



Universidade Estadual de Feira de Santana  
Programa de Pós-Graduação em Ciência da Computação

Investigando a percepção de praticantes da  
Engenharia de Software sobre  
produtividade: uma análise com base na  
literatura cinza

Julio César Andrade Silva

Feira de Santana

2025



Universidade Estadual de Feira de Santana  
Programa de Pós-Graduação em Ciência da Computação

Julio César Andrade Silva

**Investigando a percepção de praticantes da Engenharia  
de Software sobre produtividade: uma análise com base  
na literatura cinza**

Dissertação apresentada à Universidade  
Estadual de Feira de Santana como parte  
dos requisitos para a obtenção do título de  
Mestre em Ciência da Computação.

Orientador: José Amancio Macedo Santos

Feira de Santana

2025

## **Ficha Catalográfica – Biblioteca Central Julieta Carteado**

S58i	<p>Silva, Júlio César Andrade Investigando a percepção de praticantes da Engenharia de Software sobre produtividade: uma análise com base na literatura cinza./ Júlio César Andrade Silva Silva. – 2024. 112 f.: il.</p> <p>Orientador: José Amancio Macedo Santos Dissertação (mestrado) – Universidade Estadual de Feira de Santana, Programa de Pós-Graduação em Ciência da Computação, 2024.</p> <p>1.Engenharia de software – Produtividade. 2.Literatura Cinza. 3.Fatores não técnicos. I.Santos, José Amancio Macedo, orient. II.Universidade Estadual de Feira de Santana. III.Título.</p>
CDU : 004.41	

Maria de Fátima de Jesus Moreira – Bibliotecária – CRB5/1120

Júlio César Andrade Silva

## **Investigando a percepção de praticantes da Engenharia de Software sobre produtividade: uma análise com base na literatura cinza**

Dissertação apresentada à Universidade Estadual de Feira de Santana como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Feira de Santana, 16 de dezembro de 2024

### **BANCA EXAMINADORA**

Documento assinado digitalmente  
 JOAO BATISTA DA ROCHA JUNIOR  
Data: 08/01/2025 14:22:14-0300  
Verifique em <https://validar.iti.gov.br>

---

João Batista da Rocha-Junior (Coorientador(a))  
Universidade Estadual de Feira de Santana

Documento assinado digitalmente  
 MANOEL GOMES DE MENDONCA NETO  
Data: 17/12/2024 15:05:38-0300  
Verifique em <https://validar.iti.gov.br>

---

Manoel Gomes de Mendonça Neto  
Universidade Federal da Bahia

Documento assinado digitalmente  
 GLAUCO DE FIGUEIREDO CARNEIRO  
Data: 28/12/2024 11:48:47-0300  
Verifique em <https://validar.iti.gov.br>

---

Glauco de Figueiredo Carneiro  
Universidade Federal de Sergipe

# Abstract

The increasing competitiveness in the software industry has pressured teams to improve performance and reduce production costs while adhering to increasingly tight deadlines. In this context, productivity emerges as a critical aspect for organizations seeking to understand and optimize this factor. This study investigated productivity challenges from the perspective of two communities on the Stack Exchange platform: Software Engineering and Product Manager. It explored the factors that generate these challenges, their impacts on professionals and their tasks, as well as the strategies suggested to mitigate these challenges. Through a gray literature review and qualitative analysis of discussions, 21 factors affecting productivity were identified, highlighting time management, the adoption of new standards and technologies, difficulties in measuring productivity, and context switching. These challenges reflect not only technical issues but also human aspects, such as demotivation and psychological pressure, showing that productivity cannot be understood in isolation. Furthermore, the study identified 17 direct impacts on professionals, including stress and perceived productivity variation. The strategies used by the communities included practical approaches, such as task reorganization, and reflective methods, such as perspective adaptation, demonstrating the flexibility of professionals in balancing technical and human aspects. Understanding these factors and strategies is essential to inform future research efforts and enhance the performance of software development teams.

**Keywords:** Productivity, Gray Literature, Software engineering, Non-technical factors.

# Resumo

A crescente competitividade no setor de software tem pressionado as equipes a melhorar a performance e reduzir custos de produção, enquanto enfrentam prazos de entrega cada vez mais apertados. Nesse contexto, a produtividade emerge como um aspecto essencial para as organizações, que buscam formas de compreender e otimizar esse fator. Este estudo investigou os desafios de produtividade sob a perspectiva de duas comunidades da plataforma *Stack Exchange: Software Engineering e Product Manager*, explorando os fatores que os geram, seus impactos nos profissionais e suas tarefas, além das estratégias sugeridas para mitigar esses desafios. Por meio de uma revisão de literatura cinza e análise qualitativa das discussões, foram identificados 21 fatores que afetam a produtividade, com destaque para gestão do tempo, adoção de novos padrões e tecnologias, dificuldade em medir produtividade e troca de contexto. Esses desafios refletem não apenas problemas técnicos, mas também questões humanas, como desmotivação e pressão psicológica, evidenciando que a produtividade não pode ser compreendida isoladamente. Além disso, o estudo revelou 17 impactos diretos nos profissionais, como estresse e variação na produtividade percebida. As estratégias utilizadas pelas comunidades incluem abordagens práticas, como reorganização de tarefas, e reflexivas, como adaptação de perspectiva, indicando a flexibilidade dos profissionais para equilibrar aspectos técnicos e humanos. Compreender esses fatores e estratégias é essencial para informar futuros esforços de pesquisa e aprimorar o desempenho de equipes no desenvolvimento de software.

**Palavras-chave:** Produtividade, Literatura Cinza, Engenharia de software, Fatores não técnicos.

# Prefácio

Esta dissertação de mestrado foi submetida à Universidade Estadual de Feira de Santana (UEFS) como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

A dissertação foi desenvolvida no Programa de Pós-Graduação em Ciência da Computação (PGCC), tendo como orientador o Prof. Dr. **José Amancio Macedo Santos**

Esta pesquisa foi financiada pela FAPESB (Fundação de Amparo à Pesquisa do Estado da Bahia).

# Agradecimentos

Expresso minha mais profunda gratidão à minha mãe, pelo apoio incondicional e pela força que sempre me transmitiu ao longo da minha jornada acadêmica. Obrigado por acreditar em mim, por suas palavras de encorajamento e por me inspirar a lutar pelos meus sonhos.

Agradeço ao meu orientador, José Amâncio, pela orientação cuidadosa, paciência e por sempre estar presente durante o processo de pesquisa. Suas cobranças e críticas construtivas foram essenciais para que eu me superasse e aprimorasse este trabalho. Sou profundamente grato pelo suporte contínuo, mesmo em momentos em que talvez eu não o merecesse.

Meus sinceros agradecimentos à Lidiany, cuja ajuda inestimável foi fundamental em todas as etapas desta pesquisa. Seu apoio constante, suas ideias e sugestões vieram nos momentos em que mais precisei e foram essenciais para minha motivação e progresso.

Agradeço também a Sávio, Beatriz e Lidivânia, que atuaram como revisores e contribuíram significativamente para o sucesso desta pesquisa. Suas análises detalhadas e valiosas observações foram indispensáveis para o desenvolvimento deste trabalho.

Aos amigos que estiveram ao meu lado durante essa jornada, meu muito obrigado. Em especial, agradeço a Nayara e Jussara, cujos conselhos, carinho, apoio e momentos de leveza foram fundamentais para que eu mantivesse o equilíbrio e a felicidade ao longo desses anos.

Por fim, e acima de tudo, agradeço a Deus por ter me abençoado com a oportunidade de perseguir meus sonhos, por me dar forças para superar os desafios e por estar sempre presente em cada etapa do caminho. Sem Ele, nada disso seria possível. Obrigado por tudo.

# Sumário

<b>Abstract</b>	<b>i</b>
<b>Resumo</b>	<b>ii</b>
<b>Prefácio</b>	<b>iii</b>
<b>Agradecimentos</b>	<b>iv</b>
<b>Alinhamento com a Linha de Pesquisa</b>	<b>viii</b>
<b>Lista de Tabelas</b>	<b>ix</b>
<b>Lista de Figuras</b>	<b>xi</b>
<b>Lista de Abreviações</b>	<b>xii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Objetivos . . . . .	3
1.1.1 Objetivo geral . . . . .	3
1.1.2 Objetivos Específicos . . . . .	4
1.2 Metodologia . . . . .	4
1.3 Contribuições . . . . .	4
1.4 Organização do Trabalho . . . . .	5
<b>2 Revisão Bibliográfica</b>	<b>6</b>
2.1 Produtividade . . . . .	6
2.1.1 Produtividade no desenvolvimento de software . . . . .	7
2.1.2 Fatores que influenciam a produtividade no desenvolvimento de software . . . . .	8
2.2 Literatura cinza . . . . .	10
2.2.1 Importância de considerar a literatura cinza em estudos sobre engenharia de software . . . . .	12
2.2.2 Stack Exchange . . . . .	13
2.3 Trabalhos Relacionados . . . . .	13
2.3.1 Produtividade e a Engenharia de Software . . . . .	13

2.3.2 Literatura cinza e a Engenharia de Software . . . . .	15
<b>3 Metodologia</b>	<b>17</b>
3.1 Planejamento . . . . .	18
3.1.1 Revisão bibliográfica . . . . .	18
3.1.2 Escopo da pesquisa . . . . .	18
3.1.3 Questões de pesquisa . . . . .	19
3.1.4 Seleção da fonte . . . . .	20
3.2 Coleta de dados . . . . .	22
3.2.1 <i>String</i> de busca . . . . .	22
3.2.2 Critérios de inclusão e exclusão . . . . .	23
3.2.3 Tratamento dos dados . . . . .	24
3.3 Análise dos dados . . . . .	25
3.3.1 Seleção das discussões . . . . .	26
3.3.2 Análise das discussões . . . . .	26
3.3.3 Codificação e categorização . . . . .	27
3.4 Execução . . . . .	27
3.4.1 Execução da etapa de seleção dos dados . . . . .	29
3.4.2 Execução da análise dos dados . . . . .	31
3.4.3 Divergências nas análises e consenso . . . . .	33
3.4.4 Codificação e classificação . . . . .	33
<b>4 Resultados</b>	<b>37</b>
4.1 Resultado da análise do fórum Software Engineering . . . . .	37
4.1.1 RQ1: Quais são os desafios de produtividade enfrentados pelos desenvolvedores de software? . . . . .	37
4.1.2 RQ2: Como os desafios de produtividade afetam os desenvolvedores de software? . . . . .	44
4.1.3 RQ3: Como os desafios de produtividade afetam o trabalho realizado pelos desenvolvedores de software? . . . . .	49
4.1.4 RQ4: Que estratégias e atividades para melhorar a produtividade são discutidas? . . . . .	53
4.2 Fórum da comunidade Project Management . . . . .	58
4.2.1 RQ1: Quais são os desafios de produtividade enfrentados pelos desenvolvedores de software? . . . . .	58
4.2.2 RQ2: Como os desafios de produtividade afetam os desenvolvedores de software? . . . . .	60
4.2.3 RQ3: Como os desafios de produtividade afetam o trabalho realizado pelos desenvolvedores de software? . . . . .	61
4.2.4 RQ4: Quais estratégias e atividades para melhorar a produtividade são discutidas? . . . . .	62
4.3 Comparação dos fóruns Software Engineering e Project Management	63
4.3.1 RQ1: Quais são os desafios de produtividade enfrentados pelos desenvolvedores de software? . . . . .	63

4.3.2	RQ2: Como os desafios de produtividade afetam os desenvolvedores de software? . . . . .	64
4.3.3	RQ3: Como os desafios de produtividade afetam o trabalho realizado pelos desenvolvedores de software? . . . . .	65
4.3.4	RQ4: Quais estratégias os desenvolvedores utilizam para superar os desafios de produtividade? . . . . .	65
<b>5</b>	<b>Discussão</b>	<b>66</b>
5.1	Respondendo as questões de pesquisa . . . . .	66
5.1.1	RQ1: Quais são os desafios de produtividade enfrentados pelos desenvolvedores de software? . . . . .	67
5.1.2	RQ2: Como os desafios de produtividade impactam os desenvolvedores de software? . . . . .	72
5.1.3	RQ3: Como os desafios de produtividade afetam o trabalho realizado pelos desenvolvedores de software? . . . . .	74
5.1.4	RQ4: Quais estratégias e atividades para melhorar a produtividade são discutidas? . . . . .	76
5.2	Mapa conceitual: relacionando fatores e impactos sobre a produtividade	80
5.2.1	Estrutura das figuras . . . . .	80
5.2.2	Considerações sobre a representação . . . . .	81
5.2.3	Observações gerais . . . . .	82
5.3	Comparação com trabalho relacionado	83
5.4	Limitações do estudo . . . . .	86
5.5	Implicações do estudo . . . . .	87
<b>6</b>	<b>Conclusão</b>	<b>88</b>
<b>Referências</b>		<b>91</b>

# **Alinhamento com a Linha de Pesquisa**

## **Linha de Pesquisa: Software e Sistemas Computacionais**

Este estudo está alinhado à linha de pesquisa Software e Sistemas Computacionais, uma vez que busca identificar e analisar os fatores que impactam na produtividade dos profissionais que atuam no desenvolvimento de software, contribuindo para o desenvolvimento de soluções mais eficientes e eficazes para atender às necessidades desse mercado.

# **Lista de Tabelas**

3.1	Dimensão das comunidades investigadas. . . . .	22
4.1	Relação dos fatores que geram os desafios de produtividade: Fórum SE. . . . .	39
4.2	Tabela de fatores e número de ocorrências. . . . .	39
4.3	Impactos dos desafios sobre os profissionais . . . . .	46
4.4	Número de ocorrências dos problemas gerados. . . . .	50
4.5	Estratégias sugeridas pelos membros da comunidade para melhorar a produtividade. . . . .	55
4.6	Relação dos fatores que geram os desafios de produtividade. . . . .	59
4.7	Impactos dos desafios sobre os profissionais. . . . .	60
4.8	Número de ocorrências dos problemas gerados. . . . .	62
4.9	Tipos de estratégias recomendadas pelos membros do fórum Project Management. . . . .	64

# **Lista de Figuras**

2.1	Os tons de cinza da literatura. Adaptado de Adams et al. (2016) . . . . .	10
2.2	Os tons de cinza da literatura para SE. Adaptado de Garousi et al. (2016) . . . . .	11
3.1	Etapas da metodologia . . . . .	17
3.2	Resumo do processo de coleta dos dados. . . . .	25
3.3	Exemplo de discussão com resposta, upvotes e tags. . . . .	28
3.4	Fluxograma resumido do processo de análise das discussões. Adaptado de Gomes et al. (2023). . . . .	29
3.5	Etapas que antecedem a análise das discussões . . . . .	30
3.6	Exemplo de resposta extraída para a questão RQ1. . . . .	32
3.7	Exemplo de resposta extraída para a questão RQ2 . . . . .	32
3.8	Exemplo de resposta extraída para a questão RQ3 . . . . .	33
3.9	Exemplo de resposta extraída para a questão RQ4. . . . .	34
3.10	Resumo do processo de seleção das discussões no fórum SE. . . . .	35
3.11	Resumo do processo de seleção das discussões no fórum PM. . . . .	35
3.12	Exemplo de codificação da questão RQ1. . . . .	36
3.13	Exemplo de codificação da questão RQ4. . . . .	36
4.1	Fatores que geram os desafios de produtividade . . . . .	38
4.2	Impactos dos desafios de produtividade sobre os profissionais membros da comunidade <i>Software Engineering</i> . . . . .	45
4.3	Impactos dos desafios de produtividade sobre as tarefas realizadas pelos membros da comunidade SE . . . . .	50
4.4	Estratégias para minimizar o impacto dos fatores. . . . .	54
4.5	Fatores que geram os desafios de produtividade no fórum PM. . . . .	58
4.6	Impactos dos desafios de produtividade sobre os profissionais membros da comunidade <i>Software Engineering</i> . . . . .	60
4.7	Impactos dos desafios de produtividade sobre as tarefas realizadas pelos membros da comunidade PM . . . . .	61
4.8	Estratégias para minimizar o impacto dos fatores que afetam a produtividade dos profissionais. . . . .	63
5.1	Definição integrada de produtividade de software. Adaptado de Brumby et al. (2019). . . . .	75

5.2	Exemplo de códigos que foram unificados no fórum SE. . . . .	77
5.3	Fatores e seus impactos sobre os profissionais. . . . .	81
5.4	Fatores e seus impactos sobre as tarefas. . . . .	82
5.5	Resumo da comparação entre os estudos. . . . .	85

# **Lista de Abreviações**

<b>Abreviação</b>	<b>Descrição</b>
LC	Literatura Cinza
PM	Project Manager
SE	Software Engineering
ESBE	Engenharia de Software Baseada em Evidências
LOC	Lines of Code
SEDE	Stack Exchange Data Explorer

# Capítulo 1

## Introdução

Na esfera organizacional, é amplamente reconhecido que os recursos humanos são o ativo mais valioso. Isso se deve à sua capacidade de gerar conhecimento e produzir resultados, bem como à responsabilidade pelo planejamento e execução de metas. Sendo assim, a produtividade humana é um fator determinante para a geração de produtos e serviços, destacando-se como um elemento crucial para a obtenção do sucesso empresarial (Tangen, 2005; Wagner e Deissenboeck, 2019).

A globalização da economia tem gerado um aumento na competitividade empresarial em diversos setores, incluindo a indústria de software. Diante deste contexto, as organizações têm enfrentado a necessidade de entregar produtos com níveis de qualidade mais altos e em prazos de entrega cada vez menores. Isso demanda um aumento da performance das equipes de desenvolvimento e uma redução nos custos de produção para manter a competitividade (Murphy-Hill e Wagner, 2019).

Considerando que a Engenharia de Software é uma atividade predominantemente humana, a produtividade dos desenvolvedores é um fator crítico para reduzir os custos de produção do software (Sommerville, 2020). Contudo, para garantir bons resultados, é necessário implementar práticas eficazes embasadas em fatores que influenciam a produtividade dos desenvolvedores (Wagner e Murphy-Hill, 2019). O desafio é que a produtividade no desenvolvimento de software é influenciada por múltiplos fatores, e os gestores muitas vezes não têm consciência de quais afetam suas equipes (Canedo e Santos, 2019).

Com o desenvolvimento de software muitas vezes sendo considerado mais como uma arte do que uma ciência, tem sido desafiador estudar os fatores que afetam a produtividade em projetos de software (Forsgren et al., 2021). A falta de compreensão desse tema tem impactado diretamente a indústria, levando a confusão na estimativa de custos de desenvolvimento e insatisfação dos desenvolvedores. (Graziotin e Fagerholm, 2019).

A academia tem direcionado esforços para o estudo da produtividade no desenvolvimento de software, com o objetivo de fornecer informações que facilitem a tomada

de decisões pelas organizações (Canedo e Santos, 2019; Brumby et al., 2019). Essa investigação se torna ainda mais relevante considerando que a produtividade do desenvolvedor é essencial não apenas para o sucesso dos projetos, mas também para o bem-estar dos profissionais, uma vez que produtividade e satisfação estão intrinsecamente relacionadas (Forsgren et al., 2021).<sup>2</sup>

O tema da produtividade tem sido amplamente estudado, especialmente no contexto da percepção dos desenvolvedores sobre sua própria produtividade. O trabalho de Meyer et al. (2014) destaca que, embora exista uma vasta literatura sobre a medição da produtividade sob uma perspectiva organizacional, há uma lacuna significativa na investigação de como os desenvolvedores de software percebem, avaliam e buscam melhorar sua produtividade. Em seu estudo, os autores realizaram uma pesquisa com 379 desenvolvedores profissionais e um estudo observacional com 11 desenvolvedores, identificando que os profissionais associam dias produtivos à conclusão de tarefas grandes ou numerosas, sem interrupções significativas ou trocas de contexto. No entanto, os dados observacionais mostram que, mesmo enfrentando interrupções frequentes, os participantes ainda se consideravam produtivos. Essas aparentes contradições apontam para a necessidade de ferramentas e práticas que apoiem os desenvolvedores na reflexão e melhoria de sua produtividade.

Uma fonte de dados ainda pouco explorada para a análise de desafios relacionados à produtividade é a literatura cinza. Considerar diferentes fontes de dados é um passo crucial para a evolução de uma área de conhecimento (Garousi et al., 2016). Plataformas de perguntas e respostas, por exemplo, são amplamente utilizadas no cotidiano dos desenvolvedores de software (Ahmed e Bagherzadeh, 2018). Esses ambientes oferecem um espaço rico para a discussão de inúmeras questões, expondo diferentes perspectivas dos profissionais sobre soluções para problemas enfrentados pela comunidade de Engenharia de Software (Allamanis e Sutton, 2013; Barua et al., 2014). Entre as diversas plataformas de fóruns de discussão disponíveis, destaca-se a Stack Exchange, que abriga comunidades dedicadas a temas como engenharia de software, gerenciamento de projetos e testes de software (Asaduzzaman et al., 2013).

A inclusão dessas fontes de conhecimento em estudos científicos pode trazer diversas vantagens para os pesquisadores. Entre elas, destaca-se a possibilidade de analisar um volume maior de dados e, possivelmente, abranger uma variedade de tópicos que não são abordados na literatura científica tradicional. Além disso, as informações dessas fontes são frequentemente atualizadas em tempo real, o que as torna uma alternativa valiosa para a análise de dados em constante evolução (Garousi et al., 2016; Kamei et al., 2021b). Por isso, os fóruns de discussão têm sido utilizados como fonte de informação para diversas pesquisas na área de engenharia de software, que buscam estudar temas emergentes, como *code smells* (Tahir et al., 2020), habilidades interpessoais (Montandon et al., 2020) e dívida técnica (Freire et al., 2020).

Embora esses temas já sejam amplamente abordados na literatura acadêmica, a exploração de fóruns de discussão apresenta um potencial significativo para agregar perspectivas práticas, diretamente oriundas da experiência de profissionais da

área (Allamanis e Sutton, 2013). Esses fóruns oferecem um espaço onde indivíduos podem expressar livremente suas opiniões e experiências, diferenciando-se de métodos tradicionais de pesquisa, como entrevistas ou questionários estruturados. Além disso, a participação de pessoas com diferentes origens e perspectivas amplia o alcance das discussões, proporcionando ao pesquisador uma visão mais diversificada e abrangente do tema em questão. A inclusão dessas fontes de conhecimento é, portanto, essencial para desenvolver uma compreensão mais completa dos desafios enfrentados pela comunidade de engenharia de software e para a criação de soluções mais alinhadas à realidade prática (Bajaj et al., 2014).

Nesse contexto, é crucial enfatizar que o foco principal deste estudo não recai sobre as organizações, mas sim sobre o ser humano — ou seja, o desenvolvedor de software. Para atingir esse objetivo, o estudo adota a literatura cinza como fonte de dados, definida como um conjunto de materiais e pesquisas não publicados por canais tradicionais, como periódicos acadêmicos revisados por pares (Garousi et al., 2016). A utilização de literatura cinza, como fóruns de discussão no Stack Exchange, permite captar perspectivas autênticas diretamente dos desenvolvedores, destacando suas experiências e percepções individuais sobre produtividade. Essa abordagem busca preencher lacunas na literatura acadêmica ao conectar o conhecimento teórico da academia com as realidades práticas vividas pelos profissionais da indústria, oferecendo uma visão mais ampla e centrada no ser humano sobre os fatores que influenciam a produtividade.

Dessa forma, espera-se que este trabalho amplie a compreensão dos fatores que afetam a produtividade dos desenvolvedores e inspire novas pesquisas, ferramentas e métodos que abordem essas questões. Ao priorizar as experiências e percepções dos profissionais, este estudo busca contribuir para o desenvolvimento de estratégias que não apenas otimizem a produtividade, mas também promovam o alinhamento entre academia e prática, beneficiando a comunidade de engenharia de software como um todo.

## 1.1 Objetivos

Nesta seção, são apresentados o objetivo geral e os objetivos específicos deste trabalho.

### 1.1.1 Objetivo geral

O objetivo deste estudo é analisar os desafios de produtividade enfrentados pelos desenvolvedores de software e identificar os fatores que os geram. Além disso, busca compreender como esses desafios afetam esses profissionais e estratégias adotadas para superá-los, a fim de melhorar a produtividade no contexto da engenharia de software.

### 1.1.2 Objetivos Específicos

A seguir, são listados os objetivos específicos deste trabalho:

1. Realizar uma análise sobre os desafios de produtividade enfrentados por desenvolvedores de software, compreendendo quais fatores geram esses desafios, como esses fatores impactam os profissionais e suas tarefas, utilizando dados provenientes da literatura cinza;
2. Disponibilizar o *dataset* utilizado na pesquisa para apoiar outras análises relacionadas ao tema;
3. Comparar os fatores que afetam a produtividade identificados na literatura acadêmica com os encontrados na literatura cinza;
4. Identificar as principais estratégias utilizadas pelos desenvolvedores de software para aumentar sua produtividade e propor recomendações práticas para os profissionais de software com base nas informações coletadas na pesquisa.

## 1.2 Metodologia

A metodologia deste estudo foi inspirada no trabalho de Gomes et al. (2023), que explorou como profissionais discutem a dúvida técnica em fóruns online, utilizando uma abordagem qualitativa para analisar as percepções e práticas relatadas. Seguindo essa linha, este estudo realiza uma revisão de literatura cinza, coletando dados qualitativos de fóruns do Stack Exchange voltados para engenheiros de software e gerentes de projetos. Essa abordagem busca captar diferentes perspectivas e experiências relacionadas aos desafios de produtividade enfrentados por esses profissionais.

Para a análise dos dados coletados, utilizou-se a técnica de síntese temática, conforme descrita por Cruzes e Dyba (2011). Essa técnica permite identificar e organizar temas recorrentes, oferecendo uma visão estruturada dos fatores que influenciam a produtividade, dos impactos sobre os profissionais e as tarefas, e das estratégias utilizadas para mitigar esses desafios.

Espera-se que este estudo contribua para aprofundar a compreensão dos desafios enfrentados pela comunidade de engenharia de software e gestão de projetos, fornecendo *insights* práticos e recomendações que possam melhorar a produtividade no desenvolvimento de *software* e inspirar novas pesquisas no tema.

## 1.3 Contribuições

Esse trabalho tem relevância no sentido de: i) complementar os estudos existentes sobre os fatores que influenciam a produtividade, expandindo a análise ao incorporar dados provenientes da literatura cinza; ii) apoiar a comunidade acadêmica na

identificação e no enfrentamento de desafios recentes que afetam a produtividade dos profissionais de software; iii) analisar os impactos dos fatores de produtividade sobre os profissionais de software, com foco no ser humano e nas tarefas desempenhadas, contribuindo para a compreensão de como as variáveis individuais e contextuais influenciam o desempenho e o bem-estar dos desenvolvedores; iv) iniciar um estudo que busca entender como os desenvolvedores de software estão superando os desafios relacionados à produtividade e seus impactos, focando nas estratégias que adotam para lidar com as dificuldades que enfrentam no cotidiano de trabalho. desenvolvedores da área.

## 1.4 Organização do Trabalho

O capítulo 2 deste trabalho apresenta uma revisão bibliográfica a fim de fornecer a base teórica necessária para a compreensão do estudo. A revisão bibliográfica abrange as principais pesquisas e literaturas relacionadas aos temas estudados.

O capítulo 3 apresenta a metodologia científica adotada para condução do estudo. Foi detalhado o processo de coleta de dados, a escolha da amostra, as técnicas e ferramentas utilizadas na análise dos dados e a forma como foram realizadas as inferências a partir dos resultados.

O capítulo 4 descreve os resultados obtidos após a aplicação da metodologia científica sobre o conjunto de dados coletados.

No capítulo 5, são apresentadas reflexões a partir dos resultados obtidos, com base em literaturas e pesquisas correlatas. Essas reflexões possibilitam uma compreensão mais aprofundada dos resultados e das implicações do estudo para a área de Engenharia de Software.

O sexto e último capítulo apresenta as conclusões obtidas a partir dos resultados encontrados e sugere trabalhos futuros a serem desenvolvidos para a continuidade da pesquisa. As conclusões são baseadas nos resultados obtidos e nas reflexões apresentadas, proporcionando uma síntese dos principais achados deste trabalho.

# Capítulo 2

## Revisão Bibliográfica

O objetivo deste capítulo é apresentar uma revisão bibliográfica abrangente sobre os principais tópicos relacionados ao trabalho proposto. Primeiramente, serão discutidos conceitos gerais sobre produtividade, seguidos por uma análise específica da produtividade no contexto do desenvolvimento de software e dos fatores que influenciam nesse processo. Na sequência, será destacada a importância de considerar a literatura cinza em estudos de engenharia de software, com ênfase na análise de plataformas de discussão como o Stack Exchange. Por fim, serão apresentados trabalhos relacionados ao tema, abordando produtividade, fatores humanos e o uso de literatura cinza na engenharia de software.

### 2.1 Produtividade

A definição de produtividade em seu sentido mais amplo, econômico e social, surgiu pela primeira vez em 1950. Alguns pesquisadores definiram produtividade como a relação entre bens e serviços produzidos (Gaither e Frazier, 2002). Atualmente, embora não haja um consenso, a maioria dos pesquisadores concorda que a produtividade pode ser definida como a relação entre a saída de produtos e serviços produzidos e a entrada de recursos (sejam humanos ou não-humanos) utilizados no processo de desenvolvimento (Clincy, 2003). Entretanto, as entradas e saídas variam amplamente entre áreas distintas. Por exemplo, um profissional de engenharia pode definir produtividade como a relação entre a quantidade de produtos produzidos em função do tempo gasto para produzi-los, enquanto um contador poderia definir o termo como sendo a relação entre o lucro bruto e o investimento total (Ramírez e Nembhard, 2004). Assim, uma definição que considere apenas entradas e saídas pode ser considerada simplista.

Além da dificuldade em definir o conceito de produtividade, este termo é frequentemente tratado como sinônimo de eficiência e eficácia. No entanto, enquanto a produtividade descreve a relação entre os resultados e os recursos utilizados para

produzi-los, a eficiência compara o que foi produzido, considerando os recursos disponíveis, com o que poderia ser produzido utilizando esses mesmos recursos. Por outro lado, a eficácia pode ser definida como a capacidade de alcançar determinado resultado pretendido, ou seja, aos objetivos e interesses de uma organização, projeto ou equipe (Yilmaz et al., 2016).

Muitas organizações têm se concentrado na produtividade de seus colaboradores em todo o mundo. No entanto, o investimento na produtividade pode ser mal direcionado e ineficaz, uma vez que as empresas muitas vezes investem em máquinas ou outros recursos, esquecendo-se dos profissionais que devem ser o ponto central quando se trata de produtividade, principalmente na área de software, onde os profissionais trabalham com intelecto (Ribeiro et al., 2006).

No atual cenário capitalista, a produtividade tornou-se um fator decisivo para manter a competitividade no mercado, devido à grande quantidade de bens e serviços semelhantes (MacCormack et al., 2003). Assim, empresas que conseguem fazer mais com menos podem oferecer preços mais acessíveis e produtos de melhor qualidade aos clientes. Por isso, a produtividade no desenvolvimento de software tornou-se um foco de pesquisadores e organizações no mercado de trabalho.

### **2.1.1 Produtividade no desenvolvimento de software**

A produtividade no desenvolvimento de software é um tema relevante e desafiador há décadas, como destacado por Boehm (1981) em seu trabalho *“Software Engineering Economics”*. Boehm argumenta que a produtividade nesse campo é significativamente influenciada por fatores humanos, técnicos e organizacionais, devido à natureza criativa e intelectualmente exigente desse tipo de trabalho. Esse cenário complexo torna a definição e a mensuração de produtividade tarefas que evoluíram ao longo do tempo, refletindo as mudanças na abordagem ao desenvolver software.

Historicamente, a produtividade na engenharia de software foi definida como a “relação entre a quantidade de software produzido e a despesa ou trabalho para produzi-la” (Canedo e Santos, 2019). Em um esforço inicial para quantificar essa relação, métricas como linhas de código escritas por hora (LOC, do inglês *Lines of Code*) e o número de pontos de função implementados foram amplamente utilizadas. Embora essas métricas oferecessem um ponto de partida, tornaram-se insuficientes com o amadurecimento da área, já que elas não refletem adequadamente o valor funcional entregue aos consumidores do software.

