Bad Smells of Communication in Multidisciplinary Agile Teams of a Remote Software Project

Cesar Lobato[1], Carlos Portela[1], Maurício Souza[2], Paula Palomino[3], Rodrigo Lisbôa[4], Thiago Cordeiro[5]

[1]Programa de Pós-Graduação em Computação Aplicada (PPCA)
Universidade Federal do Pará (UFPA) - Tucuruí - PA

[2]Programa de Pós-Graduação em Ciência da Computação (PPGCC)
Universidade Federal de Lavras (UFLA) - Lavras - MG

[3]Faculdade de Tecnologia (FATEC) - Matão - SP

[4]Instituto Ciberespacial
Universidade Federal Rural da Amazônia (UFRA) - Belém - PA

[5]Programa de Pós-Graduação em Informática (PPGI)
Universidade Federal de Alagoas (UFAL) - Maceió - AL

cesarlobato27@gmail.com, csp@ufpa.br, mauricio.ronny@ufla.br,

paula.palomino@fatec.sp.gov.br, rodrigo.lisboa@ufra.edu.br, thiago@ic.ufal.br

Abstract. *Remote agile teams often face "communication bad smells" which undermine collaboration and project outcomes. The research investigated these symptoms in a remote project, identifying categories of bad smells and proposing mitigation strategies. A case study with a mixed-methods design was conducted through a survey applied to nine members of a remote project, combining quantitative metrics of frequency and quality with qualitative perceptions. Five critical bad smells were identified (Broken Schedule, Expertise Silos, Information Black Hole, Invisible Decisions, and Process Drift), and practical mitigation strategies (rotating meeting schedules, centralized knowledge bases, and decision logs).*

1. Introduction

Agile software development is characterized by collaborative social interactions and rapid iterative changes throughout the project lifecycle. In this scenario, communication serves as the foundation for effective collaboration, acting as a critical precursor to project success [Weger et al. 2022]. However, the increasing adoption of remote distributed teams, accelerated by the Covid-19 pandemic, has introduced new communication challenges while creating opportunities for innovation, particularly in multidisciplinary agile environments [Garro-Abarca et al. 2021].

These evolving work arrangements have made communication channels, their frequency [Wang et al. 2019], and overall effectiveness [Kostin e Strode 2022] decisive factors for project outcomes [Neeley 2021]. While adaptive strategies like coordinated alignment of efforts, knowledge sharing, and clear goal setting can help address these challenges [Smite et al. 2019], remote environments remain particularly

susceptible to communication bad smells, persistent anti-patterns such as information ambiguity, overload, and expectation misalignment that systematically undermine team effectiveness.

The concept of bad smells originated in software engineering with Kent Beck's work on code quality, where Fowler and Beck [Fowler 2018] identified them as indicators of deeper maintainability issues requiring refactoring. This powerful metaphor has since been productively extended to other domains, including software testing and team dynamics [Van Deursen et al. 2001], where it helps identify and address systemic communication problems in technical collaborations.

Building on this conceptual foundation, our study specifically examines communication symptoms and bad smells in distributed agile teams, with particular attention to their human and social dimensions. Through a case study of multidisciplinary remote teams using mixed-methods questionnaire data, we identify prevalent challenges, analyze characteristic bad smells, and document practitioner-recommended mitigation strategies.

This research directly aligns with the challenges outlined in the GranDSI-BR 2016–2026 [Boscarioli et al. 2017]. Among the highlighted challenges, the need to improve collaboration, coordination, and information sharing in complex and distributed socio-technical systems is particularly relevant. By investigating communication anti-patterns and proposing mitigation strategies, this paper contributes to addressing the broader research agenda that emphasizes human, organizational, and technological integration for advancing Information Systems in Brazil.

In addition to this introduction, Section 2 establishes the research context of remote communication in agile projects; Section 3 details our case study methodology; Section 4 presents findings on identified bad smells and mitigation approaches; and Section 5 concludes with contributions, implications and future works.

2. Background

As highlighted in the Agile Manifesto [Beck et al. 2001], communication is one of the fundamental pillars of agile methodologies, explicitly emerging in the prioritization of "individuals and interactions over processes and tools" [Alzoubi e Gill 2020]. Due to its transversal nature, communication has a significant impact on the results generated, serving as the basis for the development of software products. From the initial stages of product design with the client [Rauf et al. 2023] to its deployment and availability to users [Riungu-Kalliosaari et al. 2016][Diel et al. 2016], effective communication is essential for project success.

