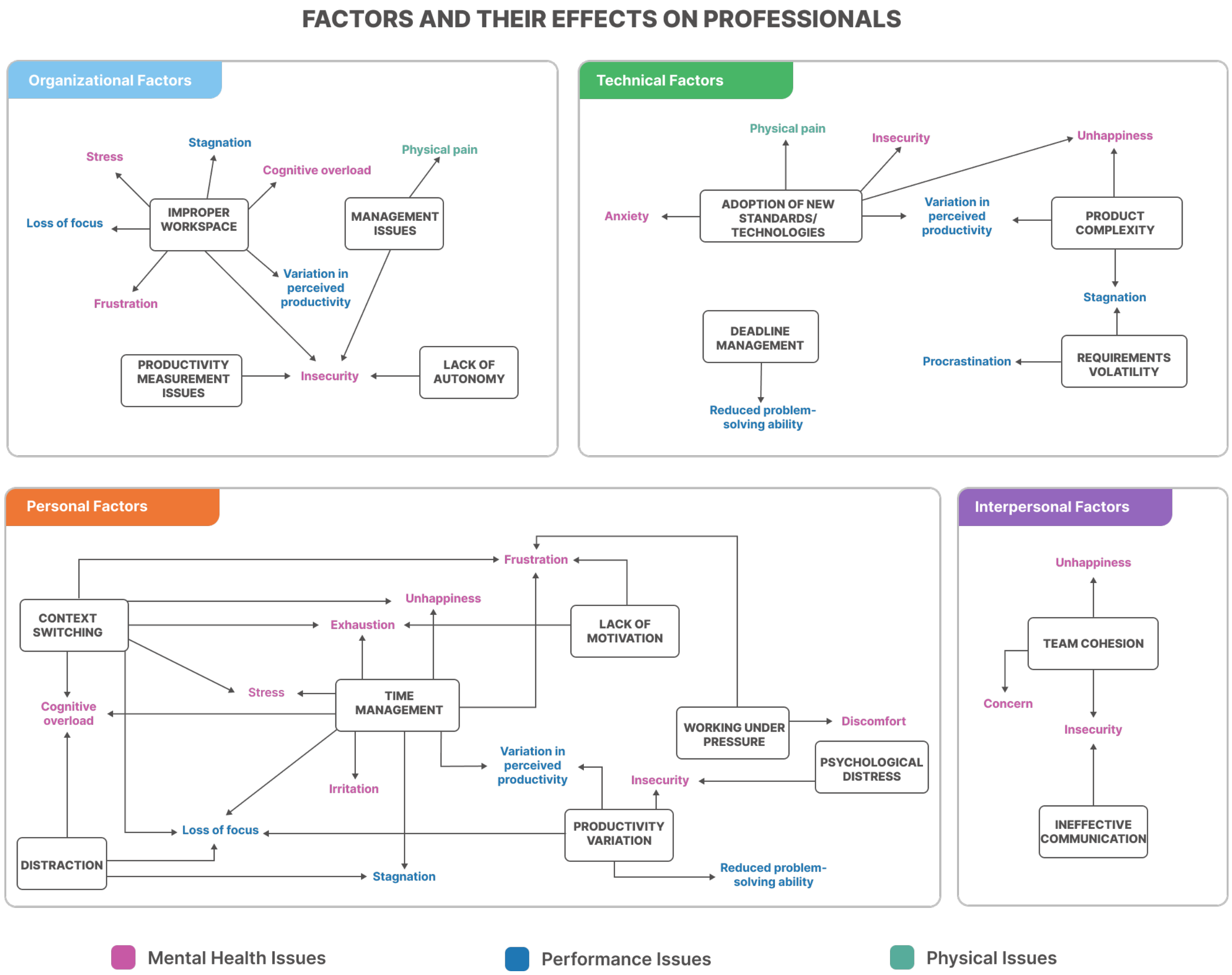

Figure 6: Map of factors and their impacts on professionals.