Essa limitação surge porque, diferentemente de outros setores, o produto final no desenvolvimento de software não pode ser avaliado diretamente pelo volume de produção. Consumidores, por exemplo, não percebem o número de linhas de código como uma medida de qualidade ou utilidade de um sistema. Assim, houve um movimento na Engenharia de Software em direção a uma nova definição de produtividade que considera o valor efetivamente entregue ao cliente e o impacto do software no contexto em que é utilizado (Sadowski e Zimmermann, 2019). Essa evolução des-

taca a complexidade e a importância do tema no contexto do desenvolvimento de software.

Com base nesse contexto, a produtividade pode ser definida como a relação entre o valor funcional do software produzido e o trabalho ou despesa para produzi-lo (Trendowicz e Münch, 2009). Dessa forma, é possível considerar como valor funcional diversos elementos antes não considerados, como a qualidade do produto, a inovação embutida, o valor para o cliente, dentre outros. Por outro lado, como despesas poderiam ser considerados os custos do projeto (com pessoal ou recursos tecnológicos) e a qualidade de vida das pessoas envolvidas (Sadowski e Zimmermann, 2019).

As medições de produtividade utilizando linhas de código tendem a enfatizar grandes programas, em vez de programas com melhor desempenho, eficiência e qualidade. Como consequência, há muitas organizações que ainda se baseiam em uma perspectiva quantitativa da produção. Isso ocorre, principalmente, porque a produtividade está associada ao custo de desenvolvimento, que é fortemente influenciado pelo tamanho do software (Oliveira et al., 2018).

Na prática, melhorar a produtividade significa produzir os mesmos sistemas com a mesma qualidade, mas com um número reduzido de recursos. Por isso, este é um dos principais objetivos no desenvolvimento de software, uma vez que as organizações buscam produzir mais software, reduzindo os custos do processo (Oliveira et al., 2016). No entanto, como mencionado anteriormente, a produtividade é determinada pela interação de muitos fatores, de modo que nenhum fator em especial é capaz de garantir a alta produtividade em um projeto de software (Canedo e Santos, 2019).

Este trabalho adota uma visão ampliada e qualitativa de produtividade no desenvolvimento de software, considerando-a como a relação entre o valor funcional entregue pelo software e os esforços ou despesas necessários para produzi-lo, conforme proposto por Trendowicz e Münch (2009). Essa definição permite incorporar elementos que vão além de métricas quantitativas tradicionais, como linhas de código ou pontos de função, valorizando aspectos como qualidade do produto, inovação, valor para o cliente e, fundamentalmente, o impacto na qualidade de vida dos profissionais envolvidos.

### **2.1.2 Fatores que influenciam a produtividade no desenvolvimento de software**

As organizações têm aumentado sua consciência em relação à importância de compreender os fatores que influenciam a produtividade da equipe envolvida em projetos de software. Entretanto, é importante considerar que o impacto desses fatores pode variar dependendo do contexto e das características individuais dos desenvolvedores, das equipes, dos projetos e das próprias empresas (Clincy, 2003). Além disso, acredita-se que há uma grande quantidade de fatores que influenciam significativamente a produtividade dos desenvolvedores. Portanto, identificá-los e compreender

sua relação com o desempenho dos profissionais não é uma tarefa trivial (Sampaio et al., 2010).

Scacchi (1989) e Clincy (2003) apresentaram uma lista de fatores que afetam a produtividade de software. Scacchi (1989) destacou a importância de variáveis como linguagem de programação, tamanho do projeto, nível de capacidade dos integrantes do time, complexidade do produto, estabilidade das especificações, tempo de entrega, processos de desenvolvimento, ambiente de trabalho, estações de trabalho com recursos computacionais poderosos, ferramentas utilizadas no desenvolvimento e no seu auxílio, reuso, complexidade do projeto e tamanho da equipe. Já Clincy (2003) determinou uma grande quantidade de fatores como sistemas de recompensa, processo de desenvolvimento de software, tecnologia utilizada, envolvimento da equipe, tamanho da equipe, membros da equipe generalistas, engenheiros de software com habilidades em mais de um idioma, redução da burocracia, gestão, membros escolhidos de acordo com habilidades e não com experiência, liberdade para tomar decisões para os membros da equipe, gerar apenas documentação essencial, comunicação frequente, arquitetura e ferramentas utilizadas no projeto.

Segundo Júnior e Meira (2009), cada tipo de organização deve definir seu próprio modelo de medição da produtividade, o que também pode variar de acordo com o tipo do projeto. Além disso, a análise dos fatores que afetam a produtividade deve levar em consideração aspectos tecnológicos e não tecnológicos, tais como as diferenças de desempenho entre os engenheiros de software. Curtis (1991) destaca a importância dos fatores não-tecnológicos na variação da produtividade. Ele argumenta que a tecnologia e as práticas de programação podem controlar apenas uma proporção pequena dessa variação, enquanto fatores não-tecnológicos, como as diferenças de desempenho entre os engenheiros de software, podem exercer um papel significativo.

Apesar das evidências de que os fatores não-tecnológicos exercem um papel importante na produtividade, muitos estudos têm enfatizado principalmente os fatores tecnológicos. Entretanto, compreender os fatores não técnicos pode ajudar as organizações a criar ambientes de trabalho mais favoráveis melhorando a eficiência e o desempenho de seus profissionais.

Oliveira et al. (2016) definiram três categorias que podem ser utilizadas para agrupar os fatores que afetam a produtividade no desenvolvimento de software, sendo elas:

- **Aspectos humanos:** Fatores relacionados às características dos membros que compõem a equipe de desenvolvimento como motivação, comunicação, experiência profissional, etc;
- **Aspectos relacionados a produtos de software:** Fatores relacionados a características do software como linguagem de programação, o tamanho e complexidade do software, entre outros;
- **Aspectos relacionados ao processo de desenvolvimento:** Fatores relacionados

tanto às diretrizes quanto aos procedimentos a serem seguidos pela organização, por exemplo, metodologia de desenvolvimento.

É relevante enfatizar que cada equipe de desenvolvimento deve avaliar cuidadosamente os fatores que impactam a produtividade em seu ambiente e não adotar, sem uma avaliação rigorosa, os fatores utilizados em outros contextos (Petersen, 2011).

Diversos estudiosos da área de Engenharia de Software têm se dedicado a propor catálogos de fatores que influenciam a produtividade no desenvolvimento de software, assim como modelos conceituais que permitem medir essa produtividade Paiva et al. (2010); Wagner e Ruhe (2008). Essas iniciativas visam apoiar a indústria no aprimoramento de suas práticas e na busca por melhores resultados. Uma fonte valiosa de informações para esses estudos é a literatura cinza, que abrange documentos não convencionais, mas que frequentemente contém *insights* práticos e relevantes.

## 2.2 Literatura cinza

A literatura cinza pode ser definida como “documentos produzidos em todos os níveis de governo, acadêmicos, negócios e indústria em formatos impressos e eletrônicos, mas que não são controlados por editores comerciais”, ou seja, onde a publicação não é a atividade principal do corpo produtor (Garousi et al., 2016). Entretanto, o que pode ser considerado literatura cinza pode variar de acordo com o campo de pesquisa. De acordo com Adams et al. (2016), é possível classificar os diferentes tipos de literatura cinzenta em termos de “tons” de cinza, ver Figura 2.1, e agrupar em duas dimensões: Controle e Expertise.

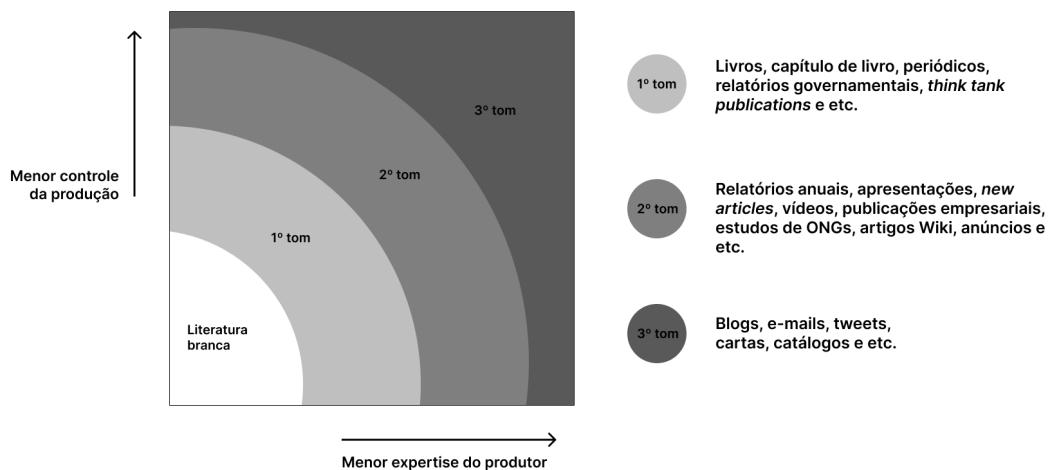


Figura 2.1: Os tons de cinza da literatura. Adaptado de Adams et al. (2016)

A dimensão de Controle refere-se ao grau em que o conteúdo da literatura cinzenta é produzido, moderado ou editado em conformidade com critérios explícitos e trans-

parentes de criação de conhecimento. Isso implica que o conteúdo da literatura cinzenta deve ser produzido seguindo as melhores práticas e padrões de qualidade para garantir a credibilidade e a relevância dos resultados apresentados. O controle da literatura cinzenta é importante para garantir a validade dos resultados da pesquisa e sua aplicabilidade prática. A dimensão de Expertise refere-se ao grau em que podemos determinar a autoridade e o conhecimento do produtor. Isso implica que os autores da literatura cinzenta devem ter um nível de conhecimento especializado sobre o tema tratado, a fim de fornecer informações precisas e relevantes. A expertise é importante porque ela ajuda a avaliar a confiabilidade e a validade dos resultados da pesquisa, bem como a sua aplicação prática (Adams et al., 2016).

As tonalidades da literatura cinzenta, agrupadas em termos de Controle e Expertise, são importantes para entender as diferentes características e qualidades das fontes de informação, fornecendo um quadro útil para a pesquisa acadêmica e a tomada de decisões baseada em evidências.

Garousi et al. (2016) adaptaram o modelo de tons de cinza da literatura cinzenta proposto por Adams et al. (2016) com foco na área de engenharia de software. A adaptação do modelo de tons de cinza permitiu que Garousi et al. (2016) definissem e classificassem os diferentes tipos de literatura cinza de acordo com as necessidades específicas da área. A adaptação desse modelo é apresentada na Figura 2.2. Podemos observar que sites de pergunta e resposta, como o StackOverflow, foram adicionados ao diagrama.

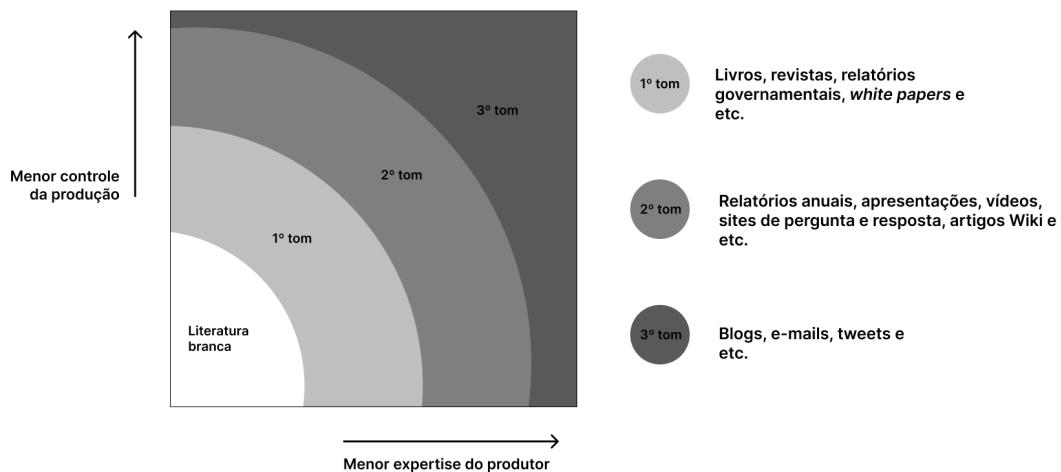


Figura 2.2: Os tons de cinza da literatura para SE. Adaptado de Garousi et al. (2016)

A literatura cinzenta pode ser de grande valor em estudos científicos, especialmente em áreas como engenharia de software, onde há uma forte relação entre a pesquisa e a prática. A literatura cinza muitas vezes fornece informações importantes e práticas

que podem ser aplicadas diretamente no mundo real, como resultados de testes, especificações técnicas, boas práticas e documentação de projetos (Kamei et al., 2021b). Além disso, a literatura cinza pode fornecer *insights* sobre a forma como as organizações pensam e operam, o que pode ser valioso em estudos sobre políticas públicas, gestão empresarial e outras áreas.

### **2.2.1 Importância de considerar a literatura cinza em estudos sobre engenharia de software**

Na área de engenharia de software, a literatura cinza pode ser especialmente importante, uma vez que muitas vezes é produzida por organizações que trabalham diretamente com desenvolvimento de software, como empresas de software, agências governamentais e organizações de padronização. Esses materiais podem fornecer informações importantes sobre o uso de ferramentas e métodos (Garousi et al., 2019).

Muitos pesquisadores acreditam que a literatura acadêmica é suficiente para compreender um tópico de pesquisa. No entanto, a literatura cinza pode fornecer informações valiosas que não estão disponíveis na literatura acadêmica convencional, como resultados de pesquisas não publicadas, dados não divulgados e informações práticas de organizações que podem ser úteis para informar políticas públicas, estratégias empresariais e desenvolvimento de produtos. Além disso, a literatura cinza pode ajudar a preencher lacunas na literatura acadêmica e fornecer uma visão mais ampla do tópico de pesquisa (Bonato, 2018).

O estudo “Assessing the Credibility of Grey Literature: A Study with Brazilian Software Engineering Researchers”, realizado por Kamei et al. (2022) teve como objetivo avaliar a credibilidade da literatura cinza entre pesquisadores brasileiros. Os resultados do estudo indicam que a maioria dos pesquisadores reconhece a importância da literatura cinza, mas muitos têm dificuldade em avaliar sua credibilidade. Entre as principais razões para isso estão a falta de padronização na produção e disseminação desses materiais, a ausência de revisão por pares e a falta de transparência em relação às fontes e metodologias utilizadas.

Para lidar com esses desafios, os pesquisadores sugerem a criação de diretrizes para avaliar a credibilidade da literatura cinza, bem como a inclusão de critérios de avaliação em programas de pós-graduação e em outras iniciativas de formação acadêmica. Além disso, é importante que as organizações que produzem literatura cinza sigam as melhores práticas para garantir a qualidade e credibilidade de seus materiais (Kamei et al., 2022).

Como mencionado anteriormente, a maior parte da literatura cinza é considerada menos prestigiosa do que a publicação em um periódico revisado por pares (Paez, 2017). No entanto, profissionais de engenharia de software produzem constantemente literatura valiosa que não é considerada pela maioria dos pesquisadores, como relatórios técnicos e discussões em fóruns do Stack Exchange. Além disso, a literatura cinza baseada em comunidades online representa bem a voz dos profissionais

em termos de discussões públicas (Kamei et al., 2021a). Esses conteúdos refletem as experiências práticas e as percepções dos profissionais, oferecendo *insights* que muitas vezes estão ausentes na literatura acadêmica tradicional.

### **2.2.2 Stack Exchange**

O Stack Exchange é uma plataforma online que hospeda mais de 100 comunidades de perguntas e respostas sobre diversos tópicos, incluindo disciplinas acadêmicas e práticas profissionais. Cada comunidade é composta por um conjunto de fóruns que permitem aos usuários postar perguntas, fornecer respostas e comentários, além de votar e editar as postagens de outros usuários. O Stack Overflow é a comunidade mais proeminente dentro do Stack Exchange e é dedicado principalmente a questões técnicas relacionadas à codificação de software (Sengupta e Haythornthwaite, 2020). No entanto, existem muitas outras comunidades dentro da plataforma, cada uma com um enfoque distinto. Essas comunidades incluem áreas da Engenharia de Software, como teste de software, qualidade, engenharia reversa e gerenciamento de projetos, bem como tópicos relacionados a disciplinas acadêmicas, como Matemática e Língua Inglesa (Bazelli et al., 2013).

O Stack Exchange tem sido amplamente utilizado por estudantes, acadêmicos e profissionais em todo o mundo para compartilhar conhecimento e colaborar em projetos. As comunidades são compostas por uma mistura de membros novatos e experientes, que trazem perspectivas diversas e contribuem para uma atmosfera de aprendizado colaborativo. Para os estudantes e acadêmicos, o Stack Exchange oferece uma oportunidade de obter informações e *insights* valiosos sobre tópicos específicos, além de se conectar com outros membros da comunidade que compartilham interesses similares. Para profissionais, o Stack Exchange pode ser uma fonte valiosa de informações sobre as melhores práticas da indústria, bem como um lugar para encontrar soluções para desafios específicos (Tahir et al., 2020).

## **2.3 Trabalhos Relacionados**

Nesta seção, serão apresentados trabalhos relacionados aos temas centrais desta pesquisa: produtividade no desenvolvimento de software e o uso de literatura cinza como fonte de dados para estudos na área de engenharia de software. Esses trabalhos fornecem uma base teórica e prática, destacando os desafios, as abordagens e as contribuições existentes.

### **2.3.1 Produtividade e a Engenharia de Software**

Diversos autores têm realizado pesquisas empíricas com o objetivo de identificar os fatores que afetam a produtividade no contexto de desenvolvimento de software. Dentre estes, destacam-se os trabalhos de Wagner e Murphy-Hill (2019), Canedo e Santos (2019), Meyer et al. (2014) e Paiva et al. (2010).

Wagner e Murphy-Hill (2019) apresentam uma lista de fatores que, empiricamente, demonstraram impactar a produtividade no desenvolvimento de software, oferecendo um checklist prático que pode ser utilizado por desenvolvedores e gerentes para melhorar o desempenho das equipes. Os fatores abordados incluem elementos técnicos relacionados ao produto, ao processo e ao ambiente de desenvolvimento, bem como fatores "soft", como a cultura corporativa, a cultura da equipe, as habilidades individuais, as experiências profissionais, o ambiente de trabalho e o projeto em si. O trabalho destaca a complexidade de medir a produtividade em atividades de conhecimento, como o desenvolvimento de software, devido à ausência de entregáveis fáceis e claramente mensuráveis. Além disso, enfatiza que os desenvolvedores enfrentam desafios adicionais decorrentes dos sistemas cada vez maiores e mais complexos com os quais trabalham atualmente.

Canedo e Santos (2019) investigam os fatores que afetam a produtividade em projetos de desenvolvimento de software, incluindo projetos de código aberto. O estudo foi motivado pela crescente competitividade na indústria de software, que demanda ciclos de vida mais otimizados e maior desempenho para garantir a competitividade no mercado. Para responder às questões de pesquisa, os autores realizaram uma Revisão Sistemática da Literatura (SLR) e uma pesquisa empírica com a comunidade de praticantes, explorando suas percepções sobre os fatores que afetam a produtividade da equipe. O estudo revelou fatores interessantes, incluindo aqueles que promovem independência e responsabilidade nas equipes, bem como outros que contribuem para uma melhor distribuição de tarefas. Entre os principais fatores identificados estão pessoas, produto, organização, investimento em tecnologia, falta de relações contratuais e o engajamento de contribuidores em projetos de código aberto. Esses achados destacam a diversidade de influências na produtividade, mostrando como diferentes fatores podem agir de forma complementar ou independente para melhorar o desempenho.

Meyer et al. (2014) exploram as percepções de produtividade dos desenvolvedores de software por meio de dois estudos empíricos: uma pesquisa com 379 desenvolvedores profissionais e um estudo observacional com 11 desenvolvedores. O estudo revelou que os desenvolvedores percebem seus dias como produtivos quando conseguem completar muitas ou grandes tarefas sem interrupções significativas ou trocas de contexto. No entanto, os dados observacionais indicaram que, mesmo enfrentando frequentes mudanças de tarefa e atividade, os participantes ainda se consideravam produtivos. Os autores analisam essas aparentes contradições e propõem maneiras de melhor apoiar os desenvolvedores em sua autorreflexão e melhoria da produtividade, por meio do desenvolvimento de novas ferramentas e da disseminação de melhores práticas. O trabalho também destaca que, à medida que a comunidade de desenvolvimento de software avança na criação de soluções, a demanda global por software cresce, tornando ainda mais essencial compreender como os desenvolvedores, que estão na linha de frente da construção de software, percebem e buscam aprimorar sua produtividade.

Paiva et al. (2010) abordam a medição e melhoria da produtividade de desenvolvedores de software como um dos maiores desafios enfrentados pelas empresas do setor. Para ajudar essas organizações a identificar possíveis causas que interferem na produtividade das equipes, os autores apresentam uma lista de 32 fatores extraídos da literatura que influenciam a produtividade dos desenvolvedores. Além disso, aplicaram um questionário com desenvolvedores para ranquear esses fatores. Os resultados destacam os fatores que possuem maior influência positiva e negativa na produtividade, aqueles que não exercem influência e os mais importantes, além de explorar o que os influencia. Por fim, os autores compararam os resultados obtidos com os encontrados na literatura, oferecendo uma visão abrangente dos elementos que afetam a produtividade no setor de desenvolvimento de software.

### **2.3.2 Literatura cinza e a Engenharia de Software**

Embora existam estudos que investigam os fatores que influenciam a produtividade no setor de software, durante nossas pesquisas não foram encontrados trabalhos que utilizam literatura cinza em suas investigações. Essa é uma lacuna importante, uma vez que a literatura cinza inclui uma variedade de fontes com informações que, muitas vezes, não são publicadas em revistas ou conferências científicas (Garousi et al., 2016). A literatura cinza pode oferecer uma visão mais ampla e diversificada sobre um tema em questão, uma vez que muitas dessas fontes incluem fóruns de discussão, relatórios técnicos e práticas utilizadas por profissionais da área. Além disso, a literatura cinza pode fornecer informações detalhadas sobre a realidade prática e atual das empresas do setor de software, permitindo que os pesquisadores possam identificar desafios e oportunidades para aprimorar diversas áreas do conhecimento (Ahmed e Bagherzadeh, 2018).

O trabalho de Gomes et al. (2023), por exemplo, investigou como profissionais de gerenciamento de projetos lidam com Dívidas Técnicas (TD), a partir de uma análise de 42 discussões sobre TD na Stack Exchange Project Management. Os resultados mostram que as formas mais comuns de TD discutidas pelos profissionais são as relacionadas a processos e pessoas, com 47 indicadores para reconhecer itens de débito. Além disso, foram encontradas 72 práticas relacionadas à gestão de TD. Em complemento, Tahir et al. (2020) investigou como os desenvolvedores discutem “code smells” e “anti-patterns” em três sites da Stack Exchange, destacando que os desenvolvedores frequentemente solicitam a opinião de seus pares para avaliar a qualidade do código. Este estudo revela a necessidade de avaliações mais sensíveis ao contexto e diretrizes mais claras para decisões sobre o uso de design patterns e a eliminação de code smells e anti-patterns na indústria.

A interação entre os fatores técnicos e humanos no desenvolvimento de software continua a ser um tema de grande interesse para a comunidade de pesquisa, especialmente à medida que se busca entender como elementos emocionais e a comunicação informal influenciam a produtividade e a gestão no setor. Nesse contexto, a literatura cinza desempenha um papel essencial ao capturar percepções e experiências

práticas diretamente dos profissionais, oferecendo insights que muitas vezes escapam às investigações acadêmicas tradicionais.

O estudo de Cerqueira et al. (2024) fornece uma análise detalhada sobre a empatia, um aspecto humano crucial para o desenvolvimento de software, mas que ainda é subexplorado em pesquisas. Por meio de uma síntese temática de literatura cinza, os autores analisaram 22 artigos publicados no DEV, uma comunidade online voltada para desenvolvedores de software. O estudo identificou que a empatia possui múltiplos significados para os profissionais da área, abrangendo conceitos como compreensão, compaixão e a capacidade de adotar a perspectiva de outros. Além disso, os resultados apontaram 19 práticas empáticas no contexto da engenharia de software, como a adoção de boas práticas de programação, a capacidade de compreender o próximo e a atenção plena. Entre os 28 efeitos listados dessas práticas, destacam-se a melhoria da qualidade do software, a produção de melhores produtos e o fortalecimento da confiança entre os membros da equipe. Por fim, o trabalho organizou essas descobertas em um framework, que não apenas oferece suporte a novas investigações, mas também pode ser utilizado por profissionais para desenvolver habilidades empáticas no ambiente de trabalho.

Esses estudos reforçam a importância de integrar fatores técnicos e humanos no desenvolvimento de software e destacam a utilidade da literatura cinza, como fóruns e comunidades online, para capturar práticas e desafios reais enfrentados pelos profissionais. Ao explorar as nuances das interações humanas, a literatura cinza complementa a literatura acadêmica, oferecendo uma base rica para estratégias de melhoria contínua e para o desenvolvimento de ambientes de trabalho mais colaborativos e produtivos no setor de software.

# Capítulo 3

## Metodologia

A pesquisa atual adota uma metodologia baseada na técnica síntese temática, conforme delineada por Cruzes e Dyba (2011) em seu estudo “Recommended Steps for Thematic Synthesis in Software Engineering”. A síntese temática é amplamente utilizada em pesquisas qualitativas para identificar, analisar e relatar padrões (temas) dentro dos dados. Este método permite interpretar e explicar esses temas, oferecendo conclusões robustas a partir de análises sistemáticas.

O processo de síntese temática, conforme descrito por Cruzes e Dyba (2011), envolve cinco etapas principais: extração dos dados, codificação dos dados, tradução das codificações em temas, criação de um modelo de categorias para agrupar os temas e avaliação da confiabilidade da síntese. Cada uma dessas etapas foi incluída e adaptada às necessidades específicas desta pesquisa, garantindo uma análise detalhada e consistente dos dados coletados.

A Figura 3.1 apresenta uma visão geral da metodologia utilizada neste estudo. As seções subsequentes fornecerão detalhes específicos sobre os procedimentos e decisões adotados em cada etapa do processo metodológico, assegurando a transparência e a replicabilidade da pesquisa.

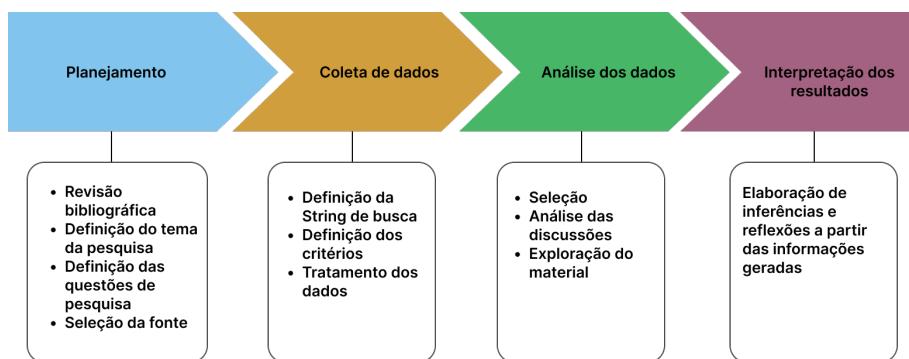


Figura 3.1: Etapas da metodologia

## 3.1 Planejamento

O estudo iniciou com um planejamento sistemático que incluiu uma revisão da literatura, com o objetivo de ampliar o entendimento acerca do estado da arte da pesquisa em torno dos aspectos humanos na Engenharia de Software, bem como identificar possíveis lacunas no campo. Em seguida, foram delimitados o tema do estudo, as questões de pesquisa e selecionadas as fontes de dados. Cada uma das fases do planejamento será descrita de maneira mais detalhada a seguir.

### 3.1.1 Revisão bibliográfica

A etapa inicial da presente pesquisa envolveu a revisão bibliográfica de livros, periódicos, artigos e outros conteúdos relevantes para a área de estudo. Entretanto, este processo não se limita à etapa inicial da pesquisa. É comum que novas informações surjam, demandando a busca de mais literatura para aprimorar e complementar as descobertas. Nesse sentido, apesar de a revisão ter sido a primeira etapa, ela se tornou um processo contínuo. Conforme destacado por Strech e Sofaer (2012), a revisão bibliográfica é crucial para garantir a relevância contínua do problema de pesquisa e para fornecer uma base sólida para a discussão dos resultados.

Inicialmente, foram realizadas buscas por estudos que abordam aspectos humanos que influenciam o desenvolvimento de software. Esses estudos foram selecionados e submetidos à análise e discussão por um grupo de pesquisadores com diferentes níveis de experiência, com o objetivo de avaliar suas contribuições e identificar possíveis lacunas de pesquisa.

Posteriormente, o estudo direcionou-se à temática da produtividade, concentrando esforços na obtenção de conhecimentos básicos e fundamentais sobre o assunto, tais como a definição de produtividade e as métricas utilizadas para sua medição no desenvolvimento de software. Para isso, foi realizada uma busca por revisões sistemáticas da literatura relacionadas à temática, que serviram como ponto de partida por organizarem o conhecimento sobre o assunto (Dutra et al., 2015; Canedo e Santos, 2019; Paiva et al., 2010). Em seguida, os trabalhos mais relevantes foram selecionados para análise mais aprofundada (Wagner e Murphy-Hill, 2019; Petersen, 2011; Hernández-López et al., 2013). Com base nesses estudos, foi possível obter conhecimento suficiente para delimitar o escopo da pesquisa.

### 3.1.2 Escopo da pesquisa

Após o início da revisão bibliográfica, constatou-se a existência de estudos que abordam fatores que influenciam a produtividade dos desenvolvedores de software (Sam-paio et al., 2010; Oliveira et al., 2018; Canedo e Santos, 2019; Trendowicz e Münch, 2009). No entanto, notou-se a ausência de trabalhos que analisem esses fatores por meio da literatura cinza. A exploração desse tipo de literatura pode fornecer uma perspectiva alternativa sobre o assunto que, para determinadas situações, pode es-

tar mais próxima da realidade vivenciada pelos profissionais da área (Garousi et al., 2016). Assim, optou-se por incluir a literatura cinza como fonte de dados na presente pesquisa. Diante disso, o tema de pesquisa foi estabelecido como "Investigando a percepção de praticantes da Engenharia de Software sobre produtividade: uma análise com base na literatura cinza". Após ser realizada a delimitação do tema, as questões de pesquisa foram definidas.

### 3.1.3 Questões de pesquisa

A partir deste estudo, buscou-se compreender como diversos fatores podem afetar a produtividade dos desenvolvedores de software e qual é a percepção desses profissionais em relação a essa temática. Inicialmente, foram analisadas pesquisas que destacam os fatores que impactam a produtividade em projetos de software, constatando-se que esses fatores estão diretamente relacionados aos desafios enfrentados pelos profissionais em seu dia a dia. Para identificar esses fatores, é crucial compreender os desafios e como eles afetam tanto os desenvolvedores quanto suas tarefas cotidianas (Chapetta e Travassos, 2020; Wagner e Murphy-Hill, 2019; Canedo e Santos, 2019). Além disso, tornou-se fundamental avaliar as estratégias adotadas pelos profissionais para superar esses desafios e atingir a produtividade desejada.

Com base nessas análises, foram definidas quatro questões de pesquisa:

**RQ1:** Quais são os desafios de produtividade enfrentados pelos desenvolvedores de software?

Esta questão tem como objetivo identificar os principais desafios de produtividade enfrentados pelos desenvolvedores de software nas comunidades analisadas. A partir dessa identificação, busca-se compreender os fatores que dão origem a esses desafios, criando uma base para investigar como eles impactam os profissionais e suas tarefas, além de explorar as soluções propostas pela comunidade para mitigar tais dificuldades.

