Communication Challenges and Practices in AI-based Software Systems: An Exploratory Study

Sheila Maria Mendes Paiva
Federal University of Campina Grande
Campina Grande, Brazil
sheila.paiva@copin.ufcg.edu.br

Arthur Alves
Federal University of Campina Grande
Campina Grande, Brazil
arthur.alves@ccc.ufcg.edu.br

Narallynne Araújo
Federal University of Campina Grande
Campina Grande, Brazil
narallynne@copin.ufcg.edu.br

Tiago Massoni
Federal University of Campina Grande
Campina Grande, Brazil
massoni@dsc.ufcg.edu.br

Franklin Ramalho
Federal University of Campina Grande
Campina Grande, Brazil
franklin@computacao.ufcg.edu.br

ABSTRACT

The integration of AI-based systems into software systems poses technical and organizational challenges, especially regarding communication between multidisciplinary teams. This qualitative study, grounded in the Socio-Technical Grounded Theory, investigates communication practices during the development and integration of AI-based systems. Conducted in two cycles with a total of 15 semi-structured interviews, the study identifies five key communication challenges: lack of shared technical vocabulary, misaligned expectations and priorities, unstructured communication practices, and inadequate use of tools. These issues often lead to misunderstandings, rework, and miscoordination across teams. To address them, participants recommended strategies such as shared glossaries, regular alignment meetings, and standardized collaborative tools. These findings contribute to the understanding of socio-technical dynamics in ML-based software projects and offer practical recommendations to improve integration workflows and guide future Software Engineering practices.

KEYWORDS

AI-based system integration, Software Engineering, Team Collaboration, Communication Practices, ML Project Challenges

1 Introduction

The integration of AI-based systems into traditional software systems has become increasingly common, raising specific challenges for Software Engineering (SE). The rapid evolution of Machine Learning (ML) complicates the adoption of established engineering practices, with studies highlighting both technical and organizational obstacles [6, 13]. While some argue that ML systems require new practices, others claim that rigorously applying traditional ones may suffice [7]. Still, the added uncertainty and complexity of ML components demand context-specific strategies.

Ozkaya [10] notes that AI systems shift engineering priorities toward attributes like explainability and verifiability, requiring changes in design and evaluation strategies. These changes reflect not only technical demands, but also the need to manage the propagation of changes and the uncertainty intrinsic to ML models. Consequently, software teams must adapt their development processes to accommodate probabilistic reasoning, data dependencies, and model retraining.

Beyond technical issues, human and collaborative aspects are critical. Prior work reports challenges in interdisciplinary collaboration, including vocabulary gaps, unclear responsibilities, and misaligned priorities [2, 8, 9, 12]. While these studies, including Piorkowski et al. [12], focus broadly on organizational strategies and educational efforts such as cross-training, they often do not explicitly address day-to-day communication practices during technical integration. This study differentiates itself by explicitly investigating these everyday communication dynamics. As Kästner [7] points out, "It truly requires data scientists, software engineers, and others to work together, understand each other, and communicate effectively."

Understanding how communication occurs between data scientists and software engineers is key to improving ML system integration. This research investigates Software Engineering practices in this context, focusing on socio-technical interactions. Grounded in the Socio-Technical Grounded Theory (STGT) for Software Engineering [5], the study was conducted in two interview-based cycles. The first, with eight participants, identified initial categories of communication challenges. These guided the second cycle, with seven additional interviews focused on refining the findings.

Our results highlight five recurring challenges: lack of shared technical vocabulary, misaligned expectations and priorities, unstructured communication practices, and inadequate collaborative tools. This study contributes by offering practical recommendations such as shared glossaries and communication rituals—to support more effective integration and guide future SE practices for ML-based systems.

2 Background

This study is grounded in the STGT, which guides the qualitative analysis of data in contexts characterized by the interaction between technical and social factors. This section outlines the theoretical foundations that support the adopted approach.

2.1 Socio-Technical Grounded Theory for Software Engineering

The STGT, proposed by Hoda [5], extends Classic Grounded Theory to socio-technical contexts in software engineering. It recognizes that practices and decisions in this domain emerge from the interplay between human agents and technical artifacts. STGT involves collecting and analyzing empirical data through qualitative methods, such as interviews or observations, to build theory grounded in participants' experiences. Its analytical process is iterative and based on immersion in the data, constant comparison, and the progressive construction of concepts and categories. Open coding transforms raw textual data into meaningful codes, and a socio-technical perspective ensures that both human and technical dimensions are considered during analysis.