The **analyzability** of the analysis and interpretations depends on human judgment, which may introduce research bias. To mitigate this limitation, we performed a calibration step in which four independent researchers reviewed the generated codes. We discussed the analysis with experienced researchers until saturation. However, we still recognize that the codes are open to interpretation. Hence, the dataset is available as supplementary material [39], allowing other researchers to examine and validate the analysis.

Concerning the **transparency** of the study, the curated dataset used in the analyses is available in our supplementary material [40] at Zenodo, allowing other researchers to access and analyze the data to verify and validate the conclusions presented. Hosting the data in this repository allows us to validate the present study and foster future research that may expand or question its findings. The proposed methodology can be applied to other communities, such as Reddit or GitHub, with appropriate adjustments. The stages of data collection, filtering, and analysis are replicable, provided that selection criteria and language style are adapted to the specific platform. Thematic analysis and the use of conceptual maps remain applicable, enabling the exploration of how professionals articulate their challenges in more informal or collaborative environments. This may reveal new patterns and enhance the findings with broader and more diverse perspectives.

About the **usefulness** of the findings, this study may be valuable for both researchers and practitioners, as it deepens the understanding of the factors that affect productivity in different contexts. This study addresses a data source under-explored in studies on productivity in software engineering: grey literature. This approach allowed us to identify additional dimensions of productivity and specific challenges practitioners face, as discussed in Section 4. In addition, the work generated practical implications for different audiences, as presented in section 5.6.

7 Conclusion

In this study, we investigate how software development professionals perceive and experience productivity-related challenges in