In the context of remotely distributed agile teams, the appropriate choice of communication channels, the frequency of interactions, and the quality of communication become critical aspects. The ability to share knowledge efficiently is one of the main challenges faced by these teams [Zahedi e Babar 2014]. Physical distance between team members tends to reduce the frequency of interactions, making it difficult to build mutual trust and, consequently, decreasing the effectiveness of communication over time [Ghobadi e Mathiassen 2016][Costa e França 2020]. In addition, bad smells (anti-patterns) in communication, such as lack of clarity, information overload, and misaligned expectations, can seriously compromise collaboration and productivity.

This paper focuses on a remote project that aims to identify information bottlenecks, plan solutions and optimize the processes of the Brazilian National Textbook Program (PNLD). To this end, the team is developing a platform that allows all actors involved in the program to be trained through tracks. The project lasted 5 years and had a total of 50 participants on the entire team. The team were organized into subteams according to their functions and areas, divided into six stages, as illustrated in Figure 1.

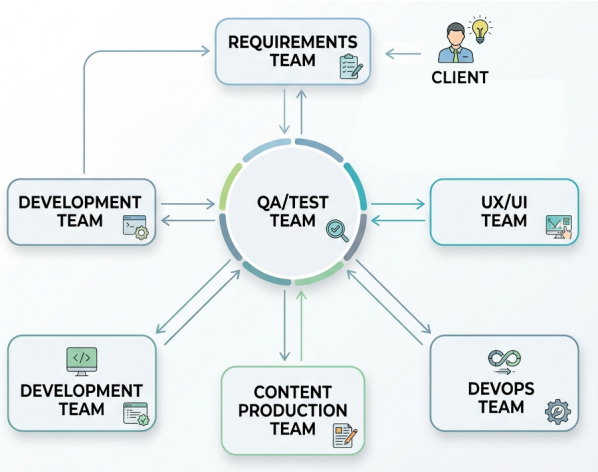

Figure 1. Flow of interaction between the project squads

In Stage 1, the client passes on the demands to the Project Manager and the Requirements Team. In Stage 2, the Requirements Team is responsible for eliciting and documenting demands in Epics and User Stories, which are discussed and prioritized with the Project Manager and Coordination. These Epics and HUs are made available to the UX/UI Team, which creates high-fidelity screen prototypes. In Stage 3, the Project Manager distributes tasks to the Development Team according to the scope of each one (coding, testing, DevOps, content production). In Stage 4, the result (software increment) is made available to the QA/Test Team, which ensures the definition of done: integrated and tested modules (unit, functional and usability). The DevOps Team integrates the code into the infrastructure in Stage 5. Finally, in Stage 6, the manager receives the sprint deliverable and presents it to the client.

3. Case Study

This study was developed aiming to investigate challenges that can significantly impact the effectiveness of communication, as well as possible solutions proposed for the listed challenges and their correlations. In this scenario, the primary means for data collection was an online questionnaire, widely disseminated in the literature with a high degree of quality in identifying aspects and perceptions, as demonstrated by [Šmite et al. 2017][Storey et al. 2016]. The questionnaire was developed with a total of 17 open and closed questions, thus representing both quantitative and qualitative data.

Thus, the following research questions were formulated:

- RQ1. What are the main communication tools used in the project and how do they impact the effectiveness of communication?

- RQ2. How do the frequency and quality of communication vary between different contact points in the project?
- RQ3. What are the main communication challenges faced by the teams and what solutions can be proposed to overcome them?
- RQ4. How can communication be improved to increase team integration and overall project efficiency?

Therefore, in order to combine analysis between quantitative and qualitative data, we adopted the mixed methods approach described by [Creswell e Creswell 2017], specifically the concurrent triangulation design, used as a basis for developing the process of applied data analysis, verification, and triangulation structures.

Thus, moving forward to the expansion of this mixed analysis model, the process in Figure 2 is applied, which considers the use of the Sankey diagram and word cloud as data evaluation techniques.