**RQ2:** Como os desafios de produtividade afetam os desenvolvedores de software?

Esta questão busca compreender os impactos dos desafios de produtividade na vida dos desenvolvedores de software, explorando aspectos humanos e emocionais, como as experiências, percepções e sentimentos dos profissionais ao lidar com essas dificuldades. O objetivo é captar como esses desafios influenciam o bem-estar, a motivação e a dinâmica de trabalho dos indivíduos nas comunidades analisadas.

**RQ3:** Como os desafios de produtividade afetam o trabalho realizado pelos desenvolvedores de software?

O objetivo desta questão é investigar como os desafios de produtividade impactam diretamente as atividades e os resultados do trabalho dos desenvolvedores de software. Busca-se analisar as consequências desses desafios na execução das tarefas e na qualidade das entregas, considerando as implicações práticas para o desempenho e a eficiência dos profissionais.

**RQ4:** Quais estratégias os desenvolvedores utilizam para superar os desafios de produtividade?

A identificação de estratégias para superar desafios de produtividade é um tópico crucial tanto para a prática quanto para a pesquisa em engenharia de software. Apesar de a produtividade ser amplamente reconhecida como um aspecto essencial no desenvolvimento de software, há uma lacuna significativa na literatura acadêmica em relação às práticas específicas adotadas por desenvolvedores para lidar com esses desafios. Muitos estudos focam nos fatores que afetam a produtividade ou na medição dessa produtividade, mas poucos exploram como os profissionais enfrentam esses obstáculos (Meyer et al., 2019; Trendowicz e Münch, 2009).

Os fóruns de discussão, como os do Stack Exchange, são ambientes ricos onde desenvolvedores compartilham soluções práticas e experiências sobre problemas do cotidiano, incluindo aqueles relacionados à produtividade. Trabalhos como o de Gomes et al. (2023) demonstraram o potencial da literatura cinza para revelar práticas valiosas adotadas pelos profissionais de software, como as estratégias para minimizar dívidas técnicas. No entanto, a literatura cinza ainda é pouco explorada para identificar as técnicas empregadas pelos desenvolvedores para superar desafios de produtividade.

Portanto, esta questão tem como objetivo preencher essa lacuna, investigando as soluções propostas pelos profissionais em fóruns de discussão e analisando como essas comunidades oferecem suporte aos indivíduos que enfrentam dificuldades relacionadas à produtividade. Além de complementar a literatura acadêmica com uma perspectiva prática, os achados desta pesquisa podem abrir novos caminhos para estudos futuros e para o desenvolvimento de ferramentas e métodos que otimizem a produtividade no setor de software.

### 3.1.4 Seleção da fonte

A escolha da fonte de pesquisa deve ser feita de forma estratégica e criteriosa, levando em consideração a adequação da fonte ao tema da pesquisa, a relevância das informações obtidas e a possibilidade de contribuir para a evolução do conhecimento científico em uma determinada área (Adams et al., 2016).

Na etapa de seleção da fonte para o presente estudo, após a definição do tema, identificou-se que os fóruns de discussão poderiam ser um ambiente ideal para encontrar problemas reais vividos pelos profissionais da área de desenvolvimento de software (Sengupta e Haythornthwaite, 2020). Essa escolha foi motivada pelo fato de que, nesses fóruns, os profissionais discutem suas dificuldades, compartilham soluções e trocam conhecimentos, tornando-os uma fonte rica e relevante de informações para a pesquisa (Kamei et al., 2019).

Além disso, como a revisão bibliográfica realizada não encontrou estudos que investigam fatores que afetam a produtividade utilizando literatura cinza. A escolha dos fóruns de discussão como fonte do estudo permite a coleta de informações ainda

não amplamente exploradas na literatura acadêmica. A utilização de fontes não convencionais de informação, como fóruns de discussão, pode trazer contribuições significativas para o avanço do conhecimento na área (Kamei et al., 2022).

A escolha de responder à questão de pesquisa utilizando um fórum de discussão também se deve ao fato de que, nesses ambientes, os profissionais de desenvolvimento de software se sentem mais livres para expressar seus sentimentos frente a um problema ou situação (Ahmed e Bagherzadeh, 2018). Além disso, ao analisar estudos em outras áreas da Engenharia de Software (Gomes et al., 2023), constatou-se que os fóruns permitem a análise de um volume significativamente maior de dados em comparação com as respostas obtidas por meio de questionários, por exemplo.

### **Escolha das comunidades do Stack Exchange**

Nesta fase da pesquisa, foram selecionadas comunidades na plataforma Stack Exchange com o objetivo de investigar os fatores que influenciam a produtividade no desenvolvimento de software. Optou-se pelo uso do Stack Exchange devido à sua relevância entre os profissionais da área e à facilidade de exportação de dados através do Stack Exchange Data Explorer (Gomes et al., 2023).

Dentre as várias comunidades relacionadas ao desenvolvimento de software, foram selecionadas as comunidades Software Engineering<sup>1</sup> e Project Management<sup>2</sup>. A comunidade Software Engineering é voltada para perguntas e respostas de profissionais, acadêmicos e estudantes envolvidos no ciclo de desenvolvimento de software, o que a torna uma escolha adequada para a pesquisa em questão.

Por sua vez, a comunidade Project Management tem sido explorada em outros estudos e apresenta discussões com informações relevantes para a comunidade de Engenharia de Software (Gomes et al., 2023). Essa comunidade é voltada para gestores de projetos, profissionais que exercem grande influência sobre as equipes de desenvolvimento. A análise dos dados dessa comunidade contribui significativamente para a identificação de fatores que afetam a produtividade em projetos de software, sob a perspectiva dos gestores de projetos.

O Stack Exchange mantém uma área denominada “Área 51”<sup>3</sup> que permite a solicitação de criação de novas comunidades e monitora um conjunto de métricas para avaliar o desempenho das comunidades existentes. Conforme divulgado pelo próprio site, as comunidades com uma taxa de resposta acima de 3,0 são consideradas “Boa”, acima ou igual a 1,0 são “Ok, mas precisa melhorar” e abaixo de 1,0 são consideradas “Ruim”. A Tabela 3.1 apresenta dados que evidenciam a dimensão das comunidades selecionadas e suas respectivas taxas de resposta (com informações obtidas em 27/06/2022).

---

<sup>1</sup><https://softwareengineering.stackexchange.com>

<sup>2</sup><https://pm.stackexchange.com>

<sup>3</sup><https://area51.stackexchange.com>

Tabela 3.1: Dimensão das comunidades investigadas.

Comunidade	Questões	Respostas	Comentários	Taxa de resposta
Project Manager	6.2 mil	19 mil	27 mil	3,1
Software Engineering	61 mil	173 mil	523 mil	7,0

Conforme apresentado na Tabela 3.1, as comunidades escolhidas para a presente pesquisa foram classificadas como “Boas” de acordo com a taxa de resposta avaliada pelo site Stack Exchange. A decisão de não utilizar a comunidade mais popular do Stack Exchange, a Stack Overflow, foi baseada na predominância de perguntas técnicas sobre codificação, as quais não estão em conformidade com o escopo do presente estudo (Bazelli et al., 2013).

## 3.2 Coleta de dados

A coleta de dados foi conduzida por meio da plataforma Stack Exchange Data Explorer (SEDE), que permite realizar consultas e acessar informações de qualquer site do Stack Exchange, incluindo suas postagens (perguntas e respostas), comentários e metadados.

A SEDE oferece um ambiente virtual para a consulta e extração de dados contidos nos diversos fóruns que compõem o Stack Exchange. Para a obtenção de dados relevantes, é possível utilizar uma *string* de busca contendo as palavras-chave pertinentes ao tema de interesse, de modo a restringir os resultados obtidos e aumentar a eficiência na busca.

### 3.2.1 *String* de busca

Para definir a *string* de busca a ser utilizada na Stack Exchange Data Explorer, foram realizados testes com diferentes termos, seguidos da análise de uma amostra dos resultados obtidos em cada teste. Dessa forma, foram identificados dois termos promissores: “*productivity*” e “*development performance*”. Para melhorar a precisão dos resultados, foi especificado que a busca por esses termos deveria considerar o título, as *tags* e o corpo dos *posts*.

A fim de obter uma busca mais abrangente sobre o tema produtividade no desenvolvimento de software, foram adicionados mais dois termos que são frequentemente considerados sinônimos de produtividade: “*development effectiveness*” e “*development efficiency*”. Essa escolha foi baseada em uma análise de strings de busca de revisões sistemáticas sobre produtividade (Chapetta e Travassos, 2020; Wagner e Murphy-Hill, 2019; Canedo e Santos, 2019). Dessa forma, a *string* de busca utilizada no Stack Exchange Data Explorer foi definida como:

*title contains ('productivity' OR 'development performance' OR 'development effectiveness' OR 'development efficiency') OR tags contains ('productivity' OR 'development effectiveness' OR 'development efficiency')*

*lopment performance' OR 'development effectiveness' OR 'development efficiency') OR body contains ('productivity' OR 'development performance' OR 'development effectiveness' OR 'development efficiency')*

Após a exportação dos dados utilizando a *string* de busca acima, foram aplicados critérios de inclusão e exclusão para filtrar as discussões encontradas.

### 3.2.2 Critérios de inclusão e exclusão

Com o objetivo de selecionar as discussões mais relevantes e de maior qualidade para este estudo, foram estabelecidos critérios de inclusão e exclusão baseados nas perguntas que originaram as discussões. A maioria desses critérios foi baseada no estudo realizado por Gomes et al. (2023) e analisada pelos pesquisadores envolvidos, que os consideraram adequados para o presente estudo. No entanto, para permitir um filtro temporal, adicionou-se um critério relacionado à data de criação da pergunta que gerou a discussão. Abaixo, estão listados os critérios estabelecidos:

#### Critérios de inclusão das discussões

- A discussão deve ter pelo menos duas postagens (uma pergunta e uma resposta);
- A discussão deve ter pelo menos duas pessoas envolvidas, ou seja, a pergunta precisa ser respondida por outra pessoa que não seja o próprio autor da pergunta;
- A pontuação da pergunta (*upvotes*<sup>4</sup>) precisa ser maior que zero;
- A discussão deve estar relacionada à produtividade;
- A data de criação da pergunta é maior igual ou maior a 01/01/2012.

#### Critérios de exclusão das discussões

- A discussão contém apenas uma pergunta;
- A discussão não envolve duas ou mais pessoas, ou seja, quem perguntou é a mesma pessoa que respondeu;
- A pontuação da discussão (*upvotes*) é menor que um;
- A discussão não está relacionada à produtividade;
- A data de criação da pergunta é menor que 01/01/2012.

---

<sup>4</sup>Os *upvotes* são uma forma de avaliação positiva de conteúdo em fóruns de discussão como o Stack Exchange. Os usuários do site podem votar em uma pergunta ou resposta que consideram útil ou relevante, clicando na seta para cima ao lado do conteúdo. Quando um usuário recebe um *upvote* em uma pergunta ou resposta, isso indica que outros membros da comunidade consideram a contribuição valiosa e relevante para a discussão em questão.

Em virtude do número expressivo de respostas a determinadas perguntas, foi necessário adicionar um critério para a análise das mesmas. Durante a etapa de análise das discussões, descrita na seção 3.3.2, somente foram consideradas as respostas que receberam, no mínimo, uma validação positiva de outro membro da comunidade. Essa estratégia permitiu reduzir o volume de respostas a serem analisadas e possibilitou uma avaliação mais criteriosa, priorizando a qualidade das contribuições apresentadas pelos membros da comunidade.

### 3.2.3 Tratamento dos dados

Conforme mencionado anteriormente, para coletar os dados, foi realizada uma busca no Stack Exchange Data Explorer utilizando uma *string* de busca. Essa busca retornou 211 discussões para o fórum Software Engineering e 65 discussões para o fórum Project Management. Para analisar esses dados, foram criadas planilhas distintas para cada fórum, nas quais cada linha corresponde a uma discussão. Parte dessas planilhas foi automaticamente preenchida com informações extraídas da base de dados do Stack Exchange.

As colunas preenchidas automaticamente são:

1. Id da discussão;
2. Link da discussão;
3. Data de início da discussão;
4. Data da última atividade na discussão;
5. Quantidade de visualização da discussão;
6. Quantidade de *upvotes*;
7. Tags;
8. Quantidade de respostas;
9. Quantidade de comentários;

Inicialmente, as informações como *tags* e a quantidade de visualizações, *upvotes*, respostas e comentários foram coletadas com o intuito de auxiliar na filtragem e seleção das discussões a fim de criar uma amostra a ser analisada. Entretanto, durante a etapa de seleção das discussões, discutida na seção 3.3.1, optou-se por não utilizar essas informações, mas sim fazer uma leitura de todas as perguntas que originaram as discussões para selecionar um conjunto de discussões mais relevantes para este estudo. Essa abordagem será discutida na seção 3.3.1.

Adicionalmente, foram criadas 7 colunas na planilha para que os revisores pudessem preenchê-las durante a etapa de análise. Essas colunas são listadas abaixo:

1. Resumo da pergunta;

2. É um caso positivo?;
3. Justificativa do falso positivo;
4. Quais desafios de produtividade são discutidos?
5. Como esses desafios afetam os desenvolvedores?
6. Como esses desafios afetam a produtividade da tarefa?
7. Quais estratégias para melhorar a produtividade são discutidas?

Na coluna “resumo da pergunta”, os pesquisadores registraram uma breve descrição da pergunta que gerou a discussão. Na coluna ”É um caso positivo?”, os pesquisadores indicaram se a discussão tratava de um desafio de produtividade envolvendo profissionais do desenvolvimento de software, sinalizando como “Positivo” em caso afirmativo. Caso contrário, o motivo do falso positivo foi explicado na coluna “Justificativa do falso positivo”.

As perguntas de 4 a 7 referem-se às questões de pesquisa deste trabalho. As perguntas de 4 a 6 foram respondidas com base na pergunta que originou a discussão. Por outro lado, a questão 7 foi respondida com base nas respostas e na pergunta, caso a pessoa que fez a pergunta tenha mencionado alguma estratégia que utilizou.

Com o intuito de fornecer suporte às respostas dadas às perguntas de 4 a 7, foi acrescentada uma coluna denominada “Quote” para cada uma das questões. Nessa coluna, os pesquisadores preencheram com a citação textual da discussão que fundamentava sua resposta.

Na Figura 3.2, pode ser observado um resumo do processo de coleta e tratamento dos dados para que sejam utilizados no processo de análise.

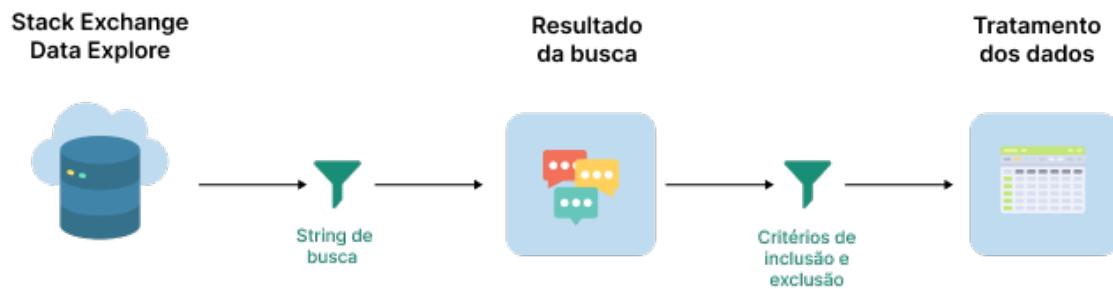


Figura 3.2: Resumo do processo de coleta dos dados.

### 3.3 Análise dos dados

Para otimizar o processo, a análise dos dados do fórum Project Management foi realizada após a conclusão da análise do fórum Software Engineering. Ambos os fóruns foram analisados utilizando os mesmos passos, que serão descritos a seguir.

### 3.3.1 Seleção das discussões

Após a organização das discussões recuperadas pela *string* de busca no fórum de Software Engineering em uma planilha, o processo de seleção dos dados teve início. Nesta fase, todas as 211 perguntas que geraram as discussões foram analisadas, enquanto as respostas e comentários foram desconsiderados. Posteriormente, o mesmo procedimento foi aplicado ao fórum Project Management. Todas as 65 perguntas que geraram as discussões foram analisadas, utilizando a mesma abordagem descrita anteriormente.

Neste estudo, a etapa de seleção foi adaptada com o objetivo de selecionar um conjunto de discussões com maior potencial de relevância, considerando a limitação de tempo e recursos humanos disponíveis para uma análise completa (pergunta e respostas) das discussões. Dessa forma, durante a etapa de análise subsequente, os revisores puderam dedicar todo o tempo disponível para analisar os casos com potencial de serem positivos, sem precisar se ocupar com muitos casos de falso positivo. Adicionalmente, é provável que o número de casos positivos analisados tenha sido maior do que se uma amostra aleatória tivesse sido escolhida.

Durante a etapa de seleção de cada fórum, todas as perguntas foram inicialmente classificadas por um revisor em uma das quatro categorias: “Falso positivo”, “Positivo”, “Provavelmente falso positivo” ou “Provavelmente positivo”. As categorias “Provavelmente falso positivo” e “Provavelmente positivo” foram incluídas para evitar uma classificação forçada em casos de incerteza. Posteriormente, um segundo revisor analisou os resultados, focando especialmente nos casos classificados como “Provavelmente falso positivo” ou “Provavelmente positivo”, e ajustou essas classificações para “Positivo” ou “Falso positivo”.

Foram classificadas como “Falso positivo” as perguntas que não descreviam problemas de produtividade enfrentados por profissionais do desenvolvimento de software. Por exemplo, muitas questões apresentavam o termo “*productivity*”, mas o autor se referia à produtividade do uso específico de uma determinada ferramenta.

Após a conclusão das etapas de seleção, um total de 68 perguntas do fórum Software Engineering e 33 perguntas do fórum Project Management foram identificadas como positivas e selecionadas para a fase de análise subsequente. É crucial destacar que a etapa de seleção teve como objetivo exclusivamente selecionar o conjunto de discussões a serem analisadas. Durante a etapa de análise, discutida na próxima seção, cada discussão foi examinada minuciosamente, permitindo ao revisor alterar o status de positivo para falso positivo caso sua revisão apontasse para essa conclusão.

### 3.3.2 Análise das discussões

Após a etapa de seleção, as 68 discussões do fórum Software Engineering que apresentaram potencial para integrar o presente estudo foram distribuídas em dois conjuntos. Cada conjunto foi submetido a uma análise realizada por uma dupla de pesquisadores. Cada pesquisador examinou seu conjunto de discussões de forma

individual e foi encorajado a registrar as informações de modo livre, nas 8 colunas descritas na seção 3.2.3, sem qualquer forma de codificação, a fim de evitar a perda de informações importantes durante o processo de análise. No caso do fórum Project Management, as 33 discussões foram analisadas apenas por uma dupla de pesquisadores, seguindo o mesmo procedimento.

Um exemplo de discussão é apresentado na Figura 3.3. Partes da mesma foram removidas com o intuito de preservar a clareza. É possível identificar na figura a estrutura fundamental de uma discussão, que consiste no título da pergunta, no corpo da pergunta, nas respostas recebidas, bem como em suas respectivas avaliações, e nos comentários postados pelos usuários, que igualmente contam com um corpo textual e uma pontuação associada.

Ao término da fase de análise das discussões, cada dupla de avaliadores realizou um consenso das suas análises individuais. Caso não houvesse concordância entre as revisões, uma terceiro pesquisador foi designado para avaliar as discussões e emitir um parecer final. Uma síntese desse procedimento encontra-se ilustrada na Figura 3.4.

### 3.3.3 Codificação e categorização

Este estudo utilizou uma abordagem qualitativa para transformar as discussões em dados passíveis de análise, por meio do método de síntese temática. Esta técnica sistemática de interpretação e categorização de dados qualitativos envolve a identificação de unidades de análise relevantes, a codificação dessas unidades e a organização em categorias mais amplas (Cruzes e Dyba, 2011). O processo de codificação e categorização ocorreu após a análise das discussões e é exemplificado na seção 3.4.

No decorrer do estudo, um dos pesquisadores foi responsável pelo processo de codificação e categorização, criando termos para representar e agrupar os indicadores. A codificação foi posteriormente revisada por outro pesquisador para garantir a precisão e a consistência dos dados. O processo é exemplificado na seção 3.4.4.

## 3.4 Execução

O processo de busca de discussões para o estudo iniciou-se em 10/07/2022 e envolveu a seleção de termos para compor uma *string* de busca. O termo “*productivity*” foi o primeiro a ser incluído, seguido de sinônimos de produtividade identificados em pesquisas de revisão sistemática da literatura, como “*development performance*”, “*development effectiveness*” e “*development efficiency*”. Embora esses termos tenham produzido resultados positivos em algumas fontes de literatura acadêmica, a análise dos dados obtidos no Stack Exchange Data Explorer indicou que o termo “*productivity*” foi responsável por mais de 90% das discussões retornadas.

É importante ressaltar que a seleção adequada de termos para uma *string* de busca é um aspecto crítico de qualquer estudo baseado em dados. A inclusão de sinônimi-

### How to keep your productivity level high when people keep changing requirements at the last minute?

Título

Asked 4 years, 8 months ago Modified 4 years, 8 months ago Viewed 611 times

Upvotes

Corpo da pergunta



I started working as a software engineer with a startup about one and a half year ago. Everyone here cares about this project and works hard. but I have been having an issue here for some time. People here have "lack of discipline" when it comes to development tasks. They ask for one functionality, I ask a thousand questions so that I am clear on the requirements of the functionality. As I don't ever like to miss a single scenario when collecting requirements. and I do my work with all my heart and brain in it.

Later on the day of the demo, they would say: let's change this to that and that to this. 5 demos later, trying to fit in their changed requirements and flow, I end up with a mess of code that works the way they want.

Now, because the flow is totally changed, it's not even the same function that I started with, there's delay in launch, there are bugs in production and marketing people are screaming about it. I always end up in some mess. and I am tired of it.

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. I like this project, I really want to stay here for a bit longer and I want to give my best even if I am working on something for the 5th time.

Any advice on keeping the productivity level high in these scenarios? How do I make sure that I don't miss anything while changing the functions? and what do I do when I feel stuck.

p.s. ATM I am making changes in the flow of a user portal that I actually finished flawlessly last week. and now I am stuck, I don't feel like doing it. and I can't even think how to fix these issues.

productivity development

Tags

4 Answers

Sorted by: Highest score (default)

Resposta



The idea that requirements are fixed and that the code you are writing will not have to be changed is extremely problematic. In reality, requirements are never fixed. Whether you need to change it prior to the initial release or not, at some point in the future, your code is very likely going to need to change. When that ceases to be true, it's because the code base is abandoned.

In your situation, you've identified that 1. your initial requirements are not what you will eventually deliver and 2. that you can't affect any change to that reality. All of this points to one thing: you need to optimize your development strategy around the ability to change. A lot of software practices you will read about here are either about the ability to understand code later, the ease of modification, or both.

The first thing I would suggest is that you stop attempting to build a fully complete polished system before showing it to your product owners.

Figura 3.3: Exemplo de discussão com resposta, upvotes e tags.

mos relevantes para o tópico de pesquisa pode melhorar a sensibilidade da busca, aumentando a probabilidade de que todas as discussões relevantes sejam incluídas. No entanto, é fundamental avaliar a eficácia de cada termo individualmente para

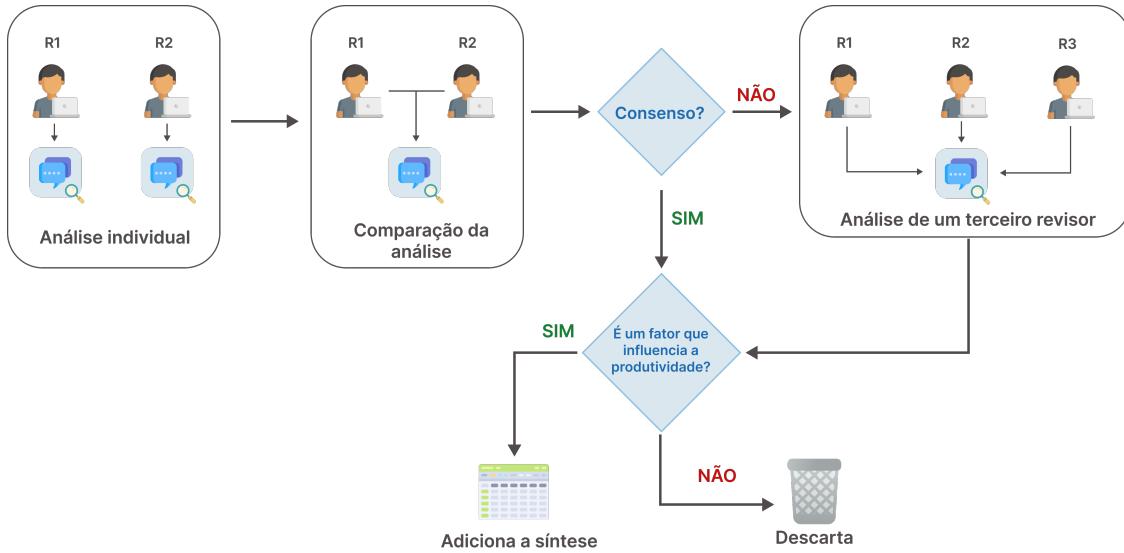


Figura 3.4: Fluxograma resumido do processo de análise das discussões. Adaptado de Gomes et al. (2023).

determinar se sua inclusão é realmente necessária (Kitchenham e Charters, 2007).

Neste caso, a análise dos resultados indicou que o termo *"productivity"* seria suficiente para identificar a grande maioria das discussões relevantes para o estudo em questão, com mais de 90% das discussões vindo desse termo.

Após a composição da *string* de busca, foi feita a exportação dos dados para um arquivo .CSV utilizando o Stack Exchange Data Explorer. A planilha utilizada por Gomes et al. (2023) foi utilizada como base, sendo feitas melhorias e alterações, como a adição da coluna “quote” para cada questão de pesquisa.

Durante o tratamento dos dados, as colunas que continham informações como id da discussão, link da discussão, data de início e data da última atividade na discussão, quantidade de visualizações, quantidade de *upvotes*, *tags*, quantidade de respostas e quantidade de comentários foram preenchidas automaticamente com dados oriundos do Stack Exchange Data Explorer. As demais colunas foram adicionadas com células vazias para serem preenchidas durante a etapa de análise.

### 3.4.1 Execução da etapa de seleção dos dados

Após a organização das discussões, foi iniciada a etapa de seleção para ambos os fóruns. No fórum Software Engineering, todas as perguntas que originaram as 211 discussões retornadas pela *string* de busca foram lidas. Destas, 59 foram classificadas como “Positivo”, 11 como “Provavelmente positivo”, 7 como “Provavelmente falso positivo” e 134 como “Falso positivo”.

No fórum Project Management, as perguntas que originaram as 33 discussões retornadas foram analisadas. Destas, 23 foram classificadas como “Positivo”, 2 como

“Provavelmente positivo”, 2 como “Provavelmente falso positivo” e 6 como “Falso positivo”.

Durante a revisão das discussões classificadas como “Provavelmente positivo” por um segundo pesquisador, 8 tiveram sua classificação alterada para “Positivo”. Por outro lado, apenas 1 discussão da categoria “Provavelmente falso positivo” teve sua classificação alterada para “Positivo” após a revisão. Ao final do processo, as 68 discussões classificadas como “Positivo” foram selecionadas para a etapa de análise, enquanto as outras 143 foram excluídas da planilha.

No caso do fórum Project Management, após a etapa de seleção, 33 discussões foram inicialmente classificadas. No entanto, durante a revisão por um segundo pesquisador, apenas 23 discussões permaneceram classificadas como “Positivo”, enquanto as demais foram reclassificadas como “Falso positivo” e, consequentemente, excluídas.

Um resumo das etapas que antecedem a análise das discussões é apresentado na Figura 3.5.

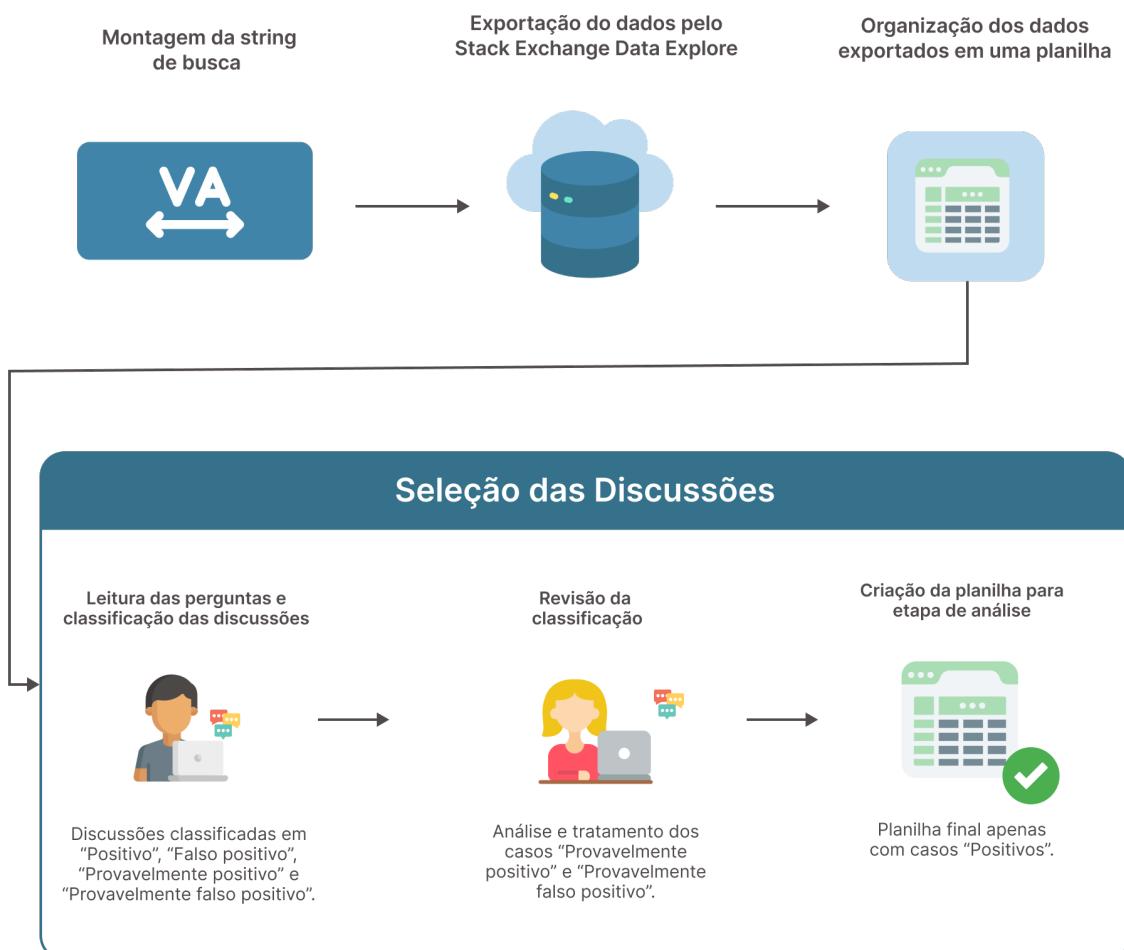


Figura 3.5: Etapas que antecedem a análise das discussões

Antes de realizar a análise completa das 68 discussões do fórum Software Engineering

selecionadas na etapa de seleção, foi conduzida uma importante etapa de padronização e garantia de consistência no processo de revisão. Para tal, duas reuniões de consenso foram realizadas entre os quatro pesquisadores envolvidos na análise. O objetivo dessas reuniões foi selecionar e analisar três discussões em conjunto, a fim de padronizar a abordagem e garantir que todos os revisores preenchessem a planilha de forma consistente. Essa etapa foi fundamental para assegurar a qualidade e a precisão dos resultados da análise posterior. No caso do fórum Project Management, a dupla de revisores realizou o mesmo processo de padronização.