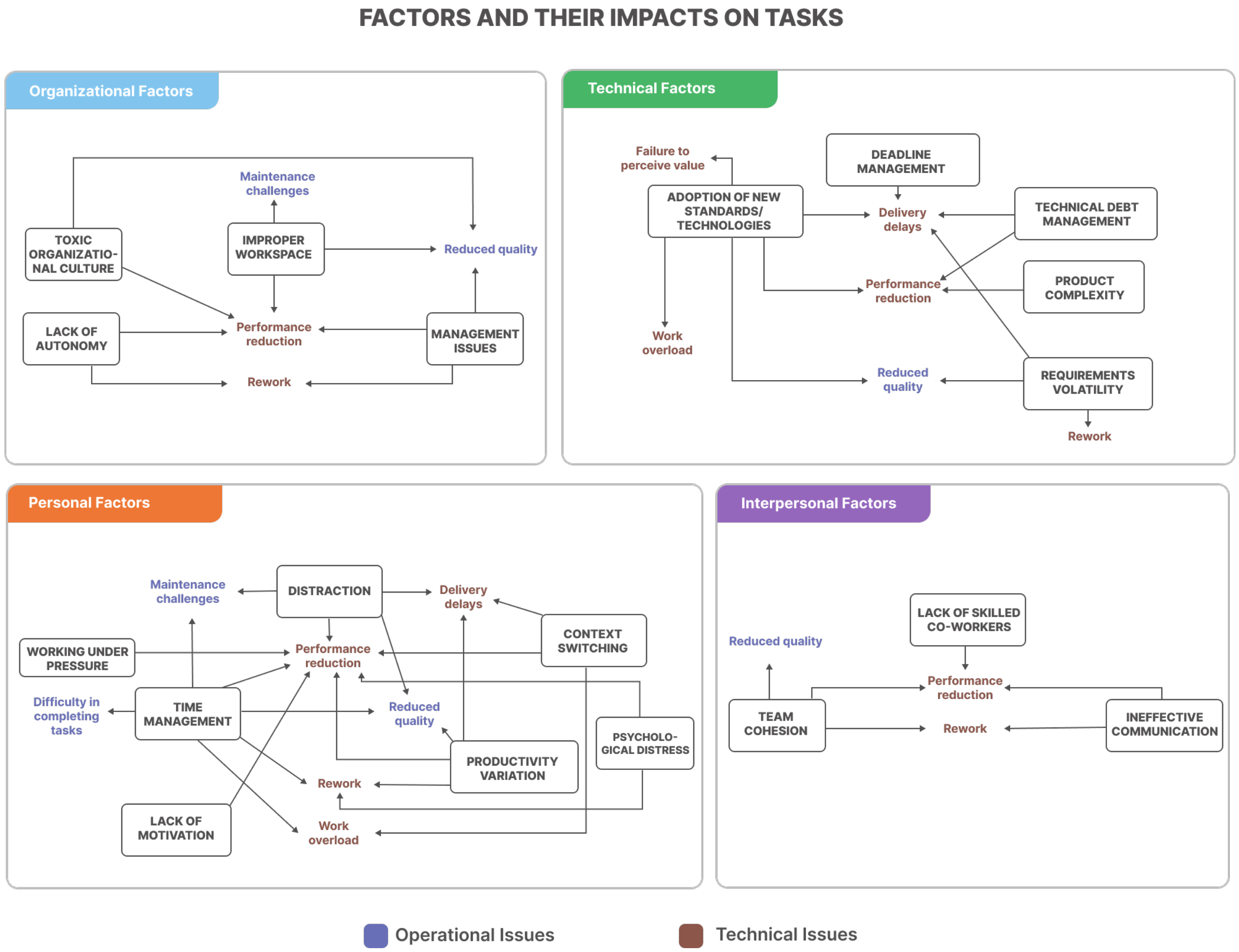

Figure 7: Map of factors and their impacts on work tasks.

their daily work. By analyzing online discussion forums, we aim to understand their concrete impacts on professionals and their tasks.

Adopting a human-centered approach, our study highlights that productivity challenges go beyond technical or organizational issues, encompassing emotional, cognitive, and social dimensions. We grouped the identified factors into technical, organizational, personal, and interpersonal categories and analyzed individual and collective performance. This perspective allowed for a broader understanding of the complexity of productivity in SE.

Our analysis also reveale notable differences between professionals' perceptions of technical roles and management positions. While the former tend to be more open about personal and emotional difficulties, the latter usually approach productivity more objectively and are results-oriented. These distinctions underscore the need for tailored strategies that reflect the realities and responsibilities of each professional profile.

The conceptual map developed throughout the study visually synthesized the relationships between factors and their impacts, contributing to a more integrated understanding of the challenges involved and helping to identify priority areas for action.

As future work, we identified several strategies shared by community members during data collection, which are currently being analyzed to understand how the professionals propose solutions to productivity challenges in software development. In addition, we plan to expand the study's scope to other sources of grey literature, including communities such as Reddit and GitHub Discussions, and to complement the analysis with semi-structured interviews. These initiatives aim to enhance the validity and generalizability of our findings, providing a broader understanding of productivity in SE.

ARTIFACT AVAILABILITY

All data supporting this study are openly available through our supplementary material [40], ensuring transparency, replicability, and accessibility.

ACKNOWLEDGMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior Brasil (CAPES) - Finance Code 001, and by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

REFERENCES

[1] Miltiadis Allamanis and Charles Sutton. 2013. Why, when, and what: Analyzing Stack Overflow questions by topic, type, and code. In *2013 10th Working Conference on Mining Software Repositories (MSR)*. IEEE, San Francisco, CA, USA, 53–56. https://doi.org/10.1109/msr.2013.6624004

[2] Muhammad Asaduzzaman, Ahmed Shah Mashiyat, Chanchal K Roy, and Kevin A Schneider. 2013. Answering questions about unanswered questions of Stack Overflow. In *2013 10th Working Conference on Mining Software Repositories (MSR)*. IEEE, IEEE, San Francisco, CA, USA, 97–100.