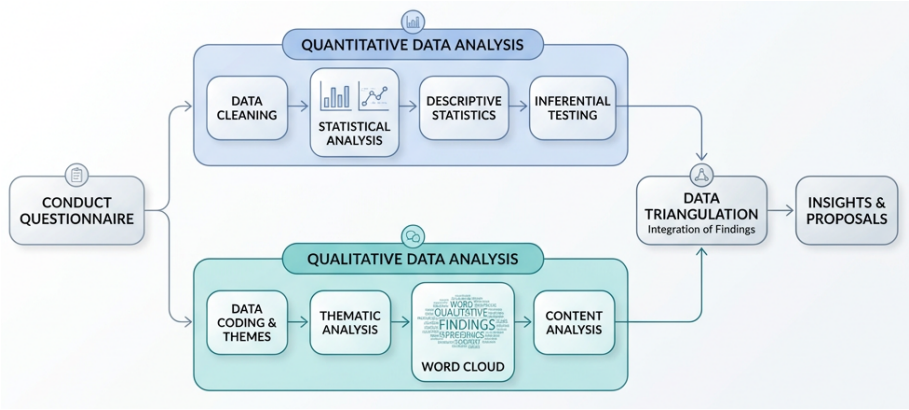

Figure 2. Extended process of quantitative and qualitative data triangulation

This diagram is composed of bars, representing the origin and destination of information, and links showing the flow between these bars [Lupton e Allwood 2017]. The data flow visualization technique was initially proposed and employed by [Riehmann et al. 2005]. This adopted process will help us understand how qualitative data naturally emerge through lived experiences in such a way that it reflects the reality of the studied phenomenon, aiming for a robust and less biased analysis, complemented by insights obtained through the mixed methods triangulation model as proposed by [Creswell e Creswell 2017].

3.1. Research Design

We developed a focused questionnaire aiming to address each focus group participating in the software project. In this scenario, all teams were notified via email. The information was collected according to the privacy and confidentiality policies of the project they were involved in, where access to the questionnaire was strictly restricted to institutional emails. The questionnaire addressed four main evaluative topics on communication, with the first two being quantitative, consisting exclusively of closed questions, and the following two being open questions, yielding qualitative data, as shown Table 1.

Table 1. Questionnaire

ID	Question
Q1	**In which project area are you working?** ( ) Requirements ( ) UX/UI ( ) Development ( ) Testing/QA ( ) DevOps ( ) Content Production ( ) Management ( ) Coordination
Q2	**What are the main tools you use in project to communicate?** ( ) Jira ( ) Confluence ( ) Discord ( ) WhatsApp ( ) Google Meet ( ) Others
Q3	**What are the main communication points you work with?** ( ) Communication with governance ( ) Communication with users ( ) Communication with leadership ( ) Communication between teams ( ) Communication between management ( ) Communication with coordination
Q4	**For the communication channels above, the evaluations on the frequency of communication followed the scale below:** ( ) 1 - None (I do not speak) ( ) 2 - Low (Rarely) ( ) 3 - Sufficient (Sometimes) ( ) 4 - Good (Almost Always) ( ) 5 - Excellent (Daily)
Q5	**On the other hand, the evaluations of communication quality followed the scale below:** ( ) 1 - Not Applicable (No interaction, thus not possible to assess quality) ( ) 2 - Very Low (Ineffective with major difficulties in transmitting or receiving information) ( ) 3 - Low (Frequent misunderstandings and failures in information transmission) ( ) 4 - Sufficient (Some areas for improvement, but generally meets its purpose) ( ) 5 - Good (Clear and effective, with rare issues or loss of information/misunderstandings) ( ) 6 - Excellent (Exceptional, highly effective, clear and precise, facilitating interactions and mutual understanding)
Q6	**In your opinion, what are the main challenges faced in communication currently?**
Q7	**What actions do you suggest to improve communication in the project?**

Finally, the analysis of the collected data will provide valuable insights for future training and development actions, contributing to a more collaborative and productive work environment.

3.2. Results

The study examined communication in key aspects: Tools, Communication Channels, Frequency, and Quality. In this scenario, a total of 9 different teams presented their responses to the questionnaire, being: Requirements (4.3%), UX/UI (4.3%), Development (21.7%), Testing/QA (30.4%), DevOps (8.7%), Digital Transformation (13%), Management (8.7%), Coordination (4.3%), and Administrative Support (4.3%).

Regarding the first main topic studied, it was found that WhatsApp is the most widely used tool within the project, being used by 100% of the professionals. Participants were also asked about their main channels of interaction within the project. The results for communication channels highlighted the most frequent point. Communication between teams was mentioned (78.3%), followed by communication between leadership (47%). Thus, suggesting a predominantly horizontal organizational structure in which participants tend to emphasize intra-team interaction more.