### 3.4.2 Execução da análise dos dados

Após a etapa de alinhamento entre os revisores, as 68 discussões do fórum Software Engineering foram divididas em dois grupos de 34, e formaram-se duas duplas de revisores para dar início ao processo de análise. Cada dupla ficou responsável por examinar um conjunto de 34 discussões. No entanto, cada revisor analisou as discussões individualmente e registrou suas conclusões em uma cópia individual da planilha.

No caso do fórum Project Management, as 23 discussões classificadas como “Positivo” foram analisadas por ambos os revisores da dupla. Eles seguiram o mesmo procedimento de análise individual, registrando suas conclusões em planilhas separadas.

A Figura 3.3 apresenta a pergunta que originou uma discussão no fórum Software Engineering, utilizada para exemplificar o processo de análise realizado. Esta discussão serve como um caso ilustrativo para demonstrar como os dados foram extraídos, analisados e codificados. Ao detalhar este exemplo, será possível compreender melhor os critérios utilizados pelos revisores e a maneira como as informações foram registradas e interpretadas durante a análise.

Conforme pode ser observado na Figura 3.3, a pergunta que originou a discussão se refere a como manter a produtividade alta quando os requisitos costumam ser alterados no último minuto. Na coluna “Resumo da pergunta”, um dos revisores descreveu a situação como “um programador trabalhando em uma empresa que muda constantemente os requisitos do projeto”. Vale ressaltar que a coluna em questão não será codificada e, portanto, não é considerada como resultado do trabalho. Sua função é fornecer uma visão geral da discussão e auxiliar os revisores em casos de necessidade de revisitar a planilha.

Para responder à questão RQ1, o revisor lê a pergunta que originou a discussão e identifica qual desafio de produtividade estava sendo discutido. Na Figura 3.6, a resposta de um dos revisores para a discussão é apresentada. Na coluna “Quote” está o trecho que levou o revisor a esta conclusão.

Na questão RQ2, o revisor identifica aspectos humanos que demonstram como o desenvolvedor se sentia diante do desafio enfrentado. Como pode ser observado na Figura 3.7, em alguns casos, foram identificados vários aspectos, sendo utilizada

RQ1: What productivity challenges are discussed?	RQ1: Quote
1) Desmotivação com o projeto devido a mudanças constantes nos requisitos.	1) Later on the day of the demo, they would say: let's change this to that and that to this. [ ... ] 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[ ... ]

Figura 3.6: Exemplo de resposta extraída para a questão RQ1.

uma numeração para relacioná-los com o trecho da discussão que o revisor usou para identificá-los.

RQ2: How do these challenges affect practitioners (human aspects)?	RQ2: Quote
1) Desmotivação 2) Procrastinação 3) Cansaço mental 4) Sensação de estar empacado	1) e 2) "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.  3) I always end up in some mess and I am tired of it.  4) and what do I do when I feel stuck. I always end up in some mess. and I am tired of it."

Figura 3.7: Exemplo de resposta extraída para a questão RQ2

Na questão RQ3, o revisor identifica como os desafios afetam o trabalho ou tarefa do profissional. Na Figura 3.8, para o desafio apresentado na Figura 3.3, foram identificados quatro aspectos.

Por fim, a questão RQ4 foi analisada de forma semelhante às demais questões, mas com um foco específico nas respostas fornecidas pelos membros da comunidade à pergunta original. Nesse processo, foram examinadas as soluções, estratégias e insights compartilhados pelos participantes da discussão. O objetivo foi identificar como essas respostas abordavam os desafios de produtividade enfrentados e quais métodos ou práticas foram sugeridos para superá-los. Este exame das respostas permitiu uma compreensão mais aprofundada das dinâmicas de interação na comunidade e das abordagens práticas recomendadas pelos membros para lidar com problemas de produtividade no desenvolvimento de software.

A figura 3.9 mostra dados extraídos das respostas dos membros da comunidade para a discussão exibida na Figura 3.3

RQ3: How do these challenges affect task productivity?	RQ3: Quote
1) Código bagunçado 2) Ocorrência de bugs em produção 3) Atraso na entrega 4) Discussões	1) 5 demos later, trying to fit in their changed requirements and flow, I end up with a mess of code that works the way they want. 2) 3) 4) Now, because the flow is totally changed, it's not even the same function that I started with, there's delay in launch, there are bugs in production and marketing people are screaming about it.

Figura 3.8: Exemplo de resposta extraída para a questão RQ3

Como mencionado anteriormente, foram inicialmente identificadas 211 discussões utilizando a string de busca no fórum SE. Após a etapa de etapa de seleção, 68 discussões foram selecionadas para serem analisadas pelos quatro revisores. No entanto, após a análise detalhada realizada por todos os revisores, apenas 57 discussões foram consideradas relevantes para este estudo e utilizadas nas etapas seguintes de codificação. Um resumo desse processo de triagem pode ser visualizado na Figura 3.10.

No caso do fórum PM, foram inicialmente identificadas 65 discussões utilizando a string de busca definida. Após a etapa de etapa de seleção, 33 discussões foram escolhidas para análise pelos quatro revisores. Destas, apenas 23 foram consideradas relevantes para este estudo e utilizadas nas etapas de codificação. O resumo do processo de triagem para o fórum PM pode ser visualizado na Figura 3.11.

### 3.4.3 Divergências nas análises e consenso

Durante o processo de revisão, houve divergências encontradas pelas duas duplas de revisores. A primeira dupla encontrou divergências em oito discussões, enquanto a segunda dupla encontrou divergências em seis discussões.

Para resolver essas divergências, foram realizadas reuniões em que os revisores apresentaram seus argumentos e debateram suas interpretações dos dados. Durante essas discussões, houve um diálogo e uma análise cuidadosa dos resultados, visando alcançar um consenso sobre as divergências encontradas.

Ao final do processo, os revisores foram capazes de chegar a um consenso sobre as divergências sem ser necessário acionar um terceiro revisor.

### 3.4.4 Codificação e classificação

Para codificar o conteúdo analisado na Questão RQ1, foi utilizado um processo interativo. Durante esse processo, o desafio extraído da questão que originou a discussão foi analisado e o fator causador foi identificado. Esse fator, então, foi adicionado a uma lista.

RQ4: What strategies and activities to improve productivity are discussed?	RQ4: Quote
1) Desenvolver com capacidade para mudanças 2) Prototipar e apresentar versões iniciais 3) Receber feedback, implementar as mudanças, mostrar o resultado e repetir o processo. 4) Padronizar o código 5) Modularizar o código 6) Adotar testes unitários	1) You need to optimize your development strategy around the ability to change 2) Stop attempting to build a fully complete polished system before showing it to your product owners. Instead build functioning prototypes and get them in front of the decision makers early 3) Take their feedback, implement them and make things a little more finished. Show it to them again. Rinse and repeat. 4) You should be following the same standards for your code whether it's a last minute change or if your code isn't due for 6 weeks. 5) Modular code: If nothing else, make sure you are breaking code down into short readable methods with a single responsibility and meaningful name. 6) Unit tests: What's often misunderstood about unit tests is that they have relatively little value in determining whether are modifying code.

Figura 3.9: Exemplo de resposta extraída para a questão RQ4.

Cada vez que uma nova questão era analisada, a lista de fatores era revisada. Se nenhum dos fatores existentes na lista se encaixasse como o causador do desafio de produtividade analisado, um novo fator era criado.

A Figura 3.12 apresenta a codificação da Questão RQ1, que resultou no código “Volatilidade dos requisitos”. As questões RQ2 e RQ3 foram codificadas de maneira análoga.

Como mencionado anteriormente, as respostas da questão RQ4 podem considerar toda a discussão, não apenas a pergunta originária. O processo de codificação dos dados extraídos foi realizado da mesma forma que para as demais questões. Um exemplo de codificação é apresentado na Figura 3.13.

Após o processo de codificação, foi realizado um refinamento dos códigos. Códigos semelhantes foram agrupados, resultando na eliminação de alguns códigos redundantes. Por exemplo, na primeira codificação, o fórum Software Engineering gerou 52 códigos diferentes, mas ao final do refinamento, apenas 20 códigos permaneceram. Esse processo de agrupamento e eliminação foi essencial para consolidar os dados e garantir uma análise mais coesa e significativa.

Após essa fase de refinamento, iniciou-se o processo de categorização. O objetivo principal desta etapa é agrupar os códigos em categorias amplas e representativas, promovendo uma organização mais eficiente das informações e possibilitando uma análise mais estruturada dos dados(Cruzes e Dyba, 2011).

A categorização dos códigos é crucial, pois permite a identificação de padrões e ten-

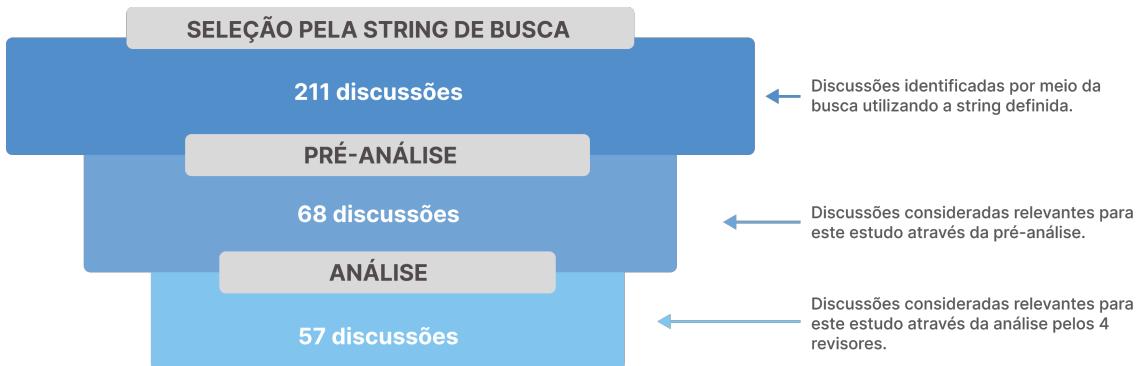


Figura 3.10: Resumo do processo de seleção das discussões no fórum SE.

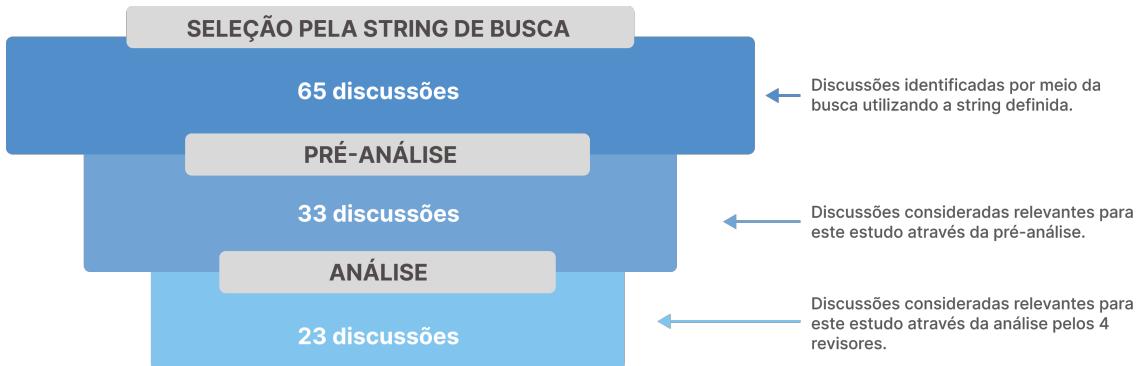


Figura 3.11: Resumo do processo de seleção das discussões no fórum PM.

dências nos dados coletados. Essa análise mais abrangente possibilita ao pesquisador explorar os significados e relações entre os elementos analisados, além de possibilitar uma síntese das informações obtidas, facilitando a interpretação e a comunicação dos resultados. Outra vantagem da categorização é a verificação da consistência e validade dos dados obtidos. Ela ajuda a identificar contradições e inconsistências, permitindo que o pesquisador possa revisar e aperfeiçoar o processo de codificação (Cruzes e Dyba, 2011). O resultado da codificação e da categorização é apresentado no próximo capítulo.

RQ1: What productivity challenges are discussed?	Código
1) Desmotivação com o projeto devido a mudanças constantes nos requisitos.	1) Volatilidade dos requisitos

Figura 3.12: Exemplo de codificação da questão RQ1.

RQ4: What strategies and activities to improve productivity are discussed?	Código
1) Desenvolver com capacidade para mudanças 2) Prototipar e apresentar versões iniciais 3) Receber feedback, implementar as mudanças, mostrar o resultado e repetir o processo. 4) Padronizar o código 5) Modularizar o código 6) Adotar testes unitários	1) Reconfigurar dinâmica de trabalho 2) Reconfigurar dinâmica de trabalho 3) Reconfigurar dinâmica de trabalho 4) Reconfigurar dinâmica de trabalho 5) Reconfigurar dinâmica de trabalho 6) Reconfigurar dinâmica de trabalho

Figura 3.13: Exemplo de codificação da questão RQ4.

# Capítulo 4

## Resultados

Neste capítulo, são apresentados os resultados obtidos das análises de 57 discussões do fórum *Software Engineering* e 23 do fórum *Project Management*, utilizando a metodologia descrita no capítulo 3.

Os resultados das análises dos dois fóruns são apresentados separadamente para uma melhor compreensão. Na próxima seção, serão detalhados os achados do fórum *Software Engineering*, permitindo uma análise aprofundada dos temas discutidos pelos membros dessa comunidade. Em seguida, na seção subsequente, podem ser visualizados os dados obtidos do fórum *Project Management*, oferecendo uma visão mais clara sobre as discussões centradas na produtividade dos profissionais dessa área. Essa abordagem facilita a comparação entre os dois contextos, destacando as particularidades e convergências entre os tópicos abordados.

### 4.1 Resultado da análise do fórum Software Engineering

Os resultados obtidos do fórum *Software Engineering* para cada questão de pesquisa são apresentados nas seções a seguir.

#### 4.1.1 RQ1: Quais são os desafios de produtividade enfrentados pelos desenvolvedores de software?

Conforme mencionado no capítulo anterior, neste estudo utilizou-se um processo de síntese temática para analisar discussões sobre desafios de produtividade no fórum *Software Engineering* do Stack Exchange. Como resultado desse processo, identificamos 19 fatores distintos que contribuem para esses desafios, os quais foram agrupados em quatro categorias principais: Fatores Organizacionais, Fatores Interpessoais, Fatores Pessoais e Fatores Técnicos. É importante ressaltar que algumas discussões apresentaram mais de um fator.

Os fatores organizacionais englobam questões relacionadas à gestão de projetos, processos de desenvolvimento de software e ambiente de trabalho. Já os fatores

interpessoais se referem a questões relacionadas ao trabalho em equipe, comunicação e colaboração entre os membros da equipe. Os fatores pessoais abrangem questões relacionadas à motivação, satisfação e bem-estar dos desenvolvedores sem o envolvimento direto de terceiros. Por fim, os fatores técnicos envolvem questões relacionadas ao conhecimento e habilidades técnicas dos desenvolvedores, além das características do produto/projeto em desenvolvimento.

As categorias com seus respectivos fatores são apresentados na Figura 4.1.

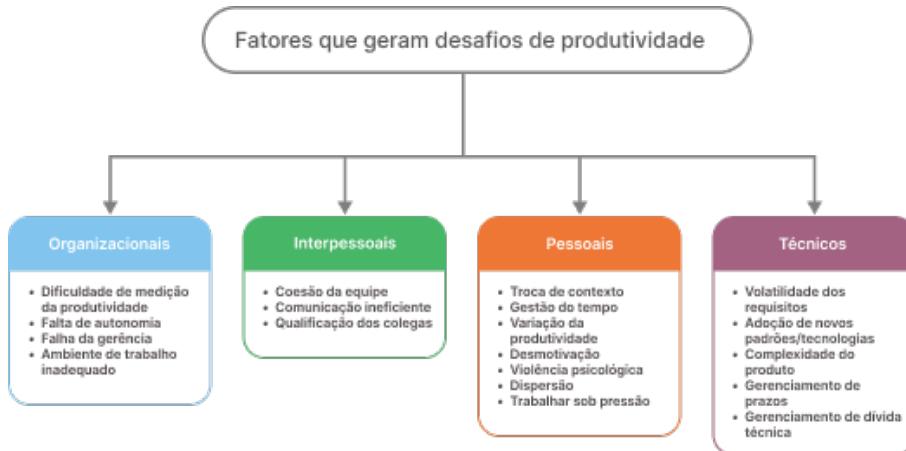


Figura 4.1: Fatores que geram os desafios de produtividade

Na Tabela 4.1, é possível visualizar a relação dos fatores que ocasionam os desafios encontrados ao responder à questão RQ1, utilizando as discussões do fórum Software Engineering. A tabela apresenta os fatores, o número de ocorrências de cada um e, ao final, o total geral de ocorrências. Os itens estão ordenados de forma decrescente, conforme o número de ocorrências, para facilitar a identificação dos fatores mais recorrentes.

### Descrição dos fatores por categoria

Nesta seção, detalhamos os fatores identificados durante o processo de codificação das discussões analisadas. Esses fatores foram agrupados em categorias, conforme explicado no capítulo de Metodologia. O processo de categorização ocorreu após a codificação, visando organizar os dados de forma a facilitar a visualização e compreensão das informações coletadas.

#### Fatores organizacionais

A seguir, estão listados os fatores que geram desafios organizacionais:

- Dificuldade de medição da produtividade: Este fator refere-se à dificuldade de medir a produtividade dos profissionais de maneira adequada e justa. Isso pode incluir a falta de critérios claros para avaliação ou o uso de ferramentas inadequadas. Por exemplo, em uma discussão, um autor mencionou: “*The*

Tabela 4.1: Relação dos fatores que geram os desafios de produtividade: Fórum SE.

Fator	Número de ocorrências
Gestão do tempo	14
Adoção de novos padrões/tecnologias	9
Dificuldade de medição da produtividade	7
Ambiente de trabalho inadequado	6
Troca de contexto	6
Falha da gerência	5
Variação da produtividade	5
Coesão da equipe	3
Dispersão	3
Complexidade do produto	2
Comunicação ineficiente	2
Falta de autonomia	2
Gerenciamento de prazos	2
Qualificação dos colegas	2
Desmotivação	2
Gerenciamento de dívida técnica	1
Trabalhar sob pressão	1
Violência psicológica	1
Volatilidade dos requisitos	1
<b>Total</b>	<b>74</b>

Tabela 4.2: Tabela de fatores e número de ocorrências.

*upper management at our company has laid out a goal for our software team to be 15% more productive over the next year. Measuring productivity in a software development environment is very subjective, but we are still required to come up with a set of metrics.*”. Nesse caso, o autor deve aumentar a produtividade de sua equipe em 15%, mas considera subjetivas as métricas de produtividade, mesmo sendo necessário realizar essa medição para cumprir as ordens da administração.

- Falta de autonomia: Este fator se refere à ausência de autonomia dos profissionais em relação ao seu trabalho, devido a restrições de gerenciamento, falta de liberdade para escolher tecnologias ou metodologias de desenvolvimento, ou outras circunstâncias que limitam a capacidade de tomar decisões. Por exemplo, em uma discussão, o autor mencionou: *“Does SOC 2 really require (in practice) that developers can't be trusted to choose on their own what programs they can run on their machines? What alternative strategies are there that poor developers can effectively leverage, in order to try and argue that they would really need to be in control of their development machines?”*. Nesse caso, o autor da questão expressa seu descontentamento com as novas normas que o impedem de instalar os programas desejados em sua máquina, sendo

necessário solicitar essa permissão a terceiros responsáveis.

- Falha da gerência: Este fator se refere a problemas relacionados à liderança da equipe ou empresa, incluindo distribuição ineficiente de tarefas, falta de planejamento, falta de organização e imposições arbitrárias. Por exemplo, um autor mencionou em uma discussão: *“Sadly, upper management has a tendency to set deadlines without consulting the programmers, which has led to many an all-nighter.”*. Nesse caso, o autor reclama da imposição de prazos sem consulta dos desenvolvedores, resultando na necessidade de trabalho extra à noite para cumpri-los.
- Ambiente de trabalho inadequado: Este fator se refere à necessidade de um ambiente de trabalho adequado para os profissionais, seja dentro ou fora das organizações, incluindo equipamentos e tecnologias apropriadas. Muitas das discussões que envolvem esse fator destacam a necessidade de lidar com interrupções no ambiente de trabalho. Por exemplo, um autor mencionou: *“I have a strange situation at work, where a colleague of mine often asks me and other co-workers for working code. I would like to help him, but this constant request of trivial snippets interrupts my thoughts and sometimes makes it hard to concentrate.”*. Nesse caso, o autor relata que as constantes interrupções de um colega dificultam sua concentração e, consequentemente, sua produtividade.

### **Fatores interpessoais**

A seguir, estão listados os fatores que geram desafios interpessoais:

- Coesão da equipe: Os desafios gerados pelo fator coesão da equipe referem-se à dificuldade de manter a equipe unida e alinhada em relação aos objetivos do projeto, integração de novos membros e falta de colaboração. Um exemplo desse problema foi relatado por um autor: *“In case of the first project I tried to convince the team to reorganize, I suggested to split (it was at first around 12 people, later became 18 and more even), I wanted to convince them that we must have mockups first and should reduce the number and/or length of the meetings. Unfortunately, all the requests and suggestions have been rejected.”*. Nesse caso, o autor tentou implementar mudanças para melhorar a coesão da equipe, mas suas sugestões foram rejeitadas, dificultando o alinhamento e a colaboração.
- Comunicação ineficiente: Os desafios de comunicação envolvem falta de comunicação clara, comunicação agressiva e comunicação ineficaz entre desenvolvedores e gerentes. Um exemplo desse problema foi relatado por um autor: *“The lead dev simply picks faults that the pep8 script doesn’t pick up and shouts at us in the next stand-up meeting.”*. Nesse caso, a comunicação agressiva do desenvolvedor líder, que aponta falhas e reage de forma negativa nas reuniões, contribui para um ambiente de trabalho estressante e improductivo.
- Qualificação dos colegas: Os desafios relacionados ao fator Qualificação dos colegas referem-se a lidar com outros profissionais no mesmo time que possuem

menos experiência ou conhecimento técnico, além de desafios relacionados à falta de atualização e aprimoramento das habilidades dos colegas. Um exemplo dessa situação foi descrito por um autor: *“How do you handle yourself in a new team where you are the senior most developer and most others in the team are junior to you by several years.”*. Neste contexto, o autor expressa a dificuldade de liderar uma equipe composta majoritariamente por membros menos experientes.

### Fatores pessoais

A seguir, estão listados os fatores que geram desafios pessoais:

- Troca de contexto: O fator troca de contexto está relacionado aos desafios de lidar com diferentes atividades dentro de um mesmo projeto, como desenvolver *back-end* e *front-end*, lidar com codificação e teste, ou trabalhar em projetos distintos simultaneamente. Essa mudança constante de contexto pode levar à queda na produtividade do desenvolvedor. Na maioria dos casos, essa queda não está relacionada diretamente à gestão do tempo, mas sim à atividade em si. Por exemplo, um desenvolvedor relatou: *“If I compare this to working on one small library or just some deep backend stuff, I find that my productivity is basically non-existent, because I constantly switch around and find bugs in 50 different places, instead of just focusing on one.”*. Nesse caso, o constante alternar entre tarefas diferentes impacta significativamente sua produtividade, demonstrando a dificuldade em manter o foco e a eficiência em múltiplas frentes de trabalho.
- Gestão do tempo: O fator Gestão do tempo está relacionado à dificuldade de gerenciar o próprio tempo de forma eficaz, levando em consideração prazos e prioridades. Nesses casos, não é a troca de contexto que impacta a produtividade do desenvolvedor, mas sim a divisão inadequada do tempo. Por exemplo, um desenvolvedor descreveu: *“At the moment I’m spending more time planning out a commit than actually writing code when adding a new feature. Less than two hours would be lucky, and sometimes I’d spend a good part of the day without writing any code.”*. Neste relato, o desenvolvedor gasta uma quantidade desproporcional de tempo planejando em comparação com a execução, o que pode levar a atrasos no projeto e à perda de eficiência.
- Variação da produtividade: Refere-se às oscilações na produtividade de um desenvolvedor ao longo do dia ou do projeto, que podem ser causadas por problemas pessoais, de saúde, desmotivação ou outros fatores externos. Por exemplo, um autor expressou: *“I think that personally, there is a huge variance in how much I can get done on a day by day basis, so I was wondering if anyone else feels the same way or has done any research.”*. Outro autor questionou a consistência da produtividade, destacando: *“My question is, is there any scientific study on the productivity of programmers? I can’t help but feel that doing 8 hours of quality programming work from Monday to Friday consistently is very hard. And if yes, is there any literature that investigates*

*this?*”. Ambos os comentários ilustram a percepção comum de que a produtividade pode variar significativamente, não apenas de pessoa para pessoa, mas também diariamente para o mesmo indivíduo.

- Desmotivação: Refere-se à falta de engajamento e interesse no trabalho que está sendo realizado, frequentemente causada por descontentamento com a empresa ou condições de trabalho. Por exemplo, um desenvolvedor compartilhou: *“The senior tech lead got a major promotion and a GIGANTIC bonus. I got nothing. I was so disappointed that I just stopped caring. Over the last year, I have just kind of floated. During my first 4 years I felt energized after a 10 hour day. Now I can barely be bothered to work 6 hours a day.”*. Este relato ilustra como o reconhecimento desigual e a falta de recompensas podem levar à desmotivação, afetando significativamente a produtividade e o entusiasmo pelo trabalho.
- Violência psicológica: Refere-se à exposição a comportamentos abusivos ou hostis por parte de colegas ou gerentes, que podem comprometer a saúde mental e o bem-estar do desenvolvedor. A violência psicológica inclui ações como humilhações, ameaças, insultos e a imposição de medo, frequentemente caracterizada por uma dinâmica de poder desequilibrada. Um exemplo disso foi relatado por um desenvolvedor: *“Personally I think coding standards are important but also think that we are wasting a lot of time obsessing with them, and when I verbalized this it provoked rage. I’m seen as a troublemaker in the team, a team that is looking for scapegoats for its failings. The lead dev simply picks faults that the pep8 script doesn’t pick up and shouts at us in the next stand-up meeting.”*. Esse depoimento mostra como expressar uma opinião divergente provocou reações agressivas, reforçando o papel do desenvolvedor como “culpado” e exemplificando a violência psicológica no ambiente de trabalho.
- Dispersão: Refere-se à dificuldade em manter o foco e a concentração durante o trabalho, o que pode ser provocado por interrupções ou demoras no processo de trabalho. Um desenvolvedor exemplificou esta questão dizendo: *“I have a very difficult time focusing on what I’m doing (programming-wise) when something (compilation, startup time, etc.) takes more than just a few seconds.”*. Este comentário destaca como atrasos técnicos, como tempos de compilação ou inicialização prolongados, podem interromper o fluxo de trabalho e dificultar a manutenção do foco.
- Trabalhar sob pressão: Refere-se à pressão para atingir metas e prazos, muitas vezes em condições estressantes e com alta demanda de trabalho. Este ambiente pode levar a decisões apressadas e a uma qualidade de trabalho reduzida. Um desenvolvedor descreveu sua experiência dizendo: *“When under a particularly strict programming deadline (like an hour), if I panic at all, my tendency is to jump into coding without a real plan and hope I figure it out as I go along.”*. Este relato ilustra como a pressão de prazos rigorosos pode induzir

o desenvolvedor a começar a codificar sem um plano definido, na esperança de resolver problemas à medida que surgem, o que pode comprometer a eficácia do processo de desenvolvimento.

### Fatores técnicos

A seguir, estão listados os fatores que geram desafios técnicos:

- Volatilidade dos requisitos: Refere-se à dificuldade de gerenciar mudanças frequentes nos requisitos do projeto durante sua execução, o que pode desestabilizar o planejamento e a implementação. Um desenvolvedor compartilhou sua experiência: *“They ask for one functionality, I ask a thousand questions so that I am clear on the requirements of the functionality. As I don’t ever like to miss a single scenario when collecting requirements, and I do my work with all my heart and brain in it. Later on the day of the demo, they would say: let’s change this to that and that to this.”*. Este exemplo ilustra como as alterações de última hora nos requisitos podem frustrar os esforços meticulosos para atender às expectativas iniciais, impondo desafios adicionais e estresse ao desenvolvedor.
- Adoção de novos padrões/tecnologias: Refere-se às dificuldades enfrentadas pelos desenvolvedores em manter a produtividade ao adotar novos padrões e tecnologias, como linguagens e ferramentas. Essa transição pode ser desafiadora, especialmente para aqueles menos familiarizados com as mudanças. Um desenvolvedor relatou: *“We’re now a few weeks into the project and I find that some of my co-workers (especially ones older of age) are complaining about not being able to find an implementation of an interface directly but needing to ‘search’ for the implementation, and needing more time to find the implementation of a particular markup on the front-page.”*. Este exemplo ilustra como a introdução de novas tecnologias pode exigir uma curva de aprendizado que afeta temporariamente a eficiência da equipe.
- Complexidade do produto: Refere-se ao desafio de manter um bom nível de produtividade ao lidar com o desenvolvimento de produtos complexos. A complexidade pode envolver diversas camadas de tecnologia e requisitos intrincados, o que frequentemente leva a problemas complicados que exigem soluções detalhadas. Um desenvolvedor descreveu sua experiência: *“I have been working on a solo project for 7 months now and the thing that is really killing my productivity is getting stuck on one little issue like an error I can’t figure out or trying to get a library to do something.”*. Este relato destaca como pequenos obstáculos técnicos em projetos complexos podem consumir uma quantidade desproporcional de tempo e esforço, afetando significativamente a produtividade.
- Gerenciamento de prazos: Refere-se aos desafios de manter um nível de produtividade suficiente para cumprir os prazos estabelecidos. Mesmo desenvolvedores competentes podem enfrentar dificuldades para entregar projetos dentro

do tempo previsto devido a várias razões, como subestimação de tempo, es-  
cupo de trabalho imprevisto, ou interrupções. Um exemplo foi compartilhado  
por um programador: *“I am an above average programmer (I take interest  
in the development of languages, frameworks, and technology in general and  
have a good grasp of systems), but I constantly find myself not being able to  
deliver things on time.”*. Este relato ilustra como, apesar da competência e do  
interesse em avanços tecnológicos, desafios no gerenciamento de prazos podem  
impedir a entrega oportuna de projetos.

- Gerenciamento de dívida técnica: Refere-se ao desafio de equilibrar a resolução de problemas técnicos acumulados e a implementação de novas funcionalidades. A dívida técnica pode acumular-se quando soluções provisórias ou inadequadas são aplicadas em favor de entregas rápidas, o que, eventualmente, reduz a eficiência do desenvolvimento. Um desenvolvedor expressou: *“I don’t expect zero tech debt. In this post, technical debt problem refers to severity that has been causing negative impact, say productivity.”*. Este comentário ressalta o entendimento de que alguma dívida técnica é inevitável, mas a preocupação reside na gravidade da dívida que começa a impactar negativamente aspectos como a produtividade.

#### 4.1.2 RQ2: Como os desafios de produtividade afetam os desenvolvedores de software?

Para responder à questão RQ2, coletamos informações sobre os aspectos humanos descritos nas discussões, incluindo os sentimentos e emoções dos profissionais em relação aos desafios de produtividade que enfrentam. É fundamental destacar que nem todas as perguntas que geraram discussões abordaram esses aspectos específicos.

Durante a análise, foram identificados 15 aspectos que descrevem como os desenvolvedores são impactados pelos desafios de produtividade. Esses aspectos foram classificados em três categorias: Problemas de Saúde Mental, Problemas de Desempenho e Problemas Físicos. A Figura 4.2 apresenta a relação desses impactos separados por categoria.

Na tabela 4.3, é possível visualizar a relação dos impactos gerados pelos desafios de produtividade encontrados ao responder a questão RQ2, bem como o número de ocorrências de cada um deles. Essa análise foi realizada com base em 57 discussões analisadas no fórum de *Software Engineering* e a descrição dos impactos é apresentada no tópico a seguir.

##### Descrição dos impactos por categorias

Nesta seção, serão apresentados os impactos que indicam como os profissionais de software membros deste fórum se sentem em relação aos problemas de produtividade.