[3] Anton Barua, Stephen W. Thomas, and Ahmed E. Hassan. 2014. What Are Developers Talking about? An Analysis of Topics and Trends in Stack Overflow. *Empirical Softw. Engg.* 19, 3 (jun 2014), 619–654. https://doi.org/10.1007/s10664-012-9231-y

[4] B.W. Boehm. 1981. *Software Engineering Economics*. Prentice-Hall, Michigan. https://books.google.com.br/books?id=VphQAAAAMAAJ

[5] Duncan P. Brumby, Christian P. Janssen, and Gloria Mark. 2019. *How Do Interruptions Affect Productivity?* Apress, Berkeley, CA, 85–107. https://doi.org/10.1007/978-1-4842-4221-6_9

[6] Edna Dias Canedo and Giovanni Almeida Santos. 2019. Factors Affecting Software Development Productivity: An empirical study. In *Proceedings of the XXXIII Brazilian Symposium on Software Engineering* (Salvador, Brazil) *(SBES '19)*. Association for Computing Machinery, New York, NY, USA, 307–316. https://doi.org/10.1145/3350768.3352491

[7] 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. https://doi.org/10.1145/3613372.3613407

[8] 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. https://doi.org/10.1109/MS.2024.3377897

[9] Lidiany Cerqueira, Lourene Nunes, Renan Guerra, Viviane Malheiros, Sávio Freire, Glauco Carneiro, Julio Cesar Leite, Rodrigo Spínola, José Amancio Macedo Santos, and Manoel Mendonça. 2025. Assessing Software Practitioners' Work Engagement and Job Satisfaction in a Large Software Company—What We Have Learned. *SN Computer Science* 6, 3 (2025), 273.

[10] Victor A. Clincy. 2003. Software Development Productivity and Cycle Time Reduction. *J. Comput. Sci. Coll.* 19, 2 (dec 2003), 278–287.

[11] Daniela S. Cruzes and Tore Dyba. 2011. Recommended Steps for Thematic Synthesis in Software Engineering. In *2011 International Symposium on Empirical Software Engineering and Measurement*. IEEE, Banff, AB, Canada, 275–284. https://doi.org/10.1109/ESEM.2011.36

[12] B. Curtis. 1991. Techies as Nontechnological Factors in Software Engineering?. In *Proceedings 13th ICSE*. IEEE Computer Society, Los Alamitos, CA, USA, 147–148.

[13] Laura Dabbish, Colleen Stuart, Jason Tsay, and Jim Herbsleb. 2012. Social coding in GitHub: transparency and collaboration in an open software repository. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work* (Seattle, Washington, USA) *(CSCW '12)*. Association for Computing Machinery, New York, NY, USA, 1277–1286. https://doi.org/10.1145/2145204.2145396

[14] Paul Dourish and Victoria Bellotti. 1992. Awareness and coordination in shared workspaces. In *Proceedings of the 1992 ACM Conference on Computer-Supported Cooperative Work* (Toronto, Ontario, Canada) *(CSCW '92)*. Association for Computing Machinery, New York, NY, USA, 107–114. https://doi.org/10.1145/143457.143468

[15] Nicole Forsgren, Margaret-Anne Storey, Chandra Maddila, Thomas Zimmermann, Brian Houck, and Jenna Butler. 2021. The SPACE of Developer Productivity: There's More to It than You Think. *Queue* 19, 1 (feb 2021), 20–48. https://doi.org/10.1145/3454122.3454124

[16] Sávio Freire, Felipe Gomes, Larissa Barbosa, Thiago Souto Mendes, Galdir Reges, Rita S. P. Maciel, Manoel Mendonça, and Rodrigo Spínola. 2023. Requirements Engineering Issues Experienced by Software Practitioners: A Study on Stack Exchange. In *Requirements Engineering: Foundation for Software Quality: 29th International Working Conference, REFSQ 2023, Barcelona, Spain, April 17–20, 2023, Proceedings* (Barcelona, Spain). Springer-Verlag, Berlin, Heidelberg, 3–20. https://doi.org/10.1007/978-3-031-29786-1_1

[17] Norman Gaither and Greg Frazier. 2002. *Administração da produção de operações* (8 ed.). Pioneira, São Paulo.