Regarding the last two study topics, participants were instructed to evaluate each communication point based on their personal experiences and perceptions. Data on the frequency and quality of communication were defined on scales ranging from 1 to 6, specific to each, presenting their nuances and unique characteristics in relation to the aspect being assessed in the study. The results demonstrate that the frequency and quality of communication vary between average and Good for frequency among Teams (4.2), Leadership (3.8), and Management (3.2). Consequently, the other communication channels, Coordination, Governance, and Users, were close to the overall average of 2.97. The quality of communication, in turn, remained at the same average level as frequency, with communication quality among Teams averaging (4.5), Leadership (4.3), and Management (3.7). The communication channels, Coordination, Governance, and Users, were close to the overall average of 3.33.

Regarding qualitative data, two open-ended questions were defined where participants were asked to describe the main challenges faced and possible solutions. To validate the results, the Sankey diagram in Figure 3 that analyzes correlations between listed challenges and proposed solutions (according to those interviewed).

4. Bad Smells of Communication

Identifying bad smells in communication in remote software development teams differs from traditional models proposed for code smells. While code smells are often detected based on metrics, heuristic rules, machine learning, or optimization (Sharma & Spinellis, 2018), identifying bad smells in communication requires a history- and context-based approach. History-based methods, as proposed by Palomba et al. (2015a), use information about the project's evolution and changes over time to infer the presence of smells. This approach is particularly relevant for the study of bad smells in communication, where the analysis of interaction patterns, information flow, and decisions made over time can reveal underlying problems.

The results of the analysis of data from interviews with case study members reveal critical symptoms in the communicative dynamics. The authors searched the literature for reports of similar symptoms, allowing them to identify correlations and define the presence of bad smells. The bad smells were named with the help of the generative Artificial Intelligence of ChatGPT and Deepseek, as shown in Table 2.

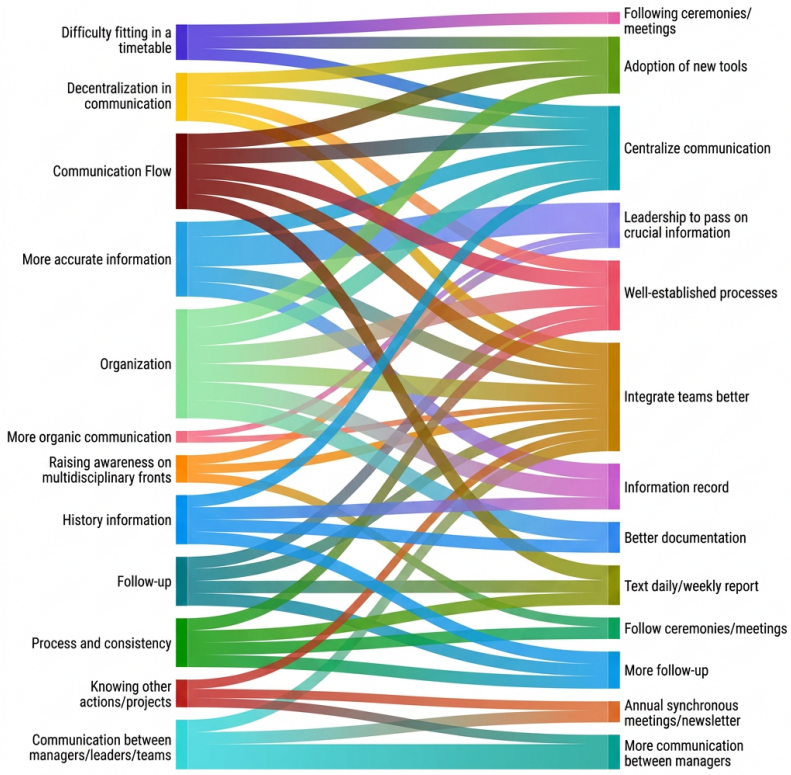

Figure 3. Sankey Diagram

Broken Schedule occurs when teams fail to synchronize schedules due to overlapping personal commitments, or lack of planning, resulting in project delays.

In **Expertise Silos**, knowledge is isolated within individuals or sub-teams, creating communication barriers and making collaboration difficult. This leads to a lack of information sharing and over-reliance on a few people.