###### Problemas de Saúde Mental

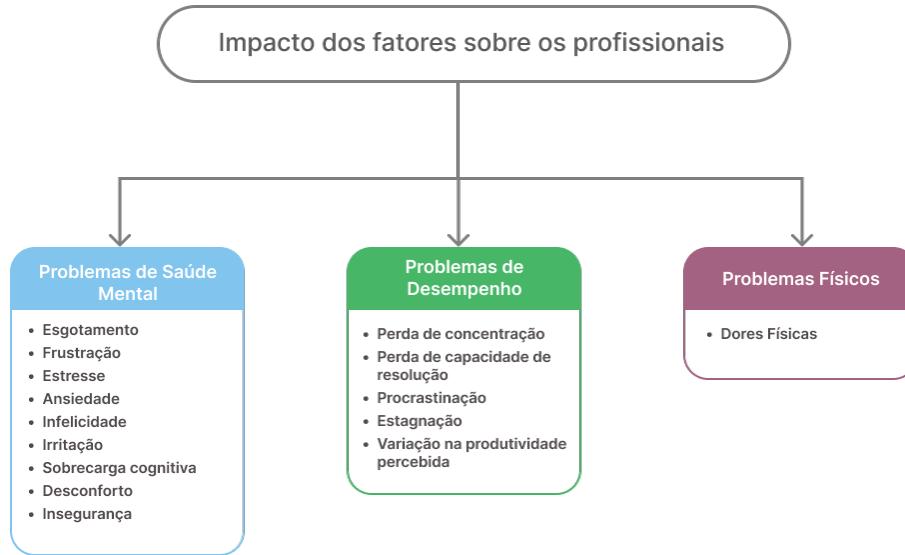


Figura 4.2: Impactos dos desafios de produtividade sobre os profissionais membros da comunidade *Software Engineering*.

Abaixo estão listados os problemas relacionados a saúde mental dos membros da comunidade:

- Esgotamento: Refere-se à sensação de cansaço mental excessivo, resultante do excesso de trabalho ou das pressões enfrentadas, muitas vezes impostas pela gerência. Este estado pode levar a uma redução significativa na produtividade e na motivação do indivíduo. Um exemplo dessa condição foi descrito por um profissional recentemente graduado: *"I feel like I have burned out, even though I am only out of college for 5 years. Now, because the flow is totally changed, it's not even the same function that I started with, there's delay in launch, there are bugs in production and marketing people are screaming about it. I always end up in some mess. and I am tired of it."*. Este relato ilustra como mudanças significativas nas responsabilidades, atrasos no lançamento, erros de produção e pressões externas podem contribuir para o esgotamento de um desenvolvedor, afetando sua capacidade de manter a eficácia no trabalho.
- Frustração: Manifesta-se como um sentimento de desapontamento, decepção e insatisfação diante das dificuldades e desafios enfrentados no trabalho. Um exemplo disso foi compartilhado por um profissional que expressou: *"I was so disappointed that I just stopped caring."*. Este comentário destaca como a frustração pode culminar em apatia e desinteresse pelo trabalho, afetando negativamente o bem-estar emocional e a motivação do indivíduo.
- Estresse: Sensação de pressão e tensão excessiva que pode emergir em ambientes de trabalho exigentes ou ao gerenciar múltiplas demandas simultaneamente. Este estado pode ser acompanhado por outros sentimentos, como frustração, sobrecarga e angústia. Um profissional relatou sua experiência: *"It can feel*

Tabela 4.3: Impactos dos desafios sobre os profissionais

Impacto	Número de ocorrências
Variação na produtividade percebida	12
Insegurança	9
Perda de concentração	9
Infelicidade	4
Esgotamento	4
Estagnação	4
Sobrecarga cognitiva	3
Frustação	3
Dores físicas	2
Perda de capacidade de resolução	2
Ansiedade	1
Desconforto	1
Estresse	1
Irritação	1
Procrastinação	1
<b>Total</b>	<b>57</b>

*like your time is pretty fragmented and interrupted moving between project to project. Sometimes frustrating, stressful & overwhelming.*”. Este comentário destaca como a constante transição entre diferentes projetos e as interrupções frequentes podem contribuir para um ambiente estressante, exacerbando a sensação de sobrecarga emocional e física.

- Insegurança: Caracteriza-se pela falta de confiança nas próprias habilidades, no sucesso do projeto ou na competência de outros membros da equipe, incluindo a gerência. Este sentimento pode prejudicar significativamente o desempenho e a produtividade. Um profissional compartilhou sua experiência, ilustrando este ponto: *“The nature of getting stuck on an issue and the only remedy to said problem has been slowly chipping away at my confidence, and I evidently see that I am slowing down in shipping out components.”*. Este relato destaca como problemas persistentes e a dificuldade em resolvê-los podem gradualmente erodir a confiança de um desenvolvedor, afetando sua eficiência e velocidade no trabalho.
- Sobrecarga cognitiva: Refere-se ao excesso de informações e tarefas a serem processadas simultaneamente, o que pode levar a dificuldades em organizar e priorizar atividades. Esse acúmulo frequentemente resulta em baixo desempenho devido à incapacidade de focar efetivamente em tarefas específicas. Um desenvolvedor descreveu sua luta com esse problema: *“How do you manage context switching overhead in software development when getting back to work over different parts of your project? I’m really struggling with the overheads of context switching.”*. Este comentário ilustra como a necessidade de alter-

nar frequentemente entre diferentes partes de um projeto pode intensificar a sobrecarga cognitiva, dificultando a manutenção de eficiência e a progressão eficaz no trabalho.

- Desconforto: Sentimento de desconforto emocional relacionado ao projeto ou à equipe. Essa sensação pode afetar negativamente a capacidade de tomar decisões eficazes. Um desenvolvedor expressou seu desconforto com a pressão de tempo ao questionar: *“I’m not always comfortable sitting there thinking while the clock ticks away. Is there a checklist or are there techniques to recognize when you understand the problem well enough to start coding? When is it most productive to think and design more vs. code some experiments and figure out the over-all design later?”*. Este comentário destaca como a ansiedade de ter que produzir resultados rapidamente pode criar um ambiente de trabalho desconfortável, dificultando o processo de decisão entre dedicar mais tempo ao planejamento ou avançar com a codificação experimental.
- Ansiedade: Sentimento de preocupação excessiva e tensão que pode ser acompanhado por sintomas físicos e emocionais negativos, afetando a capacidade de tomar decisões racionais e objetivas. Um desenvolvedor expressou sua ansiedade ao adaptar-se a novas metodologias: *“In your opinion, are there real benefits to this pattern I’m not seeing? Am I letting my anxiety over doing something new cloud my judgment, or are my concerns as well-founded as I believe they are?”*. Este comentário ilustra como a ansiedade relacionada a mudanças e à adoção de novas práticas pode turvar o julgamento e influenciar negativamente a percepção da eficácia das novas abordagens.
- Infelicidade: Sentimento de tristeza e insatisfação com o trabalho realizado, que pode surgir quando as expectativas pessoais não são atendidas ou quando o indivíduo se sente improdutivo por exemplo. Um desenvolvedor compartilhou seu descontentamento: *“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.”*. Este relato destaca como a falta de realização das atividades esperadas, como a codificação, pode levar à infelicidade no trabalho, afetando negativamente o bem-estar emocional e a satisfação profissional.
- Irritação: Sentimento de raiva e frustração que surge diante de dificuldades e desafios enfrentados no trabalho. Um desenvolvedor expressou sua irritação com responsabilidades fora de sua área de interesse: *“I am not a system administrator, I enjoy developing software. But this just annoys me. I do not want to inform myself on all the 157 Python packages and thousands of Debian packages I have on my system and know what their dependencies are.”*. Este comentário ilustra como ser forçado a lidar com tarefas não relacionadas à sua especialidade principal pode causar irritação e afetar negativamente a motivação e a satisfação no trabalho.

## Problemas de Desempenho

Abaixo estão listados os problemas que afetam o desempenho:

- Perda de concentração: Refere-se à dificuldade em manter o foco durante tarefas e atividades específicas, o que pode levar a um desempenho reduzido. Interrupções ou atrasos, mesmo que breves, podem significativamente distrair e diminuir a eficiência. Um desenvolvedor compartilhou sua experiência com esse desafio: *“I have a very difficult time focusing on what I’m doing (programming-wise) when something (compilation, startup time, etc.) takes more than just a few seconds.”*. Este relato destaca como pequenas interrupções no fluxo de trabalho, como tempos de compilação ou inicialização prolongados, podem afetar a capacidade de manter a concentração e, por consequência, impactar negativamente a produtividade.
- Perda de capacidade de resolução: Refere-se à dificuldade em encontrar soluções eficazes para os desafios que surgem, o que pode resultar em atrasos e necessidade de retrabalho. Essa dificuldade pode ser especialmente frustrante para profissionais que enfrentam expectativas altas ou pressões de tempo. Um programador descreveu seu dilema: *“I am an above average programmer (I take interest in the development of languages, frameworks and technology in general and have a good grasp of systems), but I constantly find myself not being able to deliver things on time.”*. Este comentário ilustra como, apesar da habilidade e interesse, o desafio de resolver problemas de maneira eficiente sob pressão pode impedir a entrega pontual de projetos, afetando a produtividade geral.
- Procrastinação: Refere-se ao adiamento constante das tarefas, muitas vezes devido à falta de motivação ou à desmotivação causada por trabalhos repetitivos. Esse comportamento pode resultar em atrasos significativos e na perda de prazos cruciais. Um desenvolvedor expressou sua luta pessoal: *“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.”*. Este relato destaca como a repetição e a falta de engajamento nas tarefas podem levar à procrastinação, comprometendo a eficiência e a capacidade de cumprir prazos estabelecidos.
- Estagnação: Caracteriza a falta de progresso ou avanço no trabalho, resultando em baixa motivação e desempenho reduzido. A ausência de desenvolvimento contínuo ou novos desafios pode levar a um sentimento de paralisação profissional. Um desenvolvedor compartilhou suas preocupações e buscou conselhos para lidar com essa situação: *“Any advice on keeping the productivity level high in these scenarios? How do I make sure that I don’t miss anything while changing the functions? And what do I do when I feel stuck.”*. Este comentário ilustra a dificuldade em manter a produtividade e a motivação quando confrontado com a sensação de estagnação, especialmente ao lidar com mudanças nas funções ou ao se sentir preso em tarefas repetitivas ou pouco desafiadoras.
- Variação na produtividade percebida: Sensação de altos e baixos na produtivi-

dade, onde períodos de grande eficiência são seguidos por momentos de baixa produtividade. Um desenvolvedor descreveu essa experiência: “*I feel like I do this on a lot of projects, and when I look at it for a couple hours, and don't solve my problem, I don't feel like I did anything, and feel as though it was a waste of time. How do I spend more time programming, get more stuff done, and feel more productive after every programming session?*”. Esse relato destaca a dificuldade em manter uma produtividade constante e a frustração que surge quando não se sente produtivo após uma sessão de trabalho.

### Problemas Físicos

Abaixo é apresentado o impacto físico encontrado nas discussões:

- Dores físicas: Refere-se a sintomas de dor na região da cabeça mencionados pelos desenvolvedores, sem especificar o tipo exato de dor de cabeça. Um desenvolvedor comentou sobre as dores resultantes de trabalhar com código complexo: “*In the end this led to a bunch of spaghetti code, untestable tasks (because of the complexity of third party libraries, patching the code didn't even work) and a bunch of headaches.*”. Esse comentário ilustra como a complexidade do código e os desafios técnicos podem causar desconforto físico, afetando o bem-estar do desenvolvedor.

#### 4.1.3 RQ3: Como os desafios de produtividade afetam o trabalho realizado pelos desenvolvedores de software?

Para responder a questão RQ3, foi necessário entender como os desafios de produtividade afetam as tarefas realizadas pelos desenvolvedores. Após a análise das discussões, foram identificados 10 problemas que, em seguida, foram agrupados em três categorias: Problemas Interpessoais, Problemas Técnicos e Problemas Operacionais.

A categoria Problemas Operacionais engloba desafios que afetam diretamente a eficiência e a qualidade das operações. Incluem atrasos na entrega, retrabalho, sobre-carga de trabalho, redução da performance e a não identificação de valor agregado. Estes problemas podem resultar em impactos negativos na produtividade, custos adicionais e comprometimento da qualidade dos produtos ou serviços entregues.

Os problemas técnicos abrangem dificuldades específicas relacionadas à execução e manutenção de tarefas técnicas. Isso inclui dificuldades de conclusão de tarefas, desafios na manutenção de sistemas ou produtos e a redução percebida na qualidade entregue.

Por fim, os problemas interpessoais referem-se a conflitos e dificuldades de comunicação entre os membros da equipe, como atrito com a equipe. Esses problemas podem afetar o clima organizacional e a colaboração entre os membros da equipe, resultando em atrasos no projeto e na qualidade do trabalho desenvolvido.

A Figura 4.3 apresenta a relação desses problemas separados por categoria.

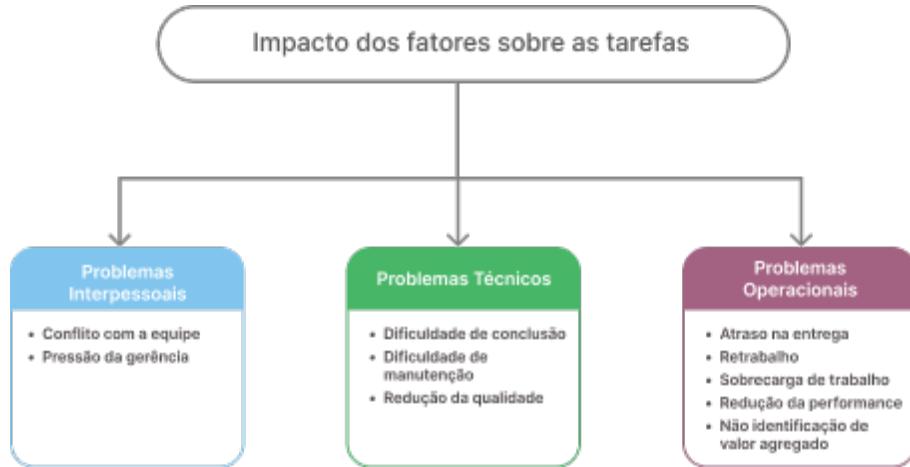


Figura 4.3: Impactos dos desafios de produtividade sobre as tarefas realizadas pelos membros da comunidade SE

Na Tabela 4.4, apresenta-se a relação dos problemas identificados ao responder à questão RQ3, juntamente com o número de ocorrências de cada um. Esta análise foi conduzida com base em 57 discussões do fórum de Engenharia de Software, e a descrição detalhada dos problemas é apresentada no próximo tópico.

Tabela 4.4: Número de ocorrências dos problemas gerados.

Problemas	Número de ocorrências
Redução da performance	33
Redução da qualidade	11
Atraso na entrega	8
Retrabalho	4
Sobrecarga de trabalho	3
Pressão da gerência	2
Conflito com a equipe	1
Dificuldade de concluir a tarefa	1
Dificuldade de manutenção	1
Não identificação do valor agregado	1
<b>Total</b>	<b>65</b>

### Descrição dos problemas por categoria

Nesta seção, serão apresentados, por categoria, os problemas que indicam como os desafios de produtividade afetam o trabalho realizado pelos desenvolvedores de software na comunidade Software Engineering.

#### Problemas Operacionais

Abaixo estão listados os problemas operacionais encontrados nas discussões:

- Atraso na entrega: Ocorre quando o desenvolvedor não consegue entregar as atividades dentro dos prazos estipulados. Um exemplo foi descrito por um profissional: *“The result is often that I don’t deliver. Now I may know at the back of my head that what I need to do, but I would appreciate your strategy on how to be the guy who ‘delivers’.”*. Esse comentário evidencia a luta para cumprir prazos e a busca por estratégias para melhorar a pontualidade na entrega.
- Retrabalho: Refere-se ao enfrentamento de situações onde há necessidade de realizar o mesmo trabalho múltiplas vezes dentro de um projeto, impactando negativamente na produtividade e na motivação do desenvolvedor. Um exemplo disso foi compartilhado por um desenvolvedor: *“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. I like this project, I really want to stay here for a bit longer and I want to give my best even if I am working on something for the 5th time.”*. Este relato destaca como o retrabalho contínuo pode levar à desmotivação e procrastinação, apesar do desenvolvedor ainda ter um forte interesse no projeto.
- Sobrecarga de trabalho: Ocorre quando os desenvolvedores estão sobrecarregados com muitas tarefas ou projetos ao mesmo tempo, resultando em falta de foco, queda de produtividade e possível exaustão. Um desenvolvedor comentou: *“These are big and difficult problems, and as the only resident FE dev it will fall on me to find solutions to them, which could halt my productivity.”*. Outro desenvolvedor acrescentou: *“Sadly, upper management has a tendency to set deadlines without consulting the programmers, which has led to many an all-nighter.”*. Esses exemplos ilustram como a sobrecarga de trabalho pode impactar negativamente a produtividade e o bem-estar dos desenvolvedores.
- Redução da performance: Problema caracterizado pela diminuição do desempenho geral das tarefas realizadas em um determinado período de tempo. Foi observado que os desenvolvedores frequentemente confundem os conceitos de produtividade, eficácia e eficiência. Por essa razão, optou-se pelo termo ‘performance’, que abrange todos esses aspectos. Um desenvolvedor relatou: *“The nature of my job is that I have to switch back and forth between projects every few weeks. I find that one of the biggest impediments to my productivity is the ramp-up time to getting all the relevant pieces of code ‘back in my head’ again after not seeing it for a period.”*. Esse comentário destaca como a mudança frequente entre projetos pode reduzir a performance devido ao tempo necessário para relembrar detalhes importantes do código.
- Não identificação do valor agregado: Este problema ocorre quando o desenvolvedor não consegue identificar ou entender o valor que sua tarefa ou projeto irá agregar ao produto final ou aos objetivos de negócios mais amplos. Um exemplo disso foi descrito por um desenvolvedor: *“In the end this led to a*

*bunch of spaghetti code, untestable tasks (because of the complexity of third party libraries, patching the code didn't even work) and a bunch of headaches because nobody even knew what the added value of these services were.*”. Este comentário ilustra como a falta de clareza sobre o valor agregado pode resultar em trabalho desorganizado e desmotivação.

### Problemas técnicos

Abaixo estão listados os problemas técnicos encontrados nas discussões:

- Dificuldade de conclusão (das tarefas): Refere-se à incapacidade de finalizar tarefas devido a vários motivos, como falta de recursos adequados, falta de conhecimento técnico, complexidade da tarefa ou falta de foco. Um desenvolvedor comentou: “*But one of my problems is that when I am executing a task, I tend to dwell on specific problems I am facing with the task rather than focusing on the completion of the task.*”. Este relato ilustra como a tendência de focar em problemas específicos pode impedir a conclusão eficiente das tarefas.
- Dificuldade de manutenção: Este problema ocorre quando o software desenvolvido é difícil de ser modificado ou atualizado após sua implementação inicial. Um desenvolvedor expressou sua frustração: “*Every time I have to debug a thousand lines of a badly written method with variables reused all over, I'm totally lost. Some modifications or refactoring I've done introduced bugs in other places of the application.*”. Esse exemplo destaca como código mal escrito e a reutilização inadequada de variáveis podem complicar a manutenção e introduzir novos bugs.
- Redução da qualidade: Problema caracterizado pela produção de código-fonte ineficiente ou por produtos que não atendem às necessidades do cliente. Um desenvolvedor descreveu sua experiência: “*At any given time I handle around 3-4 projects, some having timelines that overlap with each other. Expectedly, code quality suffers. I also do not have formal testing; it usually goes down to walking through the system until it somewhat breaks. As a result, a considerable amount of bugs escape to production, which I have to fix and in turn sets back my other projects.*”. Este comentário mostra como a sobrecarga de projetos e a falta de testes formais podem levar à produção de código de baixa qualidade e à introdução de bugs em produção, impactando negativamente outros projetos.

### Problemas interpessoais

Abaixo estão listados os problemas interpessoais encontrados nas discussões:

- Conflito com a equipe: Este problema surge quando membros da equipe têm visões conflitantes ou problemas de comunicação, afetando a harmonia e a produtividade do grupo. Um exemplo disso foi relatado por um profissional: “*Now, because the flow is totally changed, it's not even the same function that I started with, there's delay in launch, there are bugs in production and marketing people are screaming about it.*”. Este comentário ilustra como mudanças

no fluxo de trabalho e a falta de alinhamento podem gerar conflitos e complicações dentro da equipe.

- Pressão da gerência: Ocorre quando há exigências por parte da gerência por resultados rápidos, o que pode impactar negativamente a qualidade do produto final e o bem-estar da equipe. Um desenvolvedor compartilhou sua experiência: *“Since the introduction of the coding standards, the team’s productivity has measurably plummeted, however this only reinforces the obsession, i.e. the lead dev simply blames our non-adherence to standards for the lack of progress.”*. Este relato destaca como a imposição de normas rigorosas e a pressão por resultados podem reduzir a produtividade e aumentar a tensão dentro da equipe, prejudicando a qualidade do trabalho realizado.

#### **4.1.4 RQ4: Que estratégias e atividades para melhorar a produtividade são discutidas?**

Para responder à questão RQ4, foi necessário analisar as respostas dos membros da comunidade e identificar as estratégias sugeridas para enfrentar os desafios de produtividade mencionados pelos autores das perguntas que originaram as discussões.

As estratégias identificadas tanto no fórum Software Engineering quanto no fórum Project Management foram categorizadas em Desenvolvimento e bem-estar pessoal, Gestão e organização do trabalho, e Comunicação e colaboração.

A categoria Desenvolvimento e Bem-Estar Pessoal engloba estratégias que visam o aprimoramento individual e o bem-estar dos profissionais. As ações incluem cuidar da saúde física e mental, investir em aprimoramento profissional, procurar um novo emprego ou projeto, redefinir a conduta pessoal, adaptar a perspectiva diante dos desafios e desvincular-se da responsabilidade excessiva. Essas estratégias são voltadas para o fortalecimento e a resiliência individual, promovendo a capacidade de lidar com os desafios de produtividade de maneira saudável e proativa.

Em contrapartida, as estratégias da categoria Gestão e Organização do Trabalho estão focadas na otimização e estruturação do ambiente de trabalho e dos processos de gestão. Elas incluem aprimorar a gestão do tempo, estruturar um ambiente de trabalho favorável, reconfigurar a dinâmica de trabalho, reconfigurar a equipe, recorrer a ferramentas tecnológicas, avaliar as causas dos problemas e alterar o método de medição da produtividade. Essas ações têm como objetivo melhorar a eficiência e a eficácia das operações diárias, garantindo que os processos de trabalho sejam mais organizados e produtivos.

Por fim, a categoria Comunicação e Colaboração abrange estratégias que se concentram em melhorar a interação e a cooperação entre os membros da equipe. As ações incluem convencer através de dados, convencer através de diálogo e requisitar apoio. Essas estratégias são fundamentais para construir um ambiente de trabalho colaborativo e comunicativo, onde as ideias e preocupações podem ser compartilhadas.

das aberta e efetivamente, contribuindo para uma maior coesão e alinhamento entre os membros da equipe.

As categorias e suas respectivas estratégias são apresentadas na Figura 4.4.

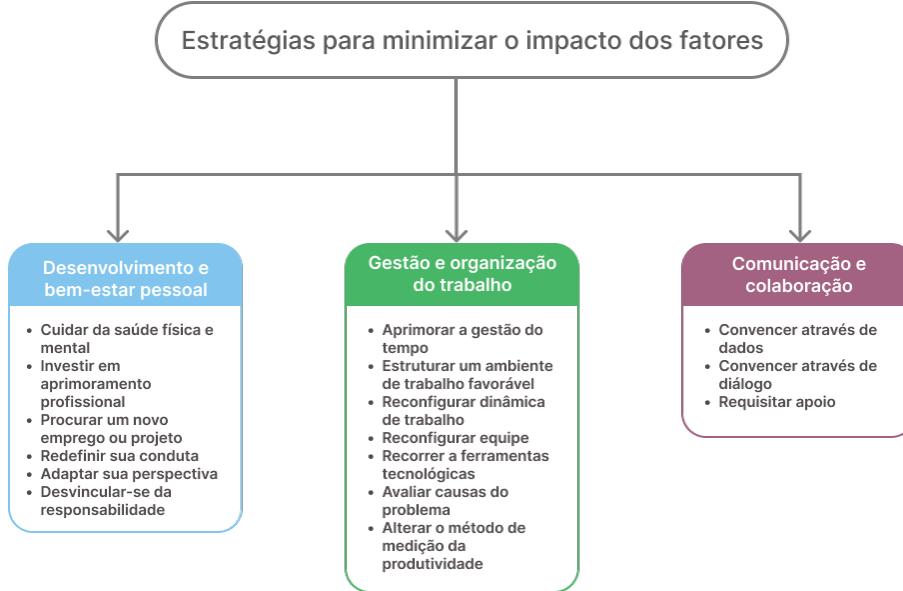


Figura 4.4: Estratégias para minimizar o impacto dos fatores.

Na Tabela 4.5, é possível visualizar a relação das estratégias utilizadas para contornar os problemas gerados pelos fatores identificados ao responder a questão RQ1, bem como o número de ocorrências de cada uma delas.

### Descrição das estratégias por categoria

Nesta seção, serão detalhadas as estratégias identificadas durante o processo de análise das discussões do fórum Software Engineering.

#### Desenvolvimento e bem-estar pessoal

A seguir, são listadas as estratégias relacionadas a desenvolvimento e bem-estar pessoal:

- Cuidar da saúde física e mental: Estratégias sugeridas para manter e melhorar a saúde física e mental, reconhecendo que um estado de bem-estar é essencial para a produtividade. Um exemplo dessa sugestão é: *"In the US they give the same advice before standardized tests 'get plenty of sleep, and have a good breakfast'."* A recomendação de um membro da comunidade ilustra a importância de hábitos saudáveis para a manutenção da produtividade.
- Investir em aprimoramento profissional: Estratégias que envolvem buscar oportunidades de desenvolvimento profissional, como cursos e treinamentos, para aumentar as habilidades e conhecimentos necessários para o trabalho.

Tabela 4.5: Estratégias sugeridas pelos membros da comunidade para melhorar a produtividade.

Estratégia	Número de ocorrências
Reconfigurar dinâmica de trabalho	189
Convencer através de diálogo	45
Aprimorar a gestão do tempo	27
Recorrer a ferramentas tecnológicas	29
Adaptar sua perspectiva	22
Redefinir sua conduta	25
Alterar o método de medição da produtividade	8
Avaliar causas do problema	10
Cuidar da saúde física e mental	12
Investir em aprimoramento profissional	9
Estruturar um ambiente de trabalho favorável	6
Procurar um novo emprego ou projeto	6
Requisitar apoio	6
Desvincular-se da responsabilidade	7
Convencer através de dados	7
Reconfigurar equipe	5
<b>Total</b>	<b>413</b>

Um exemplo sugerido por um membro da comunidade é: *“Maybe investing in some advanced training for some team members.”*. A sugestão destaca a importância de investir em capacitação contínua para melhorar a eficiência e a qualidade do trabalho.

- Procurar um novo emprego ou projeto: A recomendação desta estratégia sugere a mudança de emprego ou projeto como uma forma de encontrar um ambiente mais alinhado com as expectativas e necessidades pessoais e profissionais. Um exemplo fornecido por um membro da comunidade é: *“Why would you think you need to stay where you are? Maybe this was just your first experience, and opportunity to make your teeth on something. Now, you can go out and reach for more.”*. A recomendação sugere que buscar novas oportunidades pode ser benéfico para o crescimento profissional e pessoal.
- Redefinir sua conduta: Estratégias que referem-se à auto-reflexão e ajuste de comportamentos pessoais para se alinhar melhor com os objetivos profissionais e a dinâmica do ambiente de trabalho. Uma sugestão de um membro da comunidade é: *“Give constructive criticism of their ideas, but once the decision is made don’t continually whine about it.”*. A recomendação indica a importância de adaptar comportamentos para contribuir positivamente para o ambiente de trabalho.
- Adaptar sua perspectiva: Recomendações que envolvem a mudança de atitude

e mentalidade em relação aos desafios. Na maioria dos casos, esse tipo de estratégia sugere uma nova perspectiva sobre o problema, indicando que o autor da pergunta pode estar visualizando a situação de maneira equivocada. Uma sugestão fornecida por um membro da comunidade é: *“Consider the possibility that the problem that you are describing doesn’t really exist. If the company is profitable, and they are adequately compensating the developers, all-nighters boil down to simple planning problems and inadequate management of expectations.”*. A recomendação demonstra como uma mudança na percepção pode ajudar a lidar melhor com os desafios.

- Desvincular-se da responsabilidade: Este tipo de estratégia sugere que o autor da pergunta deve considerar que não é responsável pela resolução do problema, pois este pode estar além de sua capacidade de intervenção direta. Uma sugestão de um membro da comunidade é: *“Do your best, but don’t assume responsibility for their poor decisions.”*. A recomendação enfatiza a importância de reconhecer os limites da própria responsabilidade e evitar a sobrecarga desnecessária.

### **Gestão e organização do trabalho**

A seguir, são listadas as estratégias relacionadas à gestão e organização do trabalho:

- Aprimorar a gestão do tempo: Estratégias para melhorar a eficiência no uso do tempo, através de técnicas como priorização, planejamento e uso de ferramentas de gerenciamento de tarefas. Uma sugestão de um membro da comunidade é: *“Set a hard time limit on how long you think about something before writing code.”*. Esta recomendação enfatiza a importância de definir limites de tempo para o planejamento antes de iniciar a codificação.
- Estruturar um ambiente de trabalho favorável: Envolvem criar um ambiente físico e organizacional favoráveis à produtividade. Um membro da comunidade sugeriu: *“Also, you may already have heard and tried this, but I have a friend with a minor spatial disability, and usually it helps for him, if working on computers, to be in a darker room, to avoid having too many complex views and windows open (to avoid distraction).”*. Esta sugestão ilustra como ajustar o ambiente de trabalho pode ajudar a minimizar distrações e melhorar a concentração.
- Reconfigurar dinâmica de trabalho: Referem-se a ajustar os métodos e processos de trabalho para melhorar a eficiência e a eficácia, como a implementação de novas práticas de gerenciamento de projetos. Um exemplo sugerido por um membro da comunidade é: *“If you don’t want to context switch you should hand GUI development to someone else.”*. Esta recomendação sugere delegar tarefas específicas para evitar trocas de contexto frequentes.
- Reconfigurar equipe: Incluem a reorganização da equipe para otimizar o desempenho, seja através de mudanças na composição, nos papéis ou nas responsabilidades dos membros. Um membro da comunidade sugeriu: *“My idea for*

*now is to divide the project in two more or less independent sub-projects and let the new team members handle one of them.*”. Esta recomendação destaca a importância de dividir projetos e delegar responsabilidades para melhorar a eficiência.

- Recorrer a ferramentas tecnológicas: Esta estratégia sugere o uso de uma ferramenta, como um software, para auxiliar na resolução do problema que afeta a produtividade. Um membro da comunidade recomendou: “*I keep all the relevant apps open and then suspend the virtual machine. So, when I need to switch back to the project, I can start the VM up and it will present me with all of the information already on the screen.*”. Esta sugestão ilustra como o uso de tecnologias pode facilitar a execução de tarefas.
- Avaliar causas do problema: Envolve a análise detalhada dos problemas enfrentados para identificar suas causas raízes e, em seguida, realizar intervenções para solucionar o problema de produtividade. Um membro da comunidade sugeriu: “*You would be better off hiring no one new and looking at your processes to see where you can make changes to make things go faster.*”. Esta recomendação destaca a importância de fazer uma análise mais aprofundada antes de tomar uma decisão como expandir a equipe.
- Alterar o método de medição da produtividade: Esta estratégia está relacionada à sugestão de alterar a forma como a produtividade é medida no ambiente de trabalho do autor da pergunta. Um exemplo sugerido por um membro da comunidade é: “*Instead of using an error-prone metric, the senior architect could rely on a group of metrics which do measure productivity when used together: the number of features developed per month, the number of bugs introduced in the code base and the time spent solving those bugs, the evolution of the technical debt, etc.*”. Esta recomendação sugere o uso de múltiplas métricas para uma avaliação mais precisa da produtividade.