[18] Vahid Garousi, Michael Felderer, and Mika V. Mäntylä. 2016. The Need for Multivocal Literature Reviews in Software Engineering: Complementing Systematic Literature Reviews with Grey Literature. In *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering* (Limerick, Ireland) *(EASE '16)*. Association for Computing Machinery, New York, NY, USA, Article 26, 6 pages. https://doi.org/10.1145/2915970.2916008

[19] 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 (2023), 12–1.

[20] 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* (Pittsburgh, Pennsylvania) *(TechDebt '22)*. Association for Computing Machinery, New York, NY, USA, 31–40. https://doi.org/10.1145/3524843.3528095

[21] Daniel Graziotin and Fabian Fagerholm. 2019. *Happiness and the Productivity of Software Engineers*. Apress, Berkeley, CA, 109–124. https://doi.org/10.1007/978-1-4842-4221-6_10

[22] Fernando Kamei, Gustavo Pinto, Igor Wiese, Márcio Ribeiro, and Sérgio Soares. 2021. What Evidence We Would Miss If We Do Not Use Grey Literature?. In *Proceedings of the 15th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)* (Bari, Italy) *(ESEM '21)*. Association for Computing Machinery, New York, NY, USA, Article 24, 11 pages. https://doi.org/10.1145/3475716.3475777

[23] Per Lenberg, Robert Feldt, Lucas Gren, Lars Göran Wallgren Tengberg, Inga Tidefors, and Daniel Graziotin. 2024. Qualitative software engineering research: Reflections and guidelines. *Journal of Software: Evolution and Process* 36, 6 (2024), e2607.

[24] André N. Meyer, Thomas Fritz, Gail C. Murphy, and Thomas Zimmermann. 2014. Software developers' perceptions of productivity. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering* (Hong Kong, China) *(FSE 2014)*. Association for Computing Machinery, New York, NY, USA, 19–29. https://doi.org/10.1145/2635868.2635892

[25] João Eduardo Montandon, Cristiano Politowski, Luciana Lourdes Silva, Marco Tulio Valente, Fabio Petrillo, and Yann-Gaël Guéhéneuc. 2021. What skills do IT companies look for in new developers? A study with Stack Overflow jobs. *Information and Software Technology* 129 (2021), 106429. https://doi.org/10.1016/j.infsof.2020.106429

[26] Emerson Murphy-Hill and Stefan Wagner. 2019. *Software Productivity Through the Lens of Knowledge Work*. Apress, Berkeley, CA, 57–65. https://doi.org/10.1007/978-1-4842-4221-6_7

[27] Edson Oliveira, Tayana Conte, Marco Cristo, and Emilia Mendes. 2016. Software Project Managers' Perceptions of Productivity Factors: Findings from a Qualitative Study. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* (Ciudad Real, Spain) *(ESEM '16)*. Association for Computing Machinery, New York, NY, USA, Article 15, 6 pages. https://doi.org/10.1145/2961111.2962626

[28] Edson Oliveira, Tayana Conte, Marco Cristo, and Natasha Valentim. 2018. Influence Factors in Software Productivity - A Tertiary Literature Review. In *International Conferences on Software Engineering and Knowledge Engineering*. KSI Research Inc. and Knowledge Systems Institute Graduate School, San Francisco, California, 1795–1810. https://doi.org/10.18293/seke2018-149

[29] 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. https://doi.org/10.1145/3701625.3701670

[30] Kai Petersen. 2011. Measuring and predicting software productivity: A systematic map and review. *Information and Software Technology* 53, 4 (2011), 317–343. https://doi.org/10.1016/j.infsof.2010.12.001 Special section: Software Engineering track of the 24th Annual Symposium on Applied Computing.