Information Black Hole is when the lack of centralization and organization of information causes important data to be lost or inaccessible. This makes it difficult to monitor the project and obtain detailed information.

Invisible Decisions occur when important decisions are not documented or communicated properly, leading to misunderstandings or loss of historical context.

Finally, **Process Drift** occurs when a team does not follow defined processes consistently, resulting in inconsistencies and repeated mistakes. This occurs due to a lack of adherence or a lack of clear documentation.

4.1. Mitigation of Bad Smells

Considering the researches identified in the literature, highlighted in Table 2, it was possible to identify approaches to mitigate bad smells in our projejct. To mitigate **Broken Schedule** issues caused by poor planning, the team should implement rotating meeting times to distribute inconvenience fairly, establish core overlap hours for synchronous collaboration, and leverage scheduling tools like Calendly. Recording critical meetings and creating team agreements on response expectations can further prevent delays.

Table 2. Bad Smells of Communication Identified

Symptoms	Bad Smell	References
-Difficulty in defining a common schedule. -Difficulty in establishing more organic communication.	Broken Schedule	[Olson e Olson 2000] [Espinosa e Carmel 2003]
-Difficulty in communicating between hierarchical levels. -Lack of awareness of the multidisciplinary team.	Expertise Silos	[Tiwana 2004] [Dingsøyr et al. 2012] [Desouza e Evaristo 2004] [Bjørnson e Dingsøyr 2008] [Herbsleb e Moitra 2001]
-Decentralization in communication. -Lack of knowledge of the flow of information. -Difficulty in monitoring the project. -Difficulty in obtaining detailed information.	Information Black Hole	[Murphy-Hill et al. 2019] [Begel et al. 2009] [Meyer et al. 2014] [Storey et al. 2017] [Cataldo e Herbsleb 2012]
-Disorganization of project information. -Lack of decision recording.	Invisible Decisions	[Tiwana e Keil 2004] [Jansen e Bosch 2005] [Kruchten et al. 2006]
-Difficulty in maintaining a consistent process. -Lack of knowledge of standards of other projects.	Process Drift	[Kruchten 2010] [Boehm e Turner 2005] [Lindvall et al. 2004]

For **Expertise Silos**, regular knowledge-sharing sessions through lightning talks or documentation sprints help break down barriers, while remote pair programming across disciplines and maintaining a centralized knowledge base with role-specific guides ensure information democratization.

Information Black Holes can be addressed by designating a single well-structured page in Confluence with clear ownership, implementing documentation standards with templates and tagging systems, and preferring transparent communication channels like Discord over private messages in WhatsApp. Automating meeting minutes and conducting weekly documentation health checks maintain information accessibility.

To combat **Invisible Decisions**, the team should maintain decision logs with contextual details for technical choices, and implement a "no decision without a ticket" policy. Visualizing decisions on team dashboards and ending meetings with decision recaps posted in shared channels reinforce transparency.

Finally, **Process Drift** requires visual process maps (e.g., BPMN flows) and documentation (e.g., page in Confluence) for clarity, quarterly process retrospectives for continuous improvement, and assigned process champions to guide adherence. For all smells, starting with high-priority issues, tracking metrics like documentation coverage, and using tools like Jira for async updates and Confluence for centralized documentation ensures sustainable improvement. These strategies collectively enhance communication

efficiency while respecting the human and logistical challenges of remote agile work.

In addition to specific strategies, it is essential to adopt a continuous approach to monitoring and improvement, recognizing that the mitigation of communication bad smells is not a one-time event but a dynamic process. Defining key indicators, such as average response time, documentation update rate, and frequency of collective reviews, makes it possible to track the effectiveness of the implemented actions. Moreover, leadership plays a decisive role in fostering a culture of transparency and collaboration, ensuring that the established practices are not merely technical tools but an integral part of the team's daily routine. This combination of objective metrics and human engagement helps reduce the recurrence of communication anti-patterns and promotes sustainable long-term improvement.

5. Final Remarks

This paper explored communication challenges in multidisciplinary agile teams working remotely, identifying five key bad smells that hinder effective collaboration: Broken Schedule, Expertise Silos, Information Black Hole, Invisible Decisions, and Process Drift. Through a mixed-methods approach combining quantitative surveys and qualitative analysis, we uncovered how these communication anti-patterns emerge in practice and proposed concrete strategies to address them. Our findings highlight that while remote agile teams face significant communication barriers, these can be mitigated through thoughtful interventions targeting both technical and human factors.