### Comunicação e colaboração

A seguir, são listadas as estratégias relacionadas a comunicação e colaboração:

- Convencer através de dados: Este tipo de estratégia envolve a utilização de dados e evidências concretas para obter apoio para ideias e mudanças propostas. Um membro da comunidade sugeriu: “*So come up with your best estimates for how much time it will take to develop your way, and how much the lead developer's way.*”. Esta recomendação enfatiza a importância de apresentar estimativas baseadas em dados para apoiar uma abordagem específica.
- Buscar diálogo: Envolve a utilização de habilidades de comunicação inter-pessoal para negociar e resolver conflitos, sem ser necessário o uso de dados para provar o ponto de vista. Um membro da comunidade sugeriu: “*So, go ahead and explain the problem to a non-developer colleague anyway and you'll likely obtain most of the benefits you would've otherwise obtained by talking*

*to another developer.*.” Esta sugestão destaca a importância de dialogar com outros profissionais para encontrar a solução dos seus problemas.

- Requisitar apoio: Este tipo de estratégia está relacionado a buscar suporte de colegas, superiores ou outras partes interessadas para enfrentar desafios e implementar soluções de forma colaborativa. Um membro da comunidade recomendou: “*Your best choice in this case is to ask for help, and publicly complain about deficiencies in the library.*”. Esta recomendação enfatiza a importância de solicitar ajuda e de comunicar abertamente as dificuldades enfrentadas para obter o suporte necessário.

## 4.2 Fórum da comunidade Project Management

Os resultados obtidos do fórum Project Management para cada questão de pesquisa são apresentados nas seções a seguir.

### 4.2.1 RQ1: Quais são os desafios de produtividade enfrentados pelos desenvolvedores de software?

O processo de análise do fórum Project Management foi semelhante ao do fórum Software Engineering. Como resultado dessa análise, foram identificados 13 fatores distintos que geram desafios de produtividade. Destes, 11 fatores já haviam sido encontrados na análise do fórum SE, e surgiram dois novos fatores: Cultura ruim e Inexperiência. Os fatores identificados no fórum Project Management foram agrupados nas mesmas categorias utilizadas para os fatores do fórum de Software Engineering: Fatores Organizacionais, Fatores Interpessoais, Fatores Pessoais e Fatores Técnicos.

As categorias com seus respectivos fatores são apresentados na Figura 4.5.

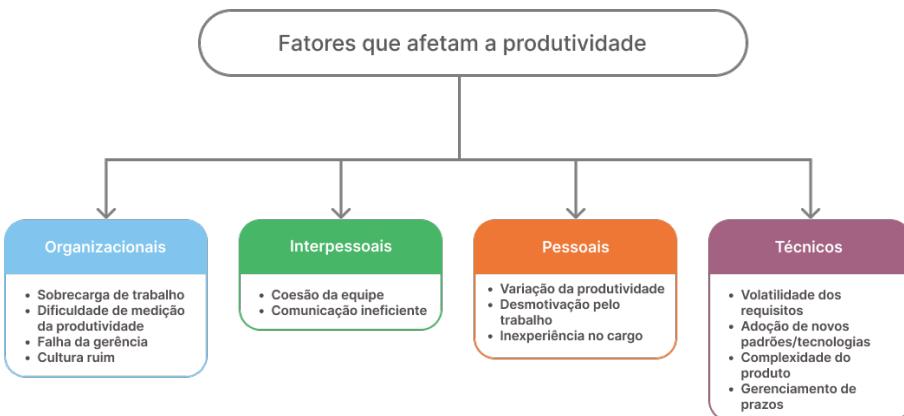


Figura 4.5: Fatores que geram os desafios de produtividade no fórum PM.

Na Tabela 4.6, é possível visualizar a relação dos fatores que ocasionam os desafios encontrados ao responder a questão RQ1, utilizando as discussões do fórum Project Management, bem como o número de ocorrências de cada um deles.

Tabela 4.6: Relação dos fatores que geram os desafios de produtividade.

<b>Fator</b>	<b>Número de ocorrências</b>
Dificuldade de medição da produtividade	11
Comunicação ineficiente	5
Desmotivação	3
Falha da gerência	3
Adoção de novos padrões/tecnologias	2
Complexidade do produto	2
Coesão da equipe	1
Cultura ruim	1
Gerenciamento de prazos	1
Inexperiência	1
Gestão do tempo	1
Variação da produtividade percebida	1
Volatilidade dos requisitos	1
<b>Total</b>	<b>33</b>

### Descrição dos fatores por categoria

Nesta seção, serão detalhados os novos fatores que foram identificados durante o processo de codificação das discussões do fórum Project Management.

#### Fatores organizacionais

A seguir, é listado o novo fator que gera desafios organizacionais:

- Cultura ruim: Refere-se a lidar com uma cultura organizacional na qual as práticas não são favoráveis para o desempenho eficiente. Um membro da comunidade compartilhou: *“I work in a company that has a bad culture: everybody has been treated the same: The same salary, no rewards for high achievers, and no punishment for bad employees, including people who have a part-time job during work hours while not doing anything.”*. Este exemplo ilustra como uma cultura organizacional negativa, que não reconhece ou recompensa os esforços dos funcionários, pode impactar a motivação e a produtividade.

#### Fatores pessoais

A seguir, é listado o novo fator que gera desafios pessoais:

- Inexperiência: Indica uma falta de habilidades ou conhecimento adequados para realizar as tarefas necessárias de forma eficaz e eficiente. Um membro da comunidade expressou: *“Most metrics seem difficult to define and measure*

*and I have no experience with it.*”. Este comentário destaca como a inexperience pode dificultar a implementação e a avaliação eficazes de métricas de desempenho, afetando a produtividade.

#### 4.2.2 RQ2: Como os desafios de produtividade afetam os desenvolvedores de software?

Durante a análise do fórum Project Management, foram identificados seis aspectos que descrevem o impacto dos desafios de produtividade nos desenvolvedores.

Diferentemente do fórum SE, no fórum Project Management, os problemas foram encontrados apenas nas categorias de Problemas de Saúde Mental e Problemas de Desempenho. A Figura 4.6 apresenta a relação desses impactos.

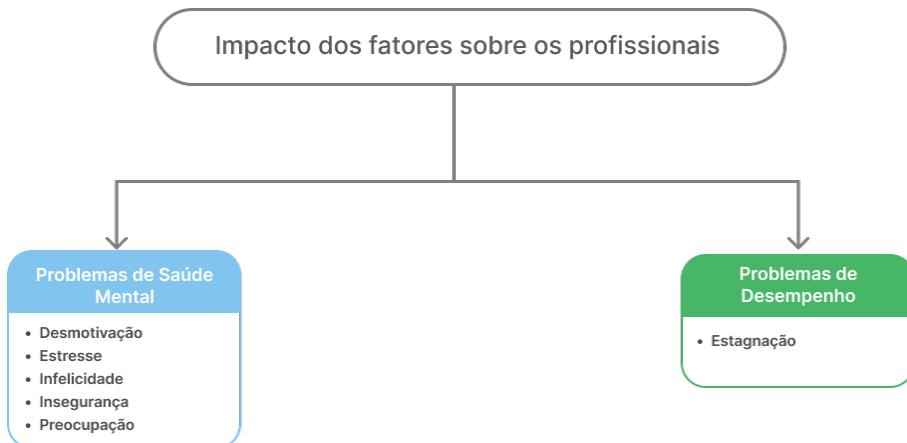


Figura 4.6: Impactos dos desafios de produtividade sobre os profissionais membros da comunidade *Software Engineering*.

Na Tabela 4.7, é possível visualizar a relação dos impactos gerados pelos desafios de produtividade encontrados ao responder a questão RQ2, bem como o número de ocorrências de cada um deles.

Tabela 4.7: Impactos dos desafios sobre os profissionais.

Fator	Número de ocorrências
Insegurança	3
Desmotivação	1
Estagnação	1
Estresse	1
Infelicidade	1
Preocupação	1
<b>Total</b>	<b>8</b>

### Descrição dos impactos por categorias

Nesta seção, serão apresentados os novos tipos de impactos que indicam como os profissionais de software membros deste fórum se sentem em relação aos problemas de produtividade.

#### Problemas de Saúde Mental

Abaixo é listado o novo problema relacionado a saúde mental dos membros da comunidade:

- Preocupação: Sentimento de apreensão e incerteza em relação ao desfecho de uma situação específica, que pode afetar o desempenho no trabalho. Um membro da comunidade expressou: *“I am concerned that my slow developer might be coasting it, from knowing that the quick developer will pick up some of the slack for us to meet the sprint goal.”*. Este comentário ilustra como a preocupação com o desempenho desigual entre os membros da equipe pode gerar apreensão e afetar a produtividade.

#### 4.2.3 RQ3: Como os desafios de produtividade afetam o trabalho realizado pelos desenvolvedores de software?

Após a análise das discussões, foram identificados 7 problemas. Destes 7, 6 já haviam sido identificados no fórum SE. Estes problemas foram agrupados utilizando as categorias: Problemas Interpessoais, Problemas Técnicos e Problemas Operacionais.

A Figura 4.7 apresenta a relação desses problemas separados por categoria.

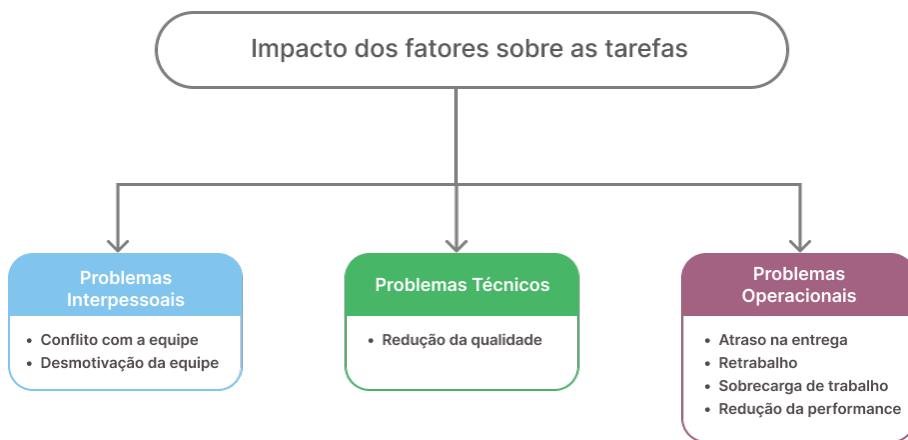


Figura 4.7: Impactos dos desafios de produtividade sobre as tarefas realizadas pelos membros da comunidade PM

Na Tabela 4.8, apresenta-se a relação dos problemas identificados ao responder à questão RQ3, juntamente com o número de ocorrências de cada um.

Tabela 4.8: Número de ocorrências dos problemas gerados.

Fator	Número de ocorrências
Redução da performance	5
Atraso na entrega	3
Conflito com a equipe	2
Redução da qualidade	2
Desmotivação da equipe	1
Retrabalho	1
Sobrecarga de trabalho	1
<b>Total</b>	<b>15</b>

### Descrição dos problemas por categoria

Nesta seção, será apresentado o novo problema que indica como os desafios de produtividade afetam o trabalho realizado pelos profissionais da comunidade PM.

#### Problemas interpessoais

Abaixo estão listados os problemas interpessoais encontrados nas discussões:

- Desmotivação da equipe: Refere-se à falta de entusiasmo, interesse ou comprometimento dos membros da equipe com as tarefas ou objetivos do projeto, o que pode resultar em baixa produtividade e desempenho insatisfatório. Um membro da comunidade compartilhou: *“Developer passion has gone. Nobody wants to spend so much time in meetings and all the decision making power has evaporated leaving the devs demotivated.”*. Este exemplo destaca como a perda de paixão pelo trabalho e a falta de poder decisório podem desmotivar a equipe, prejudicando a produtividade e o desempenho geral.

#### 4.2.4 RQ4: Quais estratégias e atividades para melhorar a produtividade são discutidas?

As estratégias identificadas no fórum Project Management foram agrupadas nas mesmas categorias utilizadas para as estratégias do fórum de Software Engineering: Desenvolvimento e bem-estar pessoal, Gestão e organização do trabalho e Comunicação e colaboração.

As categorias com suas respectivas estratégias são apresentadas na Figura 4.8.

Na Tabela 4.9, é possível visualizar a relação das estratégias utilizadas para contornar os problemas gerados pelos fatores encontrados ao responder a questão RQ1, bem como o número de ocorrências de cada uma delas.

Todos os tipos de estratégias recomendadas pelos membros do fórum Project Management nas discussões analisadas foram igualmente identificados no fórum Software Engineering, não havendo a identificação de nenhum novo tipo de estratégia.

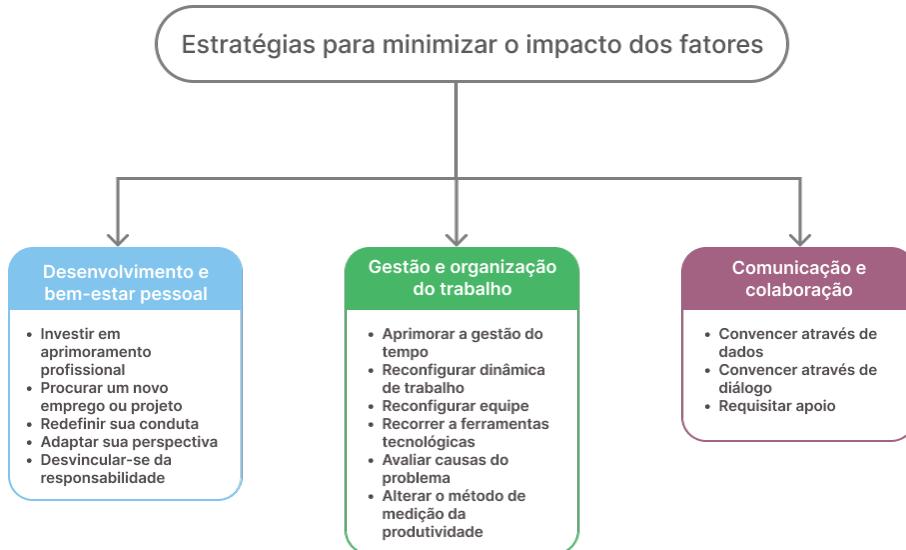


Figura 4.8: Estratégias para minimizar o impacto dos fatores que afetam a produtividade dos profissionais.

### 4.3 Comparação dos fóruns Software Engineering e Project Management

Nesta seção, é apresentada uma comparação entre os dados encontrados nos fóruns Software Engineering e Project Management. Para cada questão de pesquisa, serão ressaltadas as diferenças observadas entre os dois fóruns. É importante considerar que a quantidade de discussões analisadas no fórum SE foi mais que o dobro das analisadas no fórum PM, com 68 discussões no SE e 33 no PM. Essa disparidade no volume de dados certamente influencia os resultados e a comparação entre os fóruns.

#### 4.3.1 RQ1: Quais são os desafios de produtividade enfrentados pelos desenvolvedores de software?

Os fatores que geram desafios de produtividade encontrados no fórum SE, mas não mencionados no fórum PM, incluem:

- Ambiente de trabalho inadequado;
- Dispersão;
- Falta de autonomia;
- Gerenciamento de dívida técnica;
- Gestão do tempo;
- Qualificação dos colegas;
- Trabalhar sob pressão;

Tabela 4.9: Tipos de estratégias recomendadas pelos membros do fórum Project Management.

<b>Tipo de Estratégia</b>	<b>Número de ocorrências</b>
Alterar o método de medição da produtividade	13
Reconfigurar dinâmica de trabalho	12
Avaliar causas do problema	7
Convencer através de dados	6
Adaptar sua perspectiva	5
Aprimorar a gestão do tempo	5
Convencer através de diálogo	1
Desvincular-se da responsabilidade	1
Investir em aprimoramento profissional	1
Procurar um novo emprego ou projeto	1
Reconfigurar equipe	4
Recorrer a ferramentas tecnológicas	3
Redefinir sua conduta	1
Requisitar apoio	2
<b>Total</b>	<b>62</b>

- Troca de contexto;
- Violência psicológica.

Por outro lado, no fórum PM surgiram problemas novos que não foram mencionados no fórum SE:

- Cultura ruim;
- Inexperiência.

#### **4.3.2 RQ2: Como os desafios de produtividade afetam os desenvolvedores de software?**

Os impactos dos desafios de produtividade também variaram entre os dois fóruns. No fórum SE, foram mencionados os seguintes efeitos exclusivos:

- Ansiedade;
- Desconforto;
- Dores físicas;
- Esgotamento;
- Frustração;
- Irritação;

- Perda de concentração;
- Perda de capacidade de resolução;
- Procrastinação;
- Sobrecarga cognitiva;
- Variação da produtividade percebida.

No fórum PM, apenas a “Preocupação” foi mencionada como um impacto exclusivo.

#### **4.3.3 RQ3: Como os desafios de produtividade afetam o trabalho realizado pelos desenvolvedores de software?**

Os efeitos dos desafios de produtividade no trabalho também mostraram diferenças entre os dois fóruns. No fórum SE, os impactos exclusivos incluem:

- Dificuldade de conclusão;
- Dificuldade de manutenção;
- Não identificação de valor agregado;
- Pressão da gerência.

No fórum PM, a “Desmotivação da equipe” foi um impacto exclusivo.

#### **4.3.4 RQ4: Quais estratégias os desenvolvedores utilizam para superar os desafios de produtividade?**

As estratégias sugeridas para superar os desafios de produtividade foram bastante semelhantes entre os dois fóruns, indicando que as abordagens para lidar com problemas profissionais muitas vezes se aplicam a diferentes áreas e cargos. No entanto, no fórum SE, foram mencionadas duas estratégias exclusivas:

- Cuidar da saúde física e mental;
- Estruturar um ambiente de trabalho favorável.

Uma análise mais aprofundada e uma discussão detalhada desses resultados serão apresentadas no próximo capítulo.

# Capítulo 5

## Discussão

Neste capítulo, serão discutidos os resultados obtidos a partir das quatro questões de pesquisa deste estudo. A análise abordará não apenas as descobertas específicas de cada questão, mas também como esses resultados interagem e contribuem para uma compreensão mais ampla dos desafios de produtividade enfrentados por desenvolvedores de software.

Após a apresentação dos resultados, será introduzido um mapa conceitual que sintetiza e relaciona os achados obtidos. Este mapa servirá como uma ferramenta visual para elucidar as conexões entre os diferentes aspectos da produtividade do desenvolvedor, facilitando uma visão integrada dos principais temas estudados.

Em seguida, será realizada uma comparação entre os resultados desta pesquisa, que utilizou predominantemente literatura cinza, e aqueles derivados de estudos baseados em literatura branca. Esta análise comparativa tem o objetivo de destacar tanto as convergências quanto as divergências nas evidências obtidas de fontes acadêmicas e práticas.

Posteriormente, serão discutidas as implicações dos resultados obtidos para diferentes grupos de interesse, como pesquisadores, praticantes e educadores. Esta seção visa oferecer *insights* práticos e teóricos que possam ser aplicados para melhorar práticas correntes, orientar futuras pesquisas e informar estratégias educacionais no campo do desenvolvimento de software.

Por fim, serão abordadas as limitações da pesquisa, reconhecendo as restrições e os desafios metodológicos enfrentados durante o estudo.

### 5.1 Respondendo as questões de pesquisa

As discussões sobre os resultados obtidos através da análise de ambos os fóruns de discussão, relacionados a cada questão de pesquisa, serão apresentadas nas seções subsequentes.

### **5.1.1 RQ1: Quais são os desafios de produtividade enfrentados pelos desenvolvedores de software?**

Como discutido no capítulo anterior, os fatores que impactam a produtividade foram organizados em quatro categorias principais: organizacionais, interpessoais, pessoais e técnicos. Inicialmente, avaliou-se a possibilidade de adotar as categorias do *framework* SPACE (Satisfaction and well-being, Performance, Activity, Communication and collaboration, Efficiency and flow), bem como as categorias sugeridas por Forsgren et al. (2021). No entanto, verificou-se que as categorias do modelo SPACE são direcionadas à organização de fatores de produtividade que podem ser metrificados, o que não se mostrou adequado para todos os fatores identificados neste estudo, devido à dificuldade de mensuração de alguns deles.

O esquema de categorização foi desenvolvido seguindo o processo de agrupamento das unidades codificadas em categorias amplas, identificando padrões e similaridades entre os códigos. Após esse processo inicial, o esquema foi revisado por outros dois pesquisadores, que avaliaram a adequação das categorias propostas. A decisão de adotar as categorias Organizacionais, Interpessoais, Pessoais e Técnicos foi fundamentada na clareza, aplicabilidade e na capacidade de cobrir de forma ampla os diversos fatores que afetam as comunidades estudadas.

Todos os fatores identificados alinharam-se de forma precisa e consistente com as categorias estabelecidas, sem a necessidade de ajustes ou esforço adicional. Esta congruência demonstra a eficiência da categorização adotada e da conformidade dos dados obtidos. A seguir são discutidos alguns fatores que se destacaram nas observações realizadas: gestão de tempo; medição de produtividade; adoção de novos padrões/tecnologias; fatores interpessoais (coesão, qualificação e eficácia da comunicação); cultura ruim; e inexperiência. Um outro aspecto também destacado é a diferença entre as perspectivas observadas entre os gerentes e engenheiros de software no que diz respeito aos desafios que afetam a produtividade, evidenciando empiricamente essas diferenças.

#### **Gestão de tempo**

Na categoria de fatores Pessoais está Gestão do Tempo, sendo o tópico mais discutido no fórum de Engenharia de Software. De modo geral, os desafios vinculados à Gestão de Tempo estão relacionados a lidar com múltiplas tarefas, seja se dividir entre diferentes projetos ou gastar muito tempo em uma atividade em detrimento de outra, como acontece quando se gasta muito tempo planejando e sobra pouco tempo para a codificação. Além disso, em algumas discussões, os desenvolvedores também abordaram a melhor forma de aproveitar o tempo ocioso, como o tempo de espera durante a execução dos testes de software.

Durante a etapa de codificação, inicialmente, foi criado o código “Fragmentação do Tempo” para representar os desafios relacionados à troca de contexto e má gestão do tempo. No entanto, foi observado que, ao contrário dos desafios associados à gestão de tempo, nos quais a perda de produtividade resultava de um gerenciamento

inadequado do tempo disponível, nos desafios relacionados à troca de contexto, a perda de produtividade estava relacionada à própria atividade. Ou seja, a troca de contexto para uma atividade específica resultava em uma perda de produtividade. Por exemplo, em uma discussão, um desenvolvedor relatou:

*“When programming backend, it is quite easy (at least for me), to do most of the stuff test driven and just keep hammering out a bunch of code that works, while being able to focus. [...] But I found that as I’m moving more towards the frontend, I am constantly context switching. [...] If I compare this to working on one small library or just some deep backend stuff, I find that my productivity basically non-existent”*

Nesse caso específico, o desenvolvedor enfrenta um desafio relacionado à troca de contexto, visto que precisa alternar entre tarefas de *back-end* e *front-end*. Apesar de haver uma fragmentação do tempo, é importante destacar que esse problema não está relacionado à gestão inadequada do tempo, mas sim à natureza da atividade em si. Durante a discussão sobre esse assunto na comunidade, outros membros sugeriram que o desenvolvedor se concentrasse apenas nas tarefas de *back-end* para evitar a perda de produtividade e que as atividades de *front-end* fossem designadas a outro profissional. Como resultado dessa discussão, o código “Fragmentação do Tempo” foi renomeado para “Troca de Contexto”, e um novo código denominado “Gestão do Tempo” foi criado para abordar os desafios relacionados ao gerenciamento adequado do tempo. Vale ressaltar que há discussões nas quais o desenvolvedor enfrenta esses dois desafios ao mesmo tempo.

Em contraste, no fórum PM, Gestão do Tempo foi mencionada apenas uma vez, sugerindo que essa pode não ser uma questão de produtividade frequentemente enfrentada ou amplamente discutida pela comunidade de gerentes. Essa diferença destaca como a natureza das atividades e os desafios de produtividade podem variar significativamente entre diferentes contextos profissionais. Enquanto desenvolvedores podem enfrentar dificuldades de gerir o tempo disponível devido às demandas de seus trabalhos, gerentes podem encarar desafios distintos que não necessariamente se relacionam diretamente com a gestão do tempo.

### **Medição de produtividade**

Diferente do fórum SE, no fórum PM a maioria das discussões analisadas estão relacionadas ao fator Dificuldade de Medição da Produtividade. A discussão sobre a medição da produtividade em fóruns dedicados a profissionais de gerência, como o PM, e em comunidades técnicas como o SE, revela uma preocupação persistente e complexa que transcende os limites específicos de qualquer campo profissional isolado. A dificuldade em medir a produtividade, especialmente em setores onde o trabalho intelectual predomina, como é o caso da Ciência da Computação, é um desafio amplamente reconhecido e debatido (Graziotin e Fagerholm, 2019).

No fórum PM, a prevalência de discussões sobre a medição da produtividade reflete a realidade cotidiana dos gerentes, que frequentemente enfrentam o desafio de avaliar o desempenho de suas equipes de forma justa e eficaz. Este é um processo complexo,

pois a produtividade em campos como o desenvolvimento de software não pode ser eficientemente capturada por métricas simplistas, como a contagem de linhas de código, que não levam em consideração a qualidade, a criatividade ou a complexidade do trabalho realizado (Forsgren et al., 2021).

A insatisfação com as métricas tradicionais também é evidente no fórum SE, onde profissionais técnicos buscam alternativas mais representativas e significativas para avaliar a produtividade. A discussão nessas comunidades frequentemente gira em torno de encontrar métodos que se alinhem melhor às realidades dinâmicas e multifacetadas de projetos de software, na qual a solução de problemas complexos são mais valorizadas do que a mera quantidade de código produzido.

O interesse demonstrado pelos membros de ambas as comunidades nas práticas adotadas por outras organizações evidencia uma lacuna entre a teoria proposta pela academia e a aplicação prática desses conhecimentos na indústria. Embora estudos científicos continuem a propor e validar novas métricas de produtividade, a análise dos fóruns revela uma necessidade clara de estratégias que promovam a transferência desses conhecimentos para o ambiente corporativo.

A discussão em tais fóruns também destaca a importância de considerar aspectos humanos na medição da produtividade. Além de refinar as métricas, é crucial que qualquer nova metodologia desenvolvida leve em consideração o bem-estar dos funcionários e o clima organizacional, elementos que não são frequentemente desconsiderados em abordagens mais tradicionais, mas que têm um impacto significativo na produtividade geral da equipe Forsgren et al. (2021); Graziotin e Fagerholm (2019).

A compreensão e a medição da produtividade da equipe também são desafiadoras devido a aspectos como trabalho invisível e tarefas de articulação necessárias para a coordenação e planejamento eficazes (Suchman, 1995). Portanto, é fundamental que as organizações desenvolvam uma cultura que não apenas reconheça e apoie a diversidade de habilidades e experiências, mas também promova práticas de comunicação que garantam que todos os membros da equipe estejam engajados e bem informados (Dabbish et al., 2012).

### **Adoção de novos padrões/tecnologias**

O fator “Adoção de novos padrões/tecnologias” é um tema recorrente e significativo nas discussões da comunidade SE, evidenciando-se como o segundo problema de produtividade mais citado. Com nove menções, a questão centraliza-se predominantemente na transição para novos padrões de codificação, que, embora propostos para melhorar a qualidade e a manutenção do código, frequentemente resultam em uma queda temporária na produtividade dos desenvolvedores.

Essa redução na produtividade ocorre porque a adaptação a um novo padrão de codificação requer uma curva de aprendizado inicial, durante a qual os desenvolvedores precisam não só familiarizar-se com as novas práticas, mas também aplicá-las efetivamente em suas tarefas diárias. O tempo investido em aprender e se ajustar a essas novas normas muitas vezes desvia o foco das atividades de desenvolvimento

propriamente ditas, levando a uma diminuição na eficiência geral. Esse impacto é evidenciado no trecho de discussão a seguir:

*“I don’t believe it’s worth exposing ourselves to increased complexity, slower/more development time and a host of liabilities that could put us months behind schedule (when business has already pressured us about our trajectory) in exchange for abstract benefits in theoretical scaling years down the road. I believe other proposed benefits like modularity and separation of concerns can be accomplished in React by simply creating a thoughtful component model, which could situate views in their own self-sufficient component trees and contain errors with error boundaries. In your opinion, are there real benefits to this pattern I’m not seeing?”*

Além dos padrões de codificação, há também casos em que os gerentes de projeto impõem novos padrões de projeto ou a adoção de tecnologias específicas, como exemplificado no trecho abaixo. Essa imposição pode vir acompanhada de desafios adicionais, principalmente quando as decisões são tomadas sem uma consulta adequada às equipes técnicas ou sem considerar a experiência e o conforto dos desenvolvedores com as tecnologias atuais.

*“Once we were halfway the project, the PM stated we had to use third party message queue capabilities instead of threads and had to implement load balancing, what eventually ended up happening was that we eventually started working with Celery, Redis, RabbitMQ, Nginx, uWSGI and a bunch of other large third party services which nobody had any real experience with. In the end this lead to a bunch of spaghetti code, untestable tasks (because of the complexity of third party libraries, patching the code didn’t even work) and a bunch of headaches because nobody even knew what the added value of these services were.”*

A análise das discussões revela que, quando novas ferramentas ou métodos são introduzidos sem treinamento adequado ou uma justificativa clara para sua necessidade, frequentemente ocorre resistência por parte dos desenvolvedores. Esse fenômeno contribui significativamente para a redução da produtividade.

### **Fatores interpessoais**

No desenvolvimento de software, a produtividade da equipe também é profundamente influenciada por fatores interpessoais como coesão da equipe, qualificação dos colegas e eficácia da comunicação Graziotin e Fagerholm (2019). A coesão é particularmente crucial em ambientes onde a integração de membros com diferentes níveis de experiência precisa ser gerenciada cuidadosamente. O impacto da falta de coesão é evidenciado no trecho de discussão a seguir:

*“I’m a rather experienced software developer. I worked with many teams and projects throughout my career so far. The recent two projects, however, challenged me in an unusual way. Namely: they were disorganized, or rather organized in a way that I perceive as not effective.”*

A qualificação dos colegas também desempenha um papel vital, pois equipes com uma ampla gama de habilidades e experiências são mais capazes de resolver pro-

blemas complexos. No entanto, a eficácia dessa diversidade depende fortemente de uma comunicação clara e eficiente (Dourish e Bellotti, 1992). A comunicação ineficiente pode criar barreiras significativas ao sucesso, limitando o acesso à informação necessária e complicando a integração eficaz do trabalho. Isso é particularmente problemático em projetos de software, onde a necessidade de documentação clara e acessível é crítica para o alinhamento e a integração eficazes das atividades da equipe (Dabbish et al., 2012).

Além disso, a colaboração eficaz pode minimizar a necessidade de atividades redundantes, como revisões de código e retrabalho desnecessário, melhorando assim o desempenho do sistema e a qualidade do produto final. Forsgren et al. (2021) destaca que a inclusão e a diversidade dentro das equipes não apenas melhoram o desempenho, mas também podem prevenir o *Burnout* ao reduzir a carga de trabalho individual através da distribuição mais equitativa das tarefas e responsabilidades.

### **Cultura ruim e Inexperiência**

No contexto da análise das discussões do fórum PM, surgiram dois novos fatores que não estavam presentes no fórum SE: “Cultura ruim” e “Inexperiência”. A cultura ruim em uma organização, onde as práticas não favorecem um desempenho eficiente, foi evidenciada em um relato de um membro da comunidade que descreveu uma empresa onde não há distinção salarial ou reconhecimento entre os funcionários, independentemente de seu desempenho. Esse cenário desmotiva os funcionários, pois não há incentivos para os mais produtivos ou consequências para os que têm baixo desempenho, impactando negativamente a motivação e a produtividade.