[31] Yuri W. Ramírez and David A. Nembhard. 2004. Measuring knowledge worker productivity. *Journal of Intellectual Capital* 5, 4 (Dec. 2004), 602–628. https://doi.org/10.1108/14691930410567040

[32] Marcelo B Ribeiro, Ricardo M Czekster, and Thais Webber. 2006. Improving Productivity of Local Software Development Teams in a Global Software Development Environment. *Engineering* 1-2 (2006), 1–2.

[33] M.R. Roller and P.J. Lavrakas. 2015. *Applied Qualitative Research Design: A Total Quality Framework Approach*. Guilford Publications, New York.

[34] Caitlin Sadowski and Thomas Zimmermann (Eds.). 2019. *Rethinking Productivity in Software Engineering*. Apress, New York. https://doi.org/10.1007/978-1-4842-4221-6

[35] Suzana Candido Barros Sampaio, Emanuella Aleixo Barros, Gibeon Soares de Aquino, Mauro Jose Carlos e Silva, and Silvio Romero de Lemos Meira. 2010. A Review of Productivity Factors and Strategies on Software Development. In *2010 Fifth International Conference on Software Engineering Advances*. IEEE, Nice, France, 196–204. https://doi.org/10.1109/icsea.2010.37

[36] Beatriz Santana, Sávio Freire, José Amancio Macedo Santos, and Manoel Mendonça. 2024. Psychological Safety in the Software Work Environment. *IEEE Softw.* 41, 4 (July 2024), 86–94. https://doi.org/10.1109/MS.2024.3386532

[37] 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 (Campo Grande, Brazil) *(SBES '23)*. Association for Computing Machinery, New York, NY, USA, 503–513. https://doi.org/10.1145/3613372.3613411

[38] W. Scacchi. 1989. *Understanding Software Productivity: A Comparative Empirical Review*. Technical Report. Institute for Software Research, University of California.

[39] Miguel-Angel Sicilia, Elena García-Barriocanal, and Salvador Sánchez-Alonso. 2017. Community Curation in Open Dataset Repositories: Insights from Zenodo. *Procedia Computer Science* 106 (2017), 54–60. https://doi.org/10.1016/j.procs.2017.03.009

[40] Julio César Andrade Silva, Lidiany Cerqueira Santos, Beatriz Santana, Glauco Carneiro, Sávio Freire, Manoel Mendonça, and José Amancio Macedo Santos. 2025. Analysis Spreadsheet: Software Engineering and Project Management Forums on Stack Exchange. https://doi.org/10.5281/zenodo.15285102

[41] Amjed Tahir, Jens Dietrich, Steve Counsell, Sherlock Licorish, and Aiko Yamashita. 2020. A large scale study on how developers discuss code smells and anti-pattern in Stack Exchange sites. *Information and Software Technology* 125 (2020), 106333. https://doi.org/10.1016/j.infsof.2020.106333

[42] Adam Trendowicz and Jürgen Münch. 2009. Factors Influencing Software Development Productivity—State-of-the-Art and Industrial Experiences. In *Advances in Computers, Volume 77*. Advances in Computers, Vol. 77. Elsevier, USA, 185–241. https://doi.org/10.1016/S0065-2458(09)01206-6

[43] Stefan Wagner and Florian Deissenboeck. 2019. *Defining Productivity in Software Engineering*. Apress, Berkeley, CA, 29–38. https://doi.org/10.1007/978-1-4842-4221-6_4

[44] Stefan Wagner and Emerson Murphy-Hill. 2019. Factors That Influence Productivity: A Checklist. In *Rethinking Productivity in Software Engineering*, Caitlin Sadowski and Thomas Zimmermann (Eds.). Apress, Berkeley, CA, 69–84. https://doi.org/10.1007/978-1-4842-4221-6_8