The research makes several important contributions to understanding and improving communication in distributed agile environments. First, we empirically validated a taxonomy of communication bad smells, extending the established concept of code smells to the human and social dimensions of software development. Second, we developed practical, context-aware mitigation strategies tailored to remote work constraints, such as rotating meeting times to accommodate time zones and centralized knowledge bases to break down expertise silos. Third, our methodological approach demonstrated the value of combining quantitative communication metrics with qualitative insights to provide a comprehensive view of team dynamics.

These findings have important implications for agile practice. Teams should prioritize documentation and transparency, implementing tools like decision logs and process maps while also fostering a culture of knowledge sharing through regular cross-functional interactions. Leadership plays a critical role in modeling and reinforcing good communication practices, particularly in remote settings where informal interactions are limited. Our results suggest that addressing communication smells requires both technological solutions (like improved collaboration platforms) and social interventions (such as structured feedback mechanisms).

While this paper provides valuable insights, certain limitations point to directions for future research. The single-case study design suggests a need for broader validation across different organizational contexts and team structures. Longitudinal research could help determine whether our proposed mitigation strategies lead to sustained improvements over time. Additionally, emerging technologies like AI-powered communication analysis may offer new opportunities for automatically detecting and addressing communication smells before they significantly impact team performance.

Acknowledgments

We appreciate the financial support from the Ministry of Education through TED 10523, implemented by NEES at the Federal University of Alagoas. Additionally, we thank OpenIA for supporting the Generative AI GPT-4o (Plus version) in categorizing and naming bad smells and grammatically reviewing the final text of this paper. The authors critically reviewed all AI-generated content and assume full responsibility for the final version of the paper, including its accuracy, originality, and the conclusions presented.

References

Alzoubi, Y. I. e Gill, A. Q. (2020). An empirical investigation of geographically distributed agile development: The agile enterprise architecture is a communication enabler. *IEEE Access*, 8:80269–80289.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., et al. (2001). The agile manifesto.

Begel, A., Nagappan, N., Poile, C., e Layman, L. (2009). Coordination in large-scale software teams. In *2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering*, pages 1–7. IEEE.

Bjørnson, F. O. e Dingsøyr, T. (2008). Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used. *Information and software technology*, 50(11):1055–1068.

Boehm, B. e Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *IEEE Software*, 22(5):30–39.

Boscarioli, C., Araujo, R. M., e de S. P. Maciel, R. (2017). *GranDSI-BR – Grand Research Challenges in Information Systems in Brazil 2016–2026*. Special Committee on Information Systems (CE-SI), Brazilian Computer Society (SBC).

Cataldo, M. e Herbsleb, J. D. (2012). Coordination breakdowns and their impact on development productivity and software failures. *IEEE Transactions on Software Engineering*, 39(3):343–360.

Costa, V. G. e França, C. (2020). How office layouts influence software development? In *Proceedings of the XXXIV Brazilian Symposium on Software Engineering*, pages 173–182.

Creswell, J. W. e Creswell, J. D. (2017). *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications.

Desouza, K. C. e Evaristo, J. R. (2004). Managing knowledge in distributed projects. *Communications of the ACM*, 47(4):87–91.

Diel, E., Marczak, S., e Cruzes, D. S. (2016). Communication challenges and strategies in distributed devops. In *2016 IEEE 11th International Conference on Global Software Engineering (ICGSE)*, pages 24–28. IEEE.

Dingsøyr, T., Nerur, S., Balijepally, V., e Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development.

Espinosa, J. A. e Carmel, E. (2003). The impact of time separation on coordination in global software teams: a conceptual foundation. *Software Process: Improvement and Practice*, 8(4):249–266.

Fowler, M. (2018). *Refactoring: improving the design of existing code*. Addison-Wesley Professional.

Garro-Abarca, V., Palos-Sanchez, P., e Aguayo-Camacho, M. (2021). Virtual teams in times of pandemic: Factors that influence performance. *Frontiers in psychology*, 12:624637.

Ghobadi, S. e Mathiassen, L. (2016). A model for assessing and mitigating knowledge sharing risks in agile software development. *Information Systems Journal*, pages 1–33.

Herbsleb, J. D. e Moitra, D. (2001). Global software development. *IEEE software*, 18(2):16–20.