Paralelamente, a inexperiência foi destacada como um fator que complica a implementação e avaliação eficazes de métricas de desempenho. Um comentário de um participante do fórum revelou dificuldades em definir e medir métricas devido à sua falta de experiência, o que pode impedir a execução eficiente das tarefas e aferição de resultados produtivos. Esses dois fatores, “Cultura ruim” e “Inexperiência”, destacam desafios importantes que vão além das tarefas técnicas, afetando o ambiente de trabalho e a capacidade das equipes de responder eficazmente aos objetivos organizacionais.

### **Diferenças entre a perspectiva de gerentes e engenheiros de software**

A análise das discussões nos fóruns PM e SE revela uma distinção marcante nas fontes de problemas de produtividade relatados pelos membros de cada comunidade. No fórum PM, que é frequentado predominantemente por profissionais de gerência, a maioria dos problemas discutidos está relacionada a questões de equipe, em vez de desafios individuais. Isso contrasta com o fórum SE, no qual os participantes, geralmente programadores, tendem a relatar problemas que afetam diretamente o seu próprio trabalho e produtividade individual.

Essa diferença no foco das discussões pode ser atribuída às distintas responsabilidades e perspectivas que caracterizam os papéis desses dois grupos de profissionais.

Gerentes, como os membros do fórum PM, estão frequentemente envolvidos na coordenação de equipes, na resolução de conflitos interpessoais e na gestão de recursos humanos. Assim, seus desafios são predominantemente externos, centrados na dinâmica de equipe e no alinhamento dos esforços coletivos para atingir objetivos comuns. Por outro lado, os membros do fórum SE, que estão mais imersos em aspectos técnicos do desenvolvimento de software, enfrentam desafios que são intrinsecamente ligados à execução de tarefas específicas dentro de projetos de desenvolvimento.

A discussão sobre essas diferenças é crucial para entender como abordagens diferenciadas são necessárias quando se trata de melhorar a produtividade em diferentes contextos profissionais. Essa compreensão pode orientar a implementação de estratégias mais eficazes e personalizadas que atendam às necessidades específicas de cada grupo, seja no desenvolvimento de habilidades de gestão para líderes ou no aprimoramento de ferramentas e processos para desenvolvedores.

### **5.1.2 RQ2: Como os desafios de produtividade impactam os desenvolvedores de software?**

Para responder à segunda questão de pesquisa, os impactos identificados durante a análise das discussões foram inicialmente categorizados como “Psicológicos e emocionais” e “Físicos”. No entanto, uma análise mais detalhada revelou que certos fatores influenciavam diretamente o desempenho da equipe, o que levou a uma reclassificação mais precisa. Assim, três novas categorias foram definidas: “Problemas de saúde mental”, “Problemas de desempenho” e “Problemas físicos”. Entre os 16 impactos identificados no fórum Stack Exchange, os mais frequentes foram “Variação na produtividade percebida”, “Perda de concentração” e “Insegurança”.

A percepção de produtividade varia significativamente entre os membros do fórum Stack Exchange, evidenciando a falta de consenso sobre o que caracteriza a produtividade no desenvolvimento de software. Um dos principais pontos de divergência é a tendência de alguns membros considerarem exclusivamente a codificação como o principal indicador de produtividade, como exemplificado no trecho a seguir:

*“When I’m working I feel like most of the time I’m not doing actual work. For example after making a change to the code in order to test it, I have to first build the project, and start the server(say JBoss). Upon testing, I find that there is another small issue. So I bring down the server, make the changes, build again and start up the server again. The building and bringing the server up/down is not very useful work.”*

Essa variação na percepção pode resultar em sentimentos de ineficácia e frustração, especialmente quando a produtividade é avaliada com base em métricas mal definidas ou expectativas irrealistas. Tal visão tende a minimizar ou até ignorar a importância de outras atividades essenciais, como planejamento, documentação, revisão de código e testes. Embora essas atividades não aumentem diretamente a quantidade de código produzido, elas são cruciais para garantir a qualidade final do software.

A discussão sobre as diferentes percepções de produtividade deve abordar a necessidade de estabelecer parâmetros claros e abrangentes que considerem todas as atividade do trabalho de desenvolvimento. Sem um entendimento compartilhado sobre o que constitui um trabalho produtivo, profissionais podem sentir-se improdutivos por estarem envolvidos em tarefas que não são imediatamente tangíveis ou visíveis, como exemplificado no trecho abaixo:

*“I really enjoy the actual programming part of programming, and actually writing the code, and figuring it out. But on this project I’ve been working on, most of my time has been spent on kind stupid stuff (trying to run a Java program in the background (I swear javaw does the same thing as java) and trying to put a jar in a jar, and looking at alternative methods for both of these). I feel like I do this on a lot of projects, and when I look at it for a couple hours, and don’t solve my problem, I don’t feel like I did anything, and feel as though it was a waste of time. How do I spend more time programming, get more stuff done, and feel more productive after every programming session?”*

Para reduzir essas percepções distorcidas, é fundamental que as equipes de desenvolvimento adotem uma definição mais ampla de produtividade e passem a medi-la por meio de um conjunto de métricas que considerem diferentes dimensões, conforme proposto por Forsgren et al. (2021) ao desenvolver um *framework* para avaliar a produtividade.

Além disso, a variação na percepção de produtividade também se manifesta na constatação de que a produtividade tende a diminuir ao longo do tempo. Essa percepção é relatada por desenvolvedores que afirmam ter dificuldade em manter um bom nível de desempenho ao longo do tempo, como destacado no comentário abaixo:

*“I can’t help but feel that doing 8 hours of quality programming work from Monday to Friday consistently is very hard.”*

Por outro lado, a ausência de discussões sobre a variação da produtividade percebida no fórum PM pode sugerir que os membros dessa comunidade tendem a se concentrar mais em questões tangíveis e mensuráveis de desempenho, deixando de lado aspectos subjetivos ou emocionais. Isso pode refletir uma cultura profissional voltada para a resolução de problemas concretos, o que é compreensível, dada a natureza das responsabilidades de gerenciamento, que exigem uma abordagem mais objetiva e quantificável das operações.

De modo geral, a análise das discussões no fórum PM revelou uma dinâmica distinta em comparação ao fórum SE, especialmente no que diz respeito à expressão de desafios de produtividade e seus impactos. No fórum PM, apenas duas categorias de impactos foram identificadas: “Problemas de saúde mental” e “Problemas de desempenho”, sem menção a problemas físicos. Além disso, esses impactos surgiram em apenas quatro discussões, o que sugere que os membros do fórum PM raramente

expressam abertamente, nesses ambientes, como os desafios de produtividade os afetam.

Ao contrário do que se observa no fórum SE, onde os membros frequentemente buscam apoio e compartilham experiências pessoais em busca de acolhimento da comunidade, as discussões no fórum PM tendem a ser mais contidas em termos de expressão emocional. Isso pode refletir uma cultura ou norma profissional dentro da comunidade, em que abordar questões emocionais ou de saúde é visto como menos apropriado ou fora do escopo das discussões focadas em temas de gerenciamento.

Entre os problemas de desempenho identificados no fórum PM, a “Estagnação” foi o único discutido de forma clara. Esse problema refere-se à percepção de falta de progresso e ao sentimento de não alcançar resultados significativos, o que pode resultar em desmotivação e redução da produtividade ao longo do tempo. A discussão sobre estagnação ressalta preocupações quanto à capacidade de manter o crescimento profissional e a eficácia nos projetos gerenciados. Outro fator abordado foi a “Preocupação”, que, neste fórum, refere-se à apreensão em relação ao desempenho desigual entre os membros da equipe, com o receio de que essas discrepâncias impactem negativamente tanto a produtividade quanto a harmonia do grupo.

Essas diferenças nas discussões entre os dois fóruns mostram como o contexto profissional e as normas da comunidade influenciam a forma como os problemas de produtividade são percebidos e abordados. No fórum PM, a ênfase recai sobre questões concretas de desempenho e preocupações gerenciais, enquanto no fórum SE há mais abertura para discutir aspectos pessoais e emocionais, refletindo talvez uma maior necessidade de suporte emocional nessa comunidade.

### **5.1.3 RQ3: Como os desafios de produtividade afetam o trabalho realizado pelos desenvolvedores de software?**

Ao analisar os dados do fórum SE para responder a esta questão, foi identificada uma menor variação nos impactos relatados. Dos 65 problemas discutidos, 33 estavam relacionados à “Redução da performance”, sendo este o mais frequente. No fórum PM, o cenário foi semelhante: dos 15 problemas mencionados, 5 também estavam relacionados à “Redução da performance”, tornando-se o impacto predominante em ambos os fóruns.

Dado que o foco do estudo são os desafios relacionados à produtividade, é compreensível que muitos desenvolvedores relatem uma queda no desempenho pessoal ao enfrentarem esses obstáculos. No entanto, outros problemas também foram levantados, como a diminuição da qualidade do código, a necessidade de retrabalho e os atrasos nas entregas.

Um aspecto relevante que emerge dessa análise é a frequente confusão entre os conceitos de produtividade e performance dentro da comunidade. Em alguns casos, os membros que relatam uma perda de produtividade podem, na verdade, estar se

referindo a aspectos relacionados à eficácia, eficiência ou até rentabilidade. Para mitigar essa ambiguidade conceitual, optou-se pela adoção do termo “performance” nas análises, por se tratar de um conceito mais abrangente, que incorpora não apenas a produtividade, mas também a eficiência, a eficácia e a rentabilidade no âmbito da engenharia de software. A Figura 5.1 também apresenta a relação entre esses termos, ilustrando como cada um contribui para o entendimento global do desempenho em projetos de software. Conforme destacado por Brumby et al. (2019), o uso do termo “performance” permite capturar uma visão mais integrada dos fatores que impactam o trabalho dos desenvolvedores e os resultados obtidos.

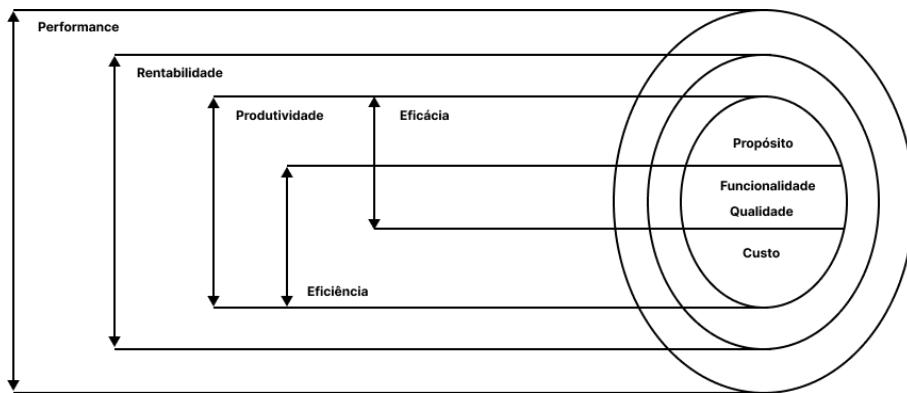


Figura 5.1: Definição integrada de produtividade de software. Adaptado de Brumby et al. (2019).

Na engenharia de software, eficiência refere-se à capacidade de utilizar recursos (tempo, memória, energia computacional, entre outros) de forma otimizada para alcançar os resultados desejados. Por outro lado, eficácia está relacionada à obtenção dos objetivos pretendidos, isto é, se o sistema ou processo entrega a funcionalidade esperada pelos *stakeholders*. Rentabilidade, por sua vez, diz respeito ao equilíbrio entre os custos envolvidos no desenvolvimento do software e os retornos financeiros ou valor gerado. Finalmente, performance é um termo mais abrangente, que engloba todas essas dimensões — eficiência, eficácia, rentabilidade e produtividade — combinadas com a capacidade de sustentar esses resultados ao longo do tempo, sendo essencial para medir o impacto geral das práticas e decisões no desenvolvimento de software Brumby et al. (2019).

Nesse contexto, a produtividade do desenvolvedor surge como uma questão complexa e repleta de nuances, com implicações profundas para as equipes de desenvolvimento de software. Compreender claramente como definir e medir a produtividade pode fornecer às organizações, gerentes e desenvolvedores as ferramentas necessárias para produzir software de forma mais eficiente. No entanto, apesar de décadas de pesquisa e prática, a definição e medição da produtividade do desenvolvedor continuam a ser um desafio Forsgren et al. (2021). No fórum PM, um novo impacto foi identificado: “Desmotivação da equipe”, caracterizado pela falta de entusiasmo, interesse ou comprometimento dos membros com as tarefas ou objetivos do projeto. Esse fenômeno

pode resultar em baixa produtividade e desempenho insatisfatório nas equipes.

Um membro da comunidade exemplificou essa situação ao compartilhar: “*Developer passion has gone. Nobody wants to spend so much time in meetings and all the decision-making power has evaporated, leaving the devs demotivated.*” Esse relato destaca como a perda de paixão pelo trabalho e a redução do poder decisório são fatores cruciais para a desmotivação dos desenvolvedores. Quando as equipes passam muito tempo em reuniões improdutivas e sentem que não possuem controle ou influência sobre as decisões do projeto, a motivação pode rapidamente se deteriorar.

Por outro lado, alguns problemas comuns no fórum SE não foram encontrados no PM, como “Dificuldade de conclusão”, “Dificuldade de manutenção”, “Não identificação de valor agregado” e “Pressão da gerência”. É importante destacar que há menos discussões sobre produtividade no fórum PM, possivelmente devido ao fato de ser uma comunidade mais específica, composta principalmente por gerentes e líderes de projetos, cujas preocupações podem ser mais focadas em questões estratégicas e de liderança. Em contraste, o fórum SE abrange membros de diversos cargos e funções, incluindo desenvolvedores, o que resulta em uma gama mais ampla de discussões sobre produtividade, abrangendo tanto aspectos operacionais quanto emocionais.

#### **5.1.4 RQ4: Quais estratégias e atividades para melhorar a produtividade são discutidas?**

Conforme mencionado anteriormente, a questão RQ4 é a única neste estudo que tem o objetivo de analisar diretamente as respostas dos participantes. Dada a especificidade das estratégias para os problemas enfrentados, o processo de codificação passou por diversas iterações, iniciando com 75 códigos e sendo refinado até alcançar a composição final de 16 códigos.

46,1% das estratégias mencionadas nas respostas do fórum SE envolvem a reformulação da dinâmica de trabalho. Esse dado evidencia que grande parte das sugestões se concentra em incentivar pessoas com problemas de produtividade a realizar ajustes em sua forma de trabalhar, destacando a relevância de mudanças na abordagem das tarefas cotidianas.

Embora inicialmente houvesse uma codificação mais detalhada dessas mudanças, o foco do estudo não é explorar minuciosamente cada atividade, mas sim compreender como a comunidade aborda e reage aos desafios de produtividade de forma geral. Essa abordagem permite uma visão mais abrangente sobre as respostas coletivas e as tendências comportamentais dentro da comunidade, ao invés de se concentrar em particularidades de cada discussão.

Para exemplificar esse agrupamento, a Figura 5.2 apresenta alguns códigos que foram combinados para criar os códigos “Redefinir sua conduta” e “Reconfigurar dinâmica de trabalho” no contexto do fórum SE. No caso de “Redefinir sua conduta”, os códigos agrupados estão relacionados a mudanças no comportamento individual no ambiente

de trabalho, sugerindo que a pessoa ajuste suas atitudes, hábitos e decisões no dia a dia profissional. Já em “Reconfigurar dinâmica de trabalho”, os códigos focam em alterações mais amplas no modo como o trabalho é conduzido, propondo ajustes em processos, ferramentas ou metodologias adotadas para aumentar a eficiência e produtividade.



Figura 5.2: Exemplo de códigos que foram unificados no fórum SE.

Esses dois agrupamentos refletem abordagens distintas para resolver problemas de produtividade: enquanto o primeiro trata de uma transformação pessoal e de conduta, o segundo busca reestruturar o contexto em que o trabalho é realizado, impactando a maneira como as tarefas são organizadas e executadas. Essa diferenciação evidencia a diversidade de estratégias discutidas pela comunidade, abrangendo tanto mudanças individuais quanto ajustes sistêmicos no ambiente de trabalho.

Os códigos “Alterar o método de medição da produtividade” e “Recorrer a ferramentas tecnológicas” também poderiam ter sido agrupados no código “Reconfigurar dinâmica de trabalho”. No entanto, optou-se por não agrupá-las devido a dois fatores importantes. Primeiro, “Recorrer a ferramentas tecnológicas” apresentou um número

expressivo de ocorrências, o que justificou sua análise separada. Segundo, “Alterar o método de medição da produtividade” está diretamente relacionado ao tema central deste estudo, tornando mais apropriado tratá-lo de forma independente.

É interessante notar que o código “Recorrer a ferramentas tecnológicas” destaca uma abordagem mais prática e técnica da comunidade em relação à produtividade. Muitas discussões sugerem o uso de softwares e outras ferramentas tecnológicas para auxiliar na solução de problemas de desempenho, refletindo uma orientação concreta e pragmática na busca por soluções. Isso demonstra que, além de discutir mudanças comportamentais ou dinâmicas de trabalho, a comunidade também valoriza a adoção de recursos tecnológicos como parte de suas estratégias para aumentar a produtividade.

Em diversas discussões, observa-se também que os profissionais frequentemente se deparam com a necessidade de convencer outras pessoas sobre sua opinião ou defender seu ponto de vista. Inicialmente, o primeiro código identificado foi “Defender sua posição de forma convincente”. No entanto, durante as análises, notou-se que as estratégias recomendadas variavam: em alguns casos, o foco estava no diálogo e em como estabelecer uma comunicação eficaz, enquanto em outros a recomendação envolvia a realização de testes ou a apresentação de evidências concretas.

Diante dessa diferenciação, foram criados dois códigos distintos: “Convencer através de dados” e “Convencer através de diálogo”. O primeiro refere-se a situações em que a persuasão ocorre por meio da apresentação de fatos, métricas ou resultados quantitativos, enquanto o segundo foca em estratégias de comunicação, buscando o entendimento e a negociação por meio da conversa.

Um exemplo claro de “Convencer através de dados” pode ser encontrado na seguinte discussão do fórum:

*“Management don’t really care what you think of a particular design pattern. So don’t bother burying them with technical details. What they do care about is when the thing will be delivered, and how much it will cost. So come up with your best estimates for how much time it will take to develop your way, and how much the lead developer’s way. If it turns out that it will be delivered later and cost more, then that’s your argument for doing it your way.”*

Esse exemplo ilustra a ideia de que, em certos contextos, apresentar dados concretos, como estimativas de tempo e custo, é mais eficaz para persuadir gestores e tomadores de decisão. Nesse caso, o argumento é construído em torno de resultados objetivos, como prazo de entrega e impacto financeiro, o que reforça a importância de adaptar a estratégia de convencimento à situação e ao público envolvido.

Assim, esses dois códigos refletem abordagens complementares, mostrando que a comunidade reconhece a importância tanto do diálogo quanto dos dados na defesa de uma posição, dependendo do contexto e dos interlocutores.

Entre as estratégias identificadas, foi possível observar que algumas adotam uma abordagem mais humanista, oferecendo conselhos pessoais e reflexivos, como é o

caso dos códigos “Adaptar sua perspectiva” e “Desvincular-se da responsabilidade”. Essas estratégias se distanciam das recomendações técnicas e evidenciam o reconhecimento, pela comunidade, de que nem todos os problemas de produtividade podem ser resolvidos com soluções pragmáticas.

A estratégia “Adaptar sua perspectiva” está muito relacionada à ideia de mudar a forma como se vê uma situação, ou seja, olhar para o problema sob um ângulo diferente. Esse código reflete sugestões onde se aconselha a pessoa a reconsiderar suas crenças ou suposições, sugerindo que talvez o erro esteja em sua interpretação da situação e que ajustar sua visão possa trazer novas soluções ou alívio. Trata-se de uma abordagem introspectiva, incentivando o indivíduo a reavaliar suas percepções e a ser mais flexível em relação aos desafios que enfrenta. Por exemplo, em um dos diálogos analisados, alguém questiona a certeza do outro sobre a percepção que tem de seus colegas:

*“Have you spoken to your development colleagues about this? How do you know they lack education? That’s quite a sweeping statement and you’ll probably find you’re wrong. I don’t think it’d go down too well if a new grad started meddling with processes without understanding why they’re like that in the first place. Managers love processes and love tracking and love being made look good, so the battle-hardened skeptic in me feels that your manager is interested in your views because it’s something they can take credit for.”*

Esse trecho sugere que a pessoa deve reconsiderar sua opinião inicial e avaliar a situação sob uma nova ótica, levando em conta outros fatores que podem estar influenciando sua visão. Por outro lado, “Desvincular-se da responsabilidade” refere-se a situações em que se reconhece que certos problemas ou obstáculos estão além do controle individual, e que, portanto, o profissional não deve se sentir responsável por resolvê-los ou se preocupar excessivamente. Essa estratégia promove um alívio do peso emocional que muitas vezes acompanha responsabilidades que, na realidade, fogem da capacidade de ação da pessoa. É uma sugestão para aceitar que, em determinados casos, não há muito o que fazer e que não vale a pena investir energia em situações que não podem ser alteradas.

Além disso, o código “Cuidar da saúde física e mental” reflete conselhos voltados ao bem-estar pessoal, recomendando, por exemplo, que a pessoa durma adequadamente, se alimente melhor ou cuide de sua saúde mental. Esse conjunto de estratégias sugere que, para muitos membros da comunidade, melhorar a produtividade não se limita a técnicas de trabalho ou ferramentas, mas também envolve cuidar do corpo e da mente, reconhecendo o impacto direto do bem-estar físico e emocional na capacidade de produzir de forma eficaz.

Essas estratégias demonstram um lado da comunidade que vai além das questões técnicas, trazendo à tona um aspecto mais humano e sensível, onde as soluções para problemas de produtividade envolvem tanto o equilíbrio emocional quanto a reflexão pessoal. Diferente do fórum SE, onde “Reconfigurar a dinâmica de trabalho” é a estratégia mais recomendada para enfrentar problemas de produtividade, no fórum

PM a ênfase recai sobre “Alterar o método de medição de produtividade”. Essa divergência está diretamente ligada ao perfil distinto das duas comunidades.

O fórum SE é composto por profissionais de diversas áreas da tecnologia, incluindo desenvolvedores, engenheiros e gerentes. Nesse ambiente, as discussões sobre produtividade frequentemente abordam aspectos técnicos e operacionais do trabalho diário, refletindo a diversidade de experiências e responsabilidades dos seus membros.

Por outro lado, o fórum PM é formado principalmente por gerentes e líderes de projeto, cuja principal responsabilidade é garantir o sucesso dos projetos que lideram. Para esses profissionais, a medição precisa da produtividade é fundamental, pois permite monitorar o progresso, alocar recursos de maneira eficiente e identificar áreas que necessitam de melhoria. Consequentemente, a estratégia de “Alterar o método de medição de produtividade” é mais prevalente nesse fórum, uma vez que métricas adequadas são essenciais para uma gestão eficaz de projetos.

Essa diferença de enfoque entre os fóruns SE e PM ilustra como as necessidades e responsabilidades específicas de cada grupo influenciam as estratégias recomendadas para melhorar a produtividade.

## 5.2 Mapa conceitual: relacionando fatores e impactos sobre a produtividade

Com o objetivo de organizar e compreender as relações entre os dados obtidos nos dois fóruns analisados, foi construído um mapa conceitual que integra as respostas às questões RQ1, RQ2 e RQ3. Este mapa busca representar as relações entre os fatores que influenciam a produtividade (RQ1), os impactos que esses fatores geram nos profissionais (RQ2), e como eles afetam suas tarefas (RQ3).

O mapa conceitual está representado em duas figuras principais:

1. A Figura 5.3 relaciona os fatores (RQ1) aos impactos nos profissionais (RQ2).
2. A Figura 5.4 conecta os fatores (RQ1) aos impactos nas tarefas (RQ3).

### 5.2.1 Estrutura das figuras

As figuras foram divididas em quatro áreas principais, de acordo com as categorias dos fatores: fatores técnicos, fatores organizacionais, fatores pessoais e fatores impecosais. Cada impacto está representado em um retângulo, com o texto em letras maiúsculas e na cor cinza. Os impactos sobre os profissionais e sobre as tarefas estão destacados em letras minúsculas e coloridas conforme a categoria do impacto. A legenda das figuras fornece as informações necessárias para identificar as categorias.

As setas indicam a relação entre os fatores e os impactos, sempre saindo de um fator em direção a um impacto. Quando um fator não é representado no mapa, como

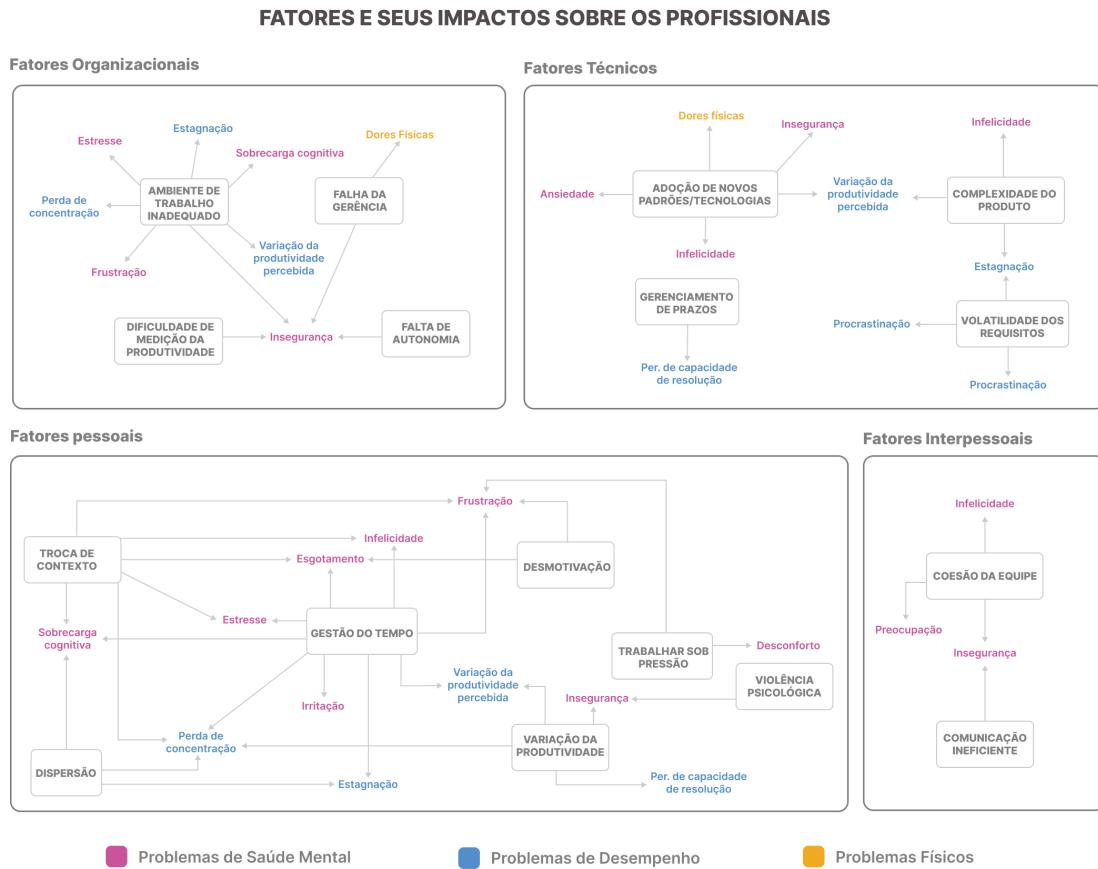


Figura 5.3: Fatores e seus impactos sobre os profissionais.

é o caso de “Inexperiência”, isso ocorre porque as discussões relacionadas a esse fator não descrevem impactos sobre pessoas ou tarefas. Essa ausência reflete uma característica das interações nos fóruns analisados, onde nem todas as discussões continham respostas para as questões RQ2 e RQ3.

### 5.2.2 Considerações sobre a representação

Este estudo adotou uma abordagem qualitativa para a análise dos dados, optando por ocultar a quantidade de ocorrências de cada relação no mapa conceitual, com o objetivo de simplificar sua visualização. No entanto, é importante ressaltar que cada seta no gráfico representa pelo menos uma ocorrência registrada entre o fator e o impacto correspondente. Para detalhes sobre o número de ocorrências, é possível consultar as tabelas apresentadas no Capítulo 4. Entre as relações analisadas, o maior número de ocorrências registrado foi 11.

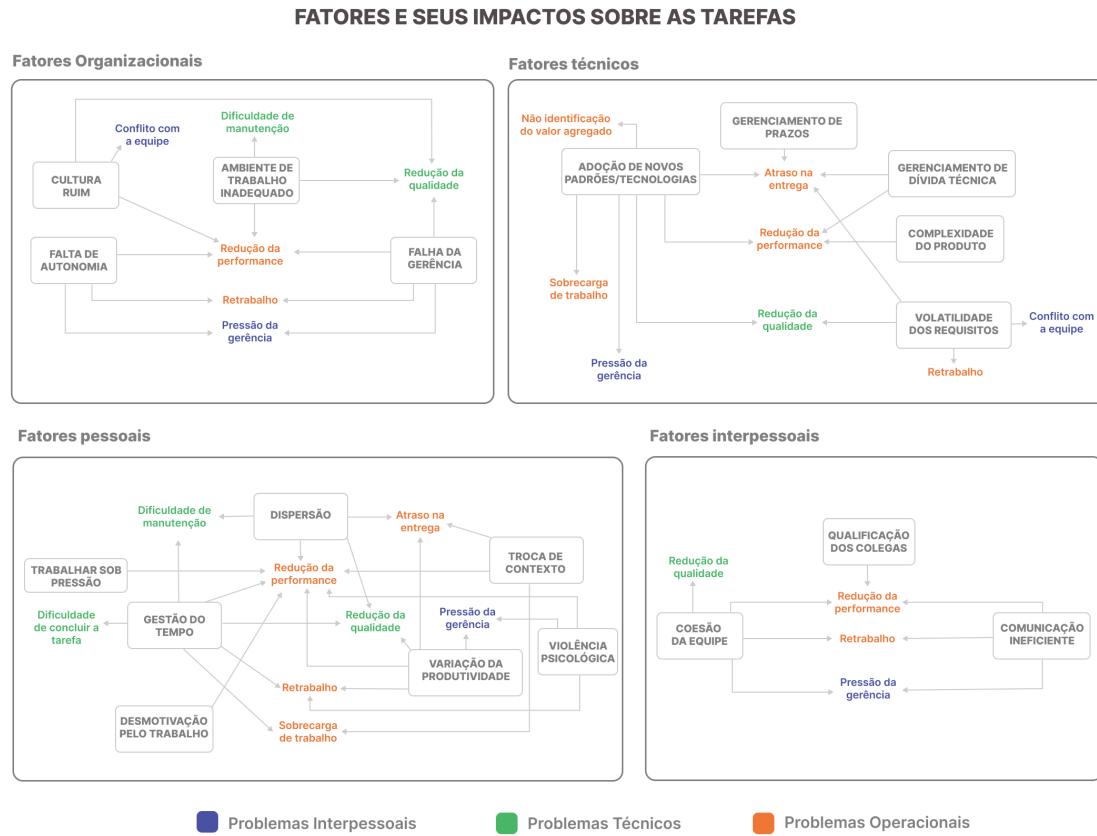


Figura 5.4: Fatores e seus impactos sobre as tarefas.

### 5.2.3 Observações gerais

A análise conceitual evidencia que, para as comunidades estudadas, os fatores técnicos, como “Adoção de novos padrões/tecnologias” e “Gerenciamento de dívida técnica”, estão associados a impactos diretos nas tarefas, como “Atraso na entrega” e “Redução da performance”. Já fatores organizacionais, como “Falha da gerência” e “Falta de autonomia”, apresentam impactos significativos tanto nos profissionais (Figura 5.3), como “Insegurança”, quanto nas tarefas (Figura 5.4), como “Retrabalho” e “Pressão da gerência”. Por outro lado, fatores pessoais, como “Gestão do tempo” e “Troca de contexto”, estão predominantemente relacionados a impactos nos profissionais, como “Esgotamento”, “Estresse” e “Infelicidade”. Por fim, fatores interpessoais, como “Qualificação dos colegas” e “Coesão da equipe”, mostram um impacto mais direto nas tarefas, como “Redução da performance”.

Ao organizar as relações em um mapa conceitual, este estudo fornece uma visão integrada das interações complexas entre os fatores que impactam a produtividade e seus efeitos. As figuras apresentadas não apenas sintetizam os resultados obtidos, mas também servem como uma base útil para direcionar futuras pesquisas e inter-

venções práticas. Essas representações permitem que pesquisadores e profissionais identifiquem áreas prioritárias para atuação, oferecendo um caminho claro para o desenvolvimento de estratégias mais eficazes para mitigar os impactos negativos na produtividade em diferentes contextos.

### 5.3 Comparação com trabalho relacionado

A fim de enriquecer a compreensão dos fatores que influenciam a produtividade em diferentes contextos, foi selecionado o estudo de Wagner e Murphy-Hill (2019) para uma análise comparativa. Este estudo reúne diversos fatores provenientes de pesquisas científicas, oferecendo uma base sólida para identificar pontos de convergência e distinção. Durante a comparação, foram identificadas correspondências entre os fatores que afetam a produtividade apresentados neste estudo e aqueles descritos por Wagner e Murphy-Hill (2019). Na maioria dos casos, embora essas correspondências não sejam diretas, elas refletem similaridades que possibilitam uma análise mais abrangente das questões de produtividade e dos impactos observados em diferentes ambientes de trabalho.

Vale ressaltar que este estudo baseou-se em literatura cinza, enquanto Wagner e Murphy-Hill (2019) utilizou literatura branca, derivada de artigos científicos revisados por pares. Essa diferença nas fontes torna interessante observar que, apesar de suas origens distintas, ambos os estudos chegaram a resultados semelhantes. Enquanto a literatura branca oferece uma base validada e teórica, a literatura cinza explora experiências práticas e relatos informais, permitindo a identificação de novos fatores que afetam a produtividade e que ainda não são amplamente discutidos na literatura acadêmica. Esses novos conhecimentos, revelados pela literatura cinza, abrem caminho para futuras investigações acadêmicas, sugerindo áreas de pesquisa e complementando a compreensão dos impactos da produtividade em contextos de trabalho reais.

Para o fórum SE, alguns fatores identificados encontram correspondências próximas no trabalho de Wagner e Murphy-Hill (2019). No caso de “Adoção de novos padrões/tecnologias”, os fatores mais próximos em Wagner e Murphy-Hill (2019) são “*Platform experience*” e “*Application domain experience*”. “*Platform experience*” refere-se a “*the familiarity with the hardware and software platforms*”, enquanto “*Application domain experience*” indica “*the familiarity with the application domain*”. Essas correspondências sugerem que a falta de domínio e a adoção de novas ferramentas podem impactar diretamente a produtividade, uma vez que a experiência e o conhecimento prévio dos profissionais em determinadas plataformas e domínios influenciam sua capacidade de adaptação e eficácia no uso dessas tecnologias.

O fator “Ambiente de trabalho inadequado” também encontra correspondência com “*Proper workplace*”, descrito por Wagner e Murphy-Hill (2019) como “*the suitability of the workplace to do creative work*”, com ambos enfatizando a importância de um ambiente propício para o trabalho criativo e produtivo. Já “Coesão da equipe” se

aproxima do fator “*Team cohesion*”, que é descrito como “*the cooperativeness of the stakeholders*”, refletindo a relevância da cooperação e da colaboração entre membros da equipe para um desempenho produtivo.

A complexidade técnica surge em ambos os estudos, representada como ”Complexidade do produto” neste estudo e ”*Product complexity*” em Wagner e Murphy-Hill (2019), com a descrição ”*the complexity of the function and structure of the software*”. Ambos abordam a complexidade estrutural e funcional do software a ser desenvolvido como um fator significativo que afeta a produtividade.

Outros fatores, como ”Comunicação ineficiente” e ”Gerenciamento de prazos”, também encontram paralelos. Para o primeiro, ”*Communication*” e ”*Credibility*” são os fatores mais próximos. ”*Communication*” refere-se a ”*the degree and efficiency of which information flows in the team*”, enquanto ”*Credibility*” aborda ”*open communication and competent organization*”, abrangendo tanto a eficiência do fluxo de informações quanto a confiança na organização. Em relação ao gerenciamento de prazos, o fator ”*Schedule*” reflete uma preocupação semelhante, descrito por Wagner e Murphy-Hill (2019) como ”*the appropriateness of the schedule for the development task*”.

O código ”Troca de contexto” encontra correspondência em ”*Time fragmentation*”, descrito como ”*the amount of necessary ‘context switches’ of a person*”, ambos tratando da necessidade de alternância entre diferentes contextos, o que pode impactar a continuidade do trabalho e a produtividade.

No campo do bem-estar psicológico, ”Violência psicológica” possui um fator relacionado em Wagner e Murphy-Hill (2019): ”*Psychological safety*”, definido como ”*the atmosphere is safe for risk-taking*”. Embora não sejam exatamente correspondentes, esses conceitos estão conectados, pois ”Violência psicológica” refere-se a práticas ou comportamentos que intimidam, ameaçam ou desestabilizam emocionalmente o indivíduo, enquanto ”*Psychological safety*” enfoca a criação de um ambiente onde as pessoas se sintam seguras para correr riscos sem medo de repercuções negativas. Essa diferença destaca que, enquanto ”Violência psicológica” aponta diretamente para ações que prejudicam o bem-estar, ”*Psychological safety*” descreve a ausência de tais ameaças, promovendo um ambiente seguro e apoiador.

No fórum PM, o fator ”Inexperiência” encontra correspondências nos fatores ”*Manager application domain experience*” e ”*Manager capability*” do trabalho de Wagner e Murphy-Hill (2019). ”*Manager application domain experience*” refere-se à ”*the familiarity of the manager with the application*”, enquanto ”*Manager capability*” abrange ”*the control of the manager over the project*”. Por sua vez, ”Volatilidade dos requisitos” corresponde ao fator ”*Requirements stability*”, descrito como ”*the number of requirements changes*”. Essas correspondências destacam a importância da experiência e da capacidade de gestão como fatores que afetam diretamente a produtividade das equipes.

Um resumo da comparação entre este estudo e o de Wagner e Murphy-Hill (2019) é

apresentado na Figura 5.5.

Fatores encontrados neste estudo	Fatores encontrados no estudo de Brumby et al., 2019
<b>Adoção de novos padrões/tecnologias</b>	<ul style="list-style-type: none"> <li>• <b>Platform experience:</b> “The familiarity with the hardware and software platforms”</li> <li>• <b>Application domain experience:</b> “The familiarity with the application domain”</li> </ul>
<b>Ambiente de trabalho inadequado</b>	<ul style="list-style-type: none"> <li>• <b>Proper workplace:</b> “The suitability of the workplace to do creative work”</li> </ul>
<b>Coesão da equipe</b>	<ul style="list-style-type: none"> <li>• <b>Team cohesion:</b> “The cooperativeness of the stakeholders”</li> </ul>
<b>Complexidade do produto</b>	<ul style="list-style-type: none"> <li>• <b>Product complexity:</b> “The complexity of the function and structure of the software”</li> </ul>
<b>Comunicação ineficiente</b>	<ul style="list-style-type: none"> <li>• <b>Communication:</b> “The degree and efficiency of which information flows in the team”</li> <li>• <b>Credibility:</b> “Open communication and competent organization”</li> </ul>
<b>Gerenciamento de prazos</b>	<ul style="list-style-type: none"> <li>• <b>Schedule:</b> “The appropriateness of the schedule for the development task”</li> </ul>
<b>Troca de contexto</b>	<ul style="list-style-type: none"> <li>• <b>Time fragmentation:</b> “The amount of necessary ‘context switches’ of a person</li> </ul>
<b>Violência psicológica</b>	<ul style="list-style-type: none"> <li>• <b>Psychological safety:</b> “The atmosphere is safe for risk-taking”</li> </ul>
<b>Inexperiência</b>	<ul style="list-style-type: none"> <li>• <b>Manager application domain experience:</b> “The familiarity of the manager with the application”</li> <li>• <b>Manager capability:</b> “The control of the manager over the project”</li> </ul>
<b>Volatilidade dos requisitos</b>	<ul style="list-style-type: none"> <li>• <b>Requirements stability:</b> “The number of requirements changes”</li> </ul>

Figura 5.5: Resumo da comparação entre os estudos.

Durante a análise comparativa, não foi encontrado um fator correspondente direto para “Cultura ruim”. No entanto, o estudo de Wagner e Murphy-Hill (2019) são abordadas questões culturais através de dois grupos de fatores: *“Corporate Culture”* e *“Team Culture”*. Esses grupos incluem fatores como *“Camaraderie”*, *“Clear goals”*, *“Support for innovation”*, *“Turnover”* e *“Respect”*, que refletem elementos essenciais para uma cultura organizacional saudável. Embora não haja uma correspondência exata, esses fatores sugerem que uma cultura bem estruturada e de apoio pode influenciar positivamente a produtividade, enquanto sua ausência pode representar uma barreira ao desempenho eficiente.

Na análise comparativa, os seguintes fatores identificados neste estudo não encontraram correspondência direta no trabalho de Wagner e Murphy-Hill (2019): “Desmotivação”, “Dificuldade de medição da produtividade”, “Dispersão”, “Falha da gerência”, “Falta de autonomia”, “Gerenciamento de dívida técnica”, “Gestão do tempo”, “Qualificação dos colegas”, “Trabalhar sob pressão” e “Variação da produtividade”.

A ausência desses fatores no trabalho de Wagner e Murphy-Hill (2019) destaca a relevância da literatura cinza, utilizada neste estudo, para acrescentar uma perspectiva mais ampla sobre a produtividade no ambiente de trabalho.

## 5.4 Limitações do estudo

Este estudo apresenta algumas limitações, e nesta seção são descritas as ações adotadas para mitigar esses aspectos na metodologia de pesquisa.

A abordagem adotada foi baseada no *Total Quality Framework (TQF)*, uma estrutura de pesquisa qualitativa que se mostrou mais apropriada para o presente trabalho do que o modelo tradicional de validade interna/externa/construto, comumente aplicado em estudos quantitativos (Roller e Lavrakas, 2015). A seguir, as limitações são discutidas em termos dos subdomínios de credibilidade (coleta de dados), analisabilidade (análise de dados), transparência (relato) e utilidade (resultados).

Quanto à credibilidade, buscou-se reforçar a completude e a precisão da coleta de dados. O escopo do estudo foi limitado à amostra representativa de textos coletados dos fóruns SE e PM. Reconhece-se que a análise de outras plataformas poderia gerar respostas diferentes às questões de pesquisa. Assim, o estudo não se propõe a ser exaustivo, e os achados são específicos à amostra de textos obtidos nessas comunidades.

A precisão da análise e das interpretações depende do julgamento humano, o que pode introduzir vieses de pesquisa. Para mitigar essa limitação, foi realizada uma etapa de calibração, na qual quatro pesquisadores independentes revisaram os códigos gerados, e a análise foi discutida com pesquisadores experientes até atingir o ponto de saturação. No entanto, reconhece-se que os códigos permanecem abertos a interpretações, e, portanto, o conjunto de dados é disponibilizado como material suplementar, permitindo que outros pesquisadores examinem e validem a análise.

Quanto à transparência do estudo, disponibiliza-se no Zenodo o *dataset* utilizado nas análises<sup>1</sup>, permitindo que outros pesquisadores acessem e analisem os dados para verificar e validar as conclusões apresentadas. O Zenodo é uma plataforma amplamente utilizada para *Open Science*, pois oferece um repositório confiável e acessível para o compartilhamento de *datasets*, códigos e publicações científicas, promovendo a reproduzibilidade, a transparência e a colaboração entre pesquisadores de diferentes áreas (Sicilia et al., 2017). Ao hospedar os dados neste repositório, busca-se não apenas validar o presente estudo, mas também fomentar futuras pesquisas que possam ampliar ou questionar seus achados.

No que diz respeito à utilidade dos achados, acredita-se que este estudo possa ser valioso tanto para pesquisadores quanto para profissionais, pois aprofunda a compreensão dos fatores que afetam a produtividade em diferentes contextos.

No que diz respeito à utilidade dos achados, este estudo aborda uma fonte de dados pouco explorada em estudos sobre produtividade na engenharia de software: a literatura cinza. Essa abordagem permitiu identificar dimensões adicionais de produtividade e desafios específicos enfrentados pelos profissionais, conforme discutido

---

<sup>1</sup><https://doi.org/10.5281/zenodo.14636553>

na Seção 5.3. Além disso, o trabalho gerou implicações práticas para diferentes públicos, conforme apresentado na seção a seguir.

## 5.5 Implicações do estudo

Os resultados deste estudo trazem implicações para pesquisadores, profissionais da área e educadores, ao explorar não apenas os fatores que afetam a produtividade, mas também como esses fatores impactam os profissionais, suas tarefas e os tipos de estratégias recomendadas pelas comunidades para lidar com esses desafios.

**Para pesquisadores:** Este estudo reforça a importância de investigar os diversos fatores que afetam a produtividade em diferentes contextos, destacando áreas que ainda necessitam de maior exploração. Embora fatores como cultura organizacional já estejam presentes na literatura científica (Brumby et al., 2019), este estudo, baseado em literatura cinza, identifica elementos adicionais que impactam diretamente a produtividade e que não são amplamente explorados. Espera-se que esses achados incentivem novas pesquisas sobre as inter-relações entre esses fatores, incluindo como eles influenciam a produtividade e o bem-estar dos profissionais em contextos industriais e de alta complexidade.

**Para profissionais:** Os achados deste estudo destacam a importância de compreender e gerenciar os variados fatores que impactam a produtividade, como comunicação, coesão de equipe e gestão de prazos. Empresas e líderes de equipe, em particular, podem se beneficiar ao considerar esses fatores para implementar práticas de gestão que não apenas otimizem a produtividade, mas também preservem o bem-estar dos colaboradores.

**Para educadores:** Os resultados deste estudo oferecem uma base valiosa para a formação de novos profissionais em engenharia de software, ao destacar como fatores de produtividade e desafios cotidianos impactam a performance dos profissionais. O estudo também revela como esses fatores podem afetar o bem-estar dos praticantes, permitindo que educadores preparem melhor os estudantes para os desafios do mercado. Educadores podem usar esses *insights* para desenvolver intervenções pedagógicas que ajudem estudantes a se preparar para o mercado, integrando habilidades de gerenciamento de tempo, adaptação à dinâmica de equipe e enfrentamento de dificuldades organizacionais e pessoais.

Essas implicações sugerem que os achados do estudo, ao identificar tanto os fatores que afetam a produtividade quanto os impactos que eles podem causar, oferecem um avanço significativo para o conhecimento, a prática e a formação em engenharia de software.

# Capítulo 6

## Conclusão

O título deste trabalho, “Investigando a percepção de praticantes da Engenharia de Software sobre produtividade: uma análise com base na literatura cinza”, reflete diretamente o foco central deste estudo: compreender como engenheiros de software percebem e enfrentam os desafios relacionados à produtividade em seu cotidiano. Ao privilegiar o ser humano como elemento central, e não as organizações, o estudo enfatiza as experiências individuais, as dificuldades enfrentadas e as estratégias adotadas pelos praticantes para superar barreiras que comprometem sua eficiência e desempenho.

A análise incluiu duas comunidades distintas na plataforma Stack Exchange: *Software Engineering* e *Product Manager*, que representam diferentes papéis e perspectivas dentro do ecossistema da engenharia de software. A primeira comunidade é composta majoritariamente por desenvolvedores e engenheiros, que estão diretamente envolvidos nas atividades técnicas de codificação, design e implementação. Já a segunda é formada por gerentes de produto, cujas responsabilidades estão mais alinhadas ao planejamento estratégico, monitoramento de desempenho e tomada de decisões que abrangem o produto como um todo.

O objetivo principal foi investigar os desafios de produtividade vivenciados por essas comunidades, identificando fatores que os influenciam e analisando o impacto direto sobre os profissionais e suas tarefas. Além disso, o trabalho explorou as estratégias utilizadas por esses grupos para mitigar os efeitos dos fatores que impactam negativamente sua produtividade. Ao focar na percepção dos próprios praticantes, este estudo reforça a importância de considerar as experiências individuais como fonte valiosa de dados, promovendo uma visão mais centrada no ser humano dentro da engenharia de software.

Os resultados deste estudo evidenciaram uma série de desafios de produtividade que variam em complexidade e natureza, abrangendo tanto aspectos técnicos quanto humanos. No total, foram identificados 21 fatores distintos que geram desafios que impactam diretamente o desempenho dos profissionais. Entre os principais fatores,

destacam-se a gestão do tempo, a adoção de novos padrões e tecnologias, a dificuldade em medir a produtividade de maneira eficaz e a troca constante de contexto, que dificulta a continuidade do trabalho. Esses desafios refletem a crescente pressão por resultados rápidos e eficientes, intensificada pela competitividade no setor de software (Murphy-Hill e Wagner, 2019), onde as equipes precisam melhorar a performance e reduzir os custos de produção para atender às expectativas de prazos de entrega cada vez mais apertados.

O estudo também identificou 17 aspectos humanos que demonstram como esses desafios afetam os profissionais. Entre os impactos observados estão a desmotivação pelo trabalho, o aumento da pressão psicológica, dificuldades de concentração e a variação da produtividade percebida. Esses fatores revelam que a produtividade não pode ser vista apenas sob uma perspectiva técnica; há um componente humano significativo, que envolve questões como saúde mental, bem-estar e satisfação com o trabalho. O impacto emocional e psicológico desses desafios é especialmente relevante, pois pode levar a um ciclo negativo, em que a infelicidade diminui ainda mais a capacidade produtiva dos profissionais.

Além dos desafios e impactos identificados, este estudo explorou as estratégias que as comunidades utilizam para enfrentar os fatores que afetam a produtividade. Foram observadas diversas abordagens, desde ajustes na dinâmica de trabalho até mudanças na forma de medir a produtividade e recomendações de ferramentas para otimizar o desempenho. Essas estratégias incluem métodos práticos, como reorganização de tarefas e uso de métricas específicas, e abordagens mais reflexivas, como a adaptação da perspectiva e a promoção de um ambiente de trabalho seguro e colaborativo. A variedade dessas estratégias reflete a flexibilidade dos profissionais em tentar adaptar o ambiente e os processos para lidar com os desafios de produtividade, sugerindo que a comunidade reconhece a importância de equilibrar os aspectos técnicos e humanos no trabalho.

Outra contribuição deste estudo foi a comparação entre os resultados encontrados através da exploração de literatura cinza e aqueles presentes na literatura branca, representada pelo estudo de Brumby et al. (2019). Essa análise comparativa entre os trabalhos evidenciou que, embora existam correspondências entre alguns fatores, novos elementos específicos da produtividade foram identificados neste estudo, especialmente relacionados à prática e à vivência dos profissionais. Esses novos fatores, derivados da literatura cinza, podem ser explorados de maneira mais aprofundada em estudos futuros para avaliar sua relevância em contextos científicos e práticos. Além disso, a comparação entre as perspectivas das duas comunidades – *Software Engineering* e *Product Manager* – mostrou que cada grupo enfrenta desafios de produtividade semelhantes, mas com perspectivas diferentes, refletindo as especificidades e as prioridades de suas respectivas funções.

Este estudo oferece contribuições relevantes para a área de produtividade no setor de desenvolvimento de software. Ao trazer uma análise dos fatores que influenciam a produtividade, ele contribui para uma compreensão mais ampla dos desafios en-

frentados pelas equipes de desenvolvimento e gestão. Além disso, ao identificar tipos de estratégias utilizadas pelos profissionais, o estudo proporciona uma base empírica que pode ser útil para a formulação de novas práticas organizacionais e de gestão, orientadas a melhorar o ambiente de trabalho e a performance das equipes.

Essas descobertas podem ter implicações práticas tanto para empresas quanto para profissionais. As organizações podem usar os fatores e impactos identificados para criar ambientes de trabalho mais favoráveis e produtivos, que incentivem o equilíbrio entre a demanda por alta performance e o bem-estar dos profissionais. Para os profissionais, os resultados deste estudo podem servir como guia para enfrentar desafios de produtividade no dia a dia, proporcionando *insights* sobre práticas de sucesso compartilhadas pela própria comunidade.

Como trabalhos futuros, sugere-se uma análise mais aprofundada das estratégias utilizadas pelas comunidades, visando validar sua eficácia em contextos científicos e práticos. A aplicação dessas estratégias em ambientes controlados e a análise de seus resultados poderiam oferecer uma visão mais concreta sobre o impacto real dessas práticas na produtividade. Outra possibilidade de pesquisa futura é expandir o estudo para outras comunidades de desenvolvimento de software e áreas correlatas, com o intuito de verificar se os mesmos fatores e desafios se aplicam a outros contextos e se surgem novos fatores específicos de cada área.

# Referências

- Adams, R. J., Smart, P., e Huff, A. S. (2016). Shades of grey: Guidelines for working with the grey literature in systematic reviews for management and organizational studies. *International Journal of Management Reviews*, 19(4):432–454.
- Ahmed, S. e Bagherzadeh, M. (2018). What do concurrency developers ask about?: A large scale study using stack overflow. In *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. ACM.
- Allamanis, M. e Sutton, C. (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.
- Asaduzzaman, M., Mashiyat, A. S., Roy, C. K., e Schneider, K. A. (2013). Answering questions about unanswered questions of stack overflow. In *2013 10th Working Conference on Mining Software Repositories (MSR)*, páginas 97–100. IEEE.
- Bajaj, K., Pattabiraman, K., e Mesbah, A. (2014). Mining questions asked by web developers. In *Proceedings of the 11th Working Conference on Mining Software Repositories*, MSR 2014, página 112–121, New York, NY, USA. Association for Computing Machinery.
- Barua, A., Thomas, S. W., e Hassan, A. E. (2014). What are developers talking about? an analysis of topics and trends in stack overflow. *Empirical Softw. Engg.*, 19(3):619–654.
- Bazelli, B., Hindle, A., e Stroulia, E. (2013). On the personality traits of StackOverflow users. In *2013 IEEE International Conference on Software Maintenance*. IEEE.
- Boehm, B. (1981). *Software Engineering Economics*. Prentice-Hall advances in computing science and technology series. Prentice-Hall.
- Bonato, S. (2018). *Searching the Grey Literature*. Rowman & Littlefield.
- Brumby, D. P., Janssen, C. P., e Mark, G. (2019). *How Do Interruptions Affect Productivity?*, páginas 85–107. Apress, Berkeley, CA.

- Canedo, E. D. e Santos, G. A. (2019). Factors affecting software development productivity: An empirical study. In *Proceedings of the XXXIII Brazilian Symposium on Software Engineering*, SBES 2019, página 307–316, New York, NY, USA. Association for Computing Machinery.
- Cerqueira, L., Freire, S., Neves, D. F., Bastos, J. P. S., Santana, B., Spínola, R., Mendonça, M., e Santos, J. A. M. (2024). Empathy and its effects on software practitioners' well-being and mental health. *IEEE Software*, 41(4):95–104.
- Chapetta, W. A. e Travassos, G. H. (2020). Towards an evidence-based theoretical framework on factors influencing the software development productivity. *Empirical Software Engineering*, 25:3501–3543.
- Clincy, V. A. (2003). Software development productivity and cycle time reduction. *J. Comput. Sci. Coll.*, 19(2):278–287.
- Cruzes, D. S. e Dyba, T. (2011). Recommended steps for thematic synthesis in software engineering. In *2011 International Symposium on Empirical Software Engineering and Measurement*, páginas 275–284.
- Curtis, B. (1991). Techies as nontechnological factors in software engineering? In *Proceedings 13th International Conference on Software Engineering*, páginas 147–148.
- Dabbish, L., Stuart, C., Tsay, J., e Herbsleb, J. (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*, CSCW '12, página 1277–1286, New York, NY, USA. Association for Computing Machinery.
- Dourish, P. e Bellotti, V. (1992). Awareness and coordination in shared workspaces. In *Proceedings of the 1992 ACM Conference on Computer-Supported Cooperative Work*, CSCW '92, página 107–114, New York, NY, USA. Association for Computing Machinery.
- Dutra, A. C., Prikladnicki, R., e França, C. (2015). What do we know about high performance teams in software engineering? results from a systematic literature review. In *2015 41st Euromicro Conference on Software Engineering and Advanced Applications*, páginas 183–190.
- Forsgren, N., Storey, M.-A., Maddila, C., Zimmermann, T., Houck, B., e Butler, J. (2021). The space of developer productivity: There's more to it than you think. *Queue*, 19(1):20–48.
- Freire, S., Rios, N., Mendonça, M., Falessi, D., Seaman, C., Izurieta, C., e Spínola, R. O. (2020). Actions and impediments for technical debt prevention: Results from a global family of industrial surveys. In *SAC '20: Proceedings of the 35th Annual ACM Symposium on Applied Computing*, SAC '20, página 1548–1555, New York, NY, USA. Association for Computing Machinery.

- Gaither, N. e Frazier, G. (2002). *Administração da produção de operações*. Pioneira, 8 edição.
- Garousi, V., Felderer, M., e Mäntylä, M. V. (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*, EASE '16, New York, NY, USA. Association for Computing Machinery.
- Garousi, V., Felderer, M., e Mäntylä, M. V. (2019). Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. *Information and Software Technology*, 106:101–121.
- Gomes, F., Santos, E., Freire, S., Mendes, T. S., Mendonça, M., e Spínola, R. (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):12–1.
- Graziotin, D. e Fagerholm, F. (2019). *Happiness and the Productivity of Software Engineers*, páginas 109–124. Apress, Berkeley, CA.
- Hernández-López, A., Colomo-Palacios, R., e García-Crespo, Á. (2013). Software engineering job productivity—a systematic review. *International Journal of Software Engineering and Knowledge Engineering*, 23(03):387–406.
- Júnior, G. S. e Meira, S. R. (2009). An approach to measure value-based productivity in software projects. In *Ninth International Conference on Quality Software*, páginas 383–389.
- Kamei, F., Pinto, G., Wiese, I., Ribeiro, M., e Soares, S. (2021a). 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)*, ESEM '21, New York, NY, USA. Association for Computing Machinery.
- Kamei, F., Wiese, I., Lima, C., Polato, I., Nepomuceno, V., Ferreira, W., Ribeiro, M., Pena, C., Cartaxo, B., Pinto, G., e Soares, S. (2021b). Grey literature in software engineering: A critical review. *Information and Software Technology*, página 106609.
- Kamei, F. K., Soares, S., e Pinto, G. (2019). The use of grey literature review as evidence for software engineering. In *Anais Estendidos do X Congresso Brasileiro de Software: Teoria e Prática*, páginas 56–63, Porto Alegre, RS, Brasil. SBC.
- Kamei, F. K., Wiese, I., Pinto, G., Ferreira, W., Ribeiro, M., Souza, R., e Soares, S. (2022). Assessing the credibility of grey literature: A study with brazilian software engineering researchers. *Journal of Software Engineering Research and Development*, 10:9:1–9:20.

- Kitchenham, B. e Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report.
- MacCormack, A., Kemerer, C., Cusumano, M., e Crandall, B. (2003). Trade-offs between productivity and quality in selecting software development practices. *IEEE Software*, 20(5):78–79.
- Meyer, A. N., Fritz, T., Murphy, G. C., e Zimmermann, T. (2014). Software developers' perceptions of productivity. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*. ACM.
- Meyer, A. N., Murphy, G. C., Fritz, T., e Zimmermann, T. (2019). *Developers' Diverging Perceptions of Productivity*, páginas 137–146. Apress, Berkeley, CA.
- Montandon, J., Politowski, C., Silva, L., Valente, M., Petrillo, F., e Guéhéneuc, Y.-G. (2020). What skills do it companies look for in new developers? a study with stack overflow jobs. *Information and Software Technology*.
- Murphy-Hill, E. e Wagner, S. (2019). *Software Productivity Through the Lens of Knowledge Work*, páginas 57–65. Apress, Berkeley, CA.
- Oliveira, E., Conte, T., Cristo, M., e Mendes, E. (2016). Software project managers' perceptions of productivity factors. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. ACM.
- Oliveira, E., Conte, T., Cristo, M., e Valentim, N. (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.
- Paez, A. (2017). Gray literature: An important resource in systematic reviews. *Journal of Evidence-Based Medicine*, 10(3):233–240.
- Paiva, E., Barbosa, D., Lima, R., e Albuquerque, A. (2010). Factors that influence the productivity of software developers in a developer view. In Sobh, T. e Elleithy, K., editores, *Innovations in Computing Sciences and Software Engineering*, páginas 99–104, Dordrecht. Springer Netherlands.
- Petersen, K. (2011). Measuring and predicting software productivity: A systematic map and review. *Information and Software Technology*, 53(4):317–343. Special section: Software Engineering track of the 24th Annual Symposium on Applied Computing.
- Ramírez, Y. W. e Nembhard, D. A. (2004). Measuring knowledge worker productivity. *Journal of Intellectual Capital*, 5(4):602–628.

- Ribeiro, M. B., Czekster, R. M., e Webber, T. (2006). Improving productivity of local software development teams in a global software development environment. *Engineering*, 1-2:1-2.
- Roller, M. e Lavrakas, P. (2015). *Applied Qualitative Research Design: A Total Quality Framework Approach*. Guilford Publications.
- Sadowski, C. e Zimmermann, T., editores (2019). *Rethinking Productivity in Software Engineering*. Apress.
- Sampaio, S. C. B., Barros, E. A., de Aquino, G. S., e Silva, M. J. C., e de Lemos Meira, S. R. (2010). A review of productivity factors and strategies on software development. In *2010 Fifth International Conference on Software Engineering Advances*. IEEE.
- Scacchi, W. (1989). Understanding software productivity: A comparative empirical review. Relatório técnico, Institute for Software Research, University of California.
- Sengupta, S. e Haythornthwaite, C. (2020). Learning with comments: An analysis of comments and community on stack overflow. In *Proceedings of the Annual Hawaii International Conference on System Sciences*. Hawaii International Conference on System Sciences.
- Sicilia, M.-A., García-Barriocanal, E., e Sánchez-Alonso, S. (2017). Community Curation in Open Dataset Repositories: Insights from Zenodo. *Procedia Computer Science*, 106:54–60.
- Sommerville, I. (2020). *Engineering software products*, volume 355. Pearson London.
- Strech, D. e Sofaer, N. (2012). How to write a systematic review of reasons. *Journal of Medical Ethics*, 38(2):121–126.
- Suchman, L. (1995). Making work visible. *Commun. ACM*, 38(9):56–64.
- Tahir, A., Dietrich, J., Counsell, S., Licorish, S., e Yamashita, A. (2020). A large scale study on how developers discuss code smells and anti-pattern in stack exchange sites. *Information and Software Technology*, 125:106333.
- Tangen, S. (2005). Demystifying productivity and performance. *International Journal of Productivity and performance management*, 54(1):34–46.
- Trendowicz, A. e Münch, J. (2009). Factors influencing software development productivity—state-of-the-art and industrial experiences. In *Advances in Computers, Volume 77*, volume 77 of *Advances in Computers*, páginas 185–241. Elsevier.
- Wagner, S. e Deissenboeck, F. (2019). *Defining Productivity in Software Engineering*, páginas 29–38. Apress, Berkeley, CA.

- Wagner, S. e Murphy-Hill, E. (2019). Factors that influence productivity: A checklist. In Sadowski, C. e Zimmermann, T., editores, *Rethinking Productivity in Software Engineering*, páginas 69–84. Apress, Berkeley, CA.
- Wagner, S. e Ruhe, G. (2008). A systematic review of productivity factors in software development. *Review Literature And Arts Of The Americas*.
- Yilmaz, M., O'Connor, R. V., e Clarke, P. (2016). Effective social productivity measurements during software development — an empirical study. *International Journal of Software Engineering and Knowledge Engineering*, 26(03):457–490.