Jansen, A. e Bosch, J. (2005). Software architecture as a set of architectural design decisions. In *5th Working IEEE/IFIP Conference on Software Architecture (WICSA'05)*, pages 109–120. IEEE.

Kostin, D. e Strode, D. (2022). Effective communication in globally distributed scrum teams. *ACIS 2022 Proceedings*.

Kruchten, P. (2010). Software architecture and agile software development: a clash of two cultures? In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 2*, pages 497–498.

Kruchten, P., Lago, P., e Van Vliet, H. (2006). Building up and reasoning about architectural knowledge. In *International conference on the quality of software architectures*, pages 43–58. Springer.

Lindvall, M., Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D., May, J., e Kahkonen, T. (2004). Agile software development in large organizations. *Computer*, 37(12):26–34.

Lupton, R. e Allwood, J. (2017). Hybrid sankey diagrams: Visual analysis of multidimensional data for understanding resource use. *Resources, Conservation and Recycling*, 124:141–151.

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*, FSE 2014, pages 19–29. ACM.

Murphy-Hill, E., Jaspan, C., Sadowski, C., Shepherd, D., Phillips, M., Winter, C., Knight, A., Smith, E., e Jorde, M. (2019). What predicts software developers' productivity? *IEEE Transactions on Software Engineering*, 47(3):582–594.

Neeley, T. (2021). *Remote work revolution: Succeeding from anywhere*. Harper Business London, UK:.

Olson, G. M. e Olson, J. S. (2000). Distance matters. *Human–computer interaction*, 15(2-3):139–178.

Rauf, M. A., Bibi, S., Ali, S., AlSaedi, T., Ur Rehman, S., Mahmood, K., e Kundi, M. (2023). A cost effective communication model for requirements elicitation in global software development. *Scientific Reports*, 13(1):18730.

Riehmann, P., Hanfler, M., e Froehlich, B. (2005). Interactive sankey diagrams. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pages 233–240. IEEE.

Riungu-Kalliosaari, L., Mäkinen, S., Lwakatare, L. E., Tiihonen, J., e Männistö, T. (2016). Devops adoption benefits and challenges in practice: A case study. In *Product-Focused Software Process Improvement: 17th International Conference, PROFES 2016, Trondheim, Norway, November 22-24, 2016, Proceedings 17*, pages 590–597. Springer.

Smite, D., Moe, N. B., Levinta, G., e Floryan, M. (2019). Spotify guilds: how to succeed with knowledge sharing in large-scale agile organizations. *Ieee Software*, 36(2):51–57.

Šmite, D., Moe, N. B., Šāblis, A., e Wohlin, C. (2017). Software teams and their knowledge networks in large-scale software development. *Information and Software Technology*, 86:71–86.

Storey, M.-A., Zagalsky, A., Figueira Filho, F., Singer, L., e German, D. M. (2016). How social and communication channels shape and challenge a participatory culture in software development. *IEEE Transactions on Software Engineering*, 43(2):185–204.

Storey, M.-A., Zagalsky, A., Filho, F. F., Singer, L., e German, D. M. (2017). How software developers use Slack channels to foster collaboration. In *Proceedings of the 20th ACM Conference on Computer-Supported Cooperative Work and Social Computing*, CSCW '17, pages 2511–2525. ACM.

Tiwana, A. (2004). An empirical study of the effect of knowledge integration on software development performance. *Information and Software Technology*, 46(13):899–906.

Tiwana, A. e Keil, M. (2004). The one-minute risk assessment tool. *Communications of the ACM*, 47(11):73–77.

Van Deursen, A., Moonen, L., Van Den Bergh, A., e Kok, G. (2001). Refactoring test code. In *Proceedings of the 2nd international conference on extreme programming and flexible processes in software engineering (XP2001)*, pages 92–95. Citeseer.

Wang, Y., Graziotin, D., Kriso, S., e Wagner, S. (2019). Communication channels in safety analysis: An industrial exploratory case study. *Journal of systems and software*, 153:135–151.

Weger, K., Leder, S., Mesmer, B., Menon, V., e Schaub, H. (2022). How effectively do we communicate? an analysis of team reflexivity in transition and action phases of team collaboration. *IEEE Transactions on Professional Communication*, 65(3):392–410.

Zahedi, M. e Babar, M. A. (2014). Knowledge sharing for common understanding of technical specifications through artifactual culture. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, pages 1–10.