2.1.1 Coding Process. The coding process in STGT unfolds in iterative stages. It begins with open coding, where data is examined line by line to identify emerging concepts in an exploratory manner. This is followed by constant comparison, in which concepts are systematically compared and refined into explanatory categories. Theoretical integration then organizes these categories into hierarchical structures, capturing both technical and social elements of the phenomenon. Visual representations, such as diagrams, may support this process by clarifying the relationships between codes and categories. For instance, codes related to confusion over ML terminology led to the identification of issues in interdisciplinary understanding.

2.1.2 Formation of Theoretical Categories. Theoretical categories represent key phenomena identified through coding and comparison. They are built iteratively, grounded in participant accounts and sensitive to organizational and technical contexts. Rather than applying a predefined model, STGT encourages flexible theorization based on patterns observed in the data. This process is enriched through the writing of analytical memos, creation of diagrams, and ongoing refinement. For example, the codes *"Lack of Understanding of Terms"* and *"Misalignment of Technical Vocabulary"* contributed to the broader category *"Challenges in Interdisciplinary Communication"*.

3 Research Method

This research adopts a qualitative approach grounded in the STGT, which is appropriate for investigating phenomena in contexts where social and technical factors are intertwined. We conducted the study in two sequential and complementary stages, corresponding to the basic stage of the STGT method—focused on initial data collection and analysis to support the construction of conceptual categories. Although both stages contribute to the research, the second cycle constitutes the central focus of this study, as it deepens the investigation into communication practices and challenges.

The Figure 1 presents an overview of the methodological process adopted in the basic stage of the STGT approach. Main steps are represented as rectangles with their respective outputs, including definitions, memos, interview transcripts, codes, and refinements of the interview guide and conceptual categories. This visual summary helps to clarify the iterative nature of the process. The research was approved by the Research Ethics Committee of the Anonymous University, and all participants signed an informed consent form, in accordance with ethical guidelines.

All research artifacts referenced in this paper—including interview guides, anonymized transcripts, coding structures, and analysis diagrams—are available in a publicly accessible repository [11].

3.1 Cycle 1: Exploratory Study

The first cycle aimed to map engineering practices and challenges faced in the development of software systems with ML components. The emphasis was on identifying practical experiences, adaptation strategies, and obstacles encountered by multidisciplinary teams.

3.1.1 Data Collection. Eight semi-structured interviews were conducted with professionals working in the field, including software engineers and data scientists with experience in ML-based projects. The interviews addressed topics such as the context of model application, engineering practices used, challenges faced, and solutions adopted by the teams. All interviews were conducted remotely, recorded with consent, and transcribed for analysis. Participants were recruited through professional contacts of the researchers and snowball sampling, ensuring that all selected individuals had prior experience in projects involving AI-based systems.

The participants of Cycle 1 included four data scientists, three software developers, and one quality assurance engineer, with professional experience ranging from 4 to over 14 years. Their academic backgrounds varied from undergraduate to doctoral level, and their roles combined technical and managerial activities. Participants were affiliated with different teams and organizations, ensuring diversity in experiences with AI-based system integration.

3.1.2 Data Analysis. The analysis followed the basic stage of STGT, conducted iteratively through two main phases: (i) open coding, identifying emerging concepts line by line, and (ii) theoretical integration, systematically grouping and refining concepts into explanatory categories using the constant comparison technique. The process was conducted with the support of MAXQDA[1], aided by analytical memos and iterative revisions.

The exploratory analysis indicated that communication failures among interdisciplinary teams constitute one of the main barriers to integrating AI-based systems into software systems. The key challenges identified include the absence of a shared technical vocabulary, misaligned expectations and priorities between teams, lack of structured communication practices, and inefficient use of collaborative tools. These findings provided the empirical foundation for the in-depth investigation carried out in the subsequent stage of the study. Figure 2, developed based on the exploratory phase, summarizes the main categories and their interrelations.

[1] MAXQDA: https://www.maxqda.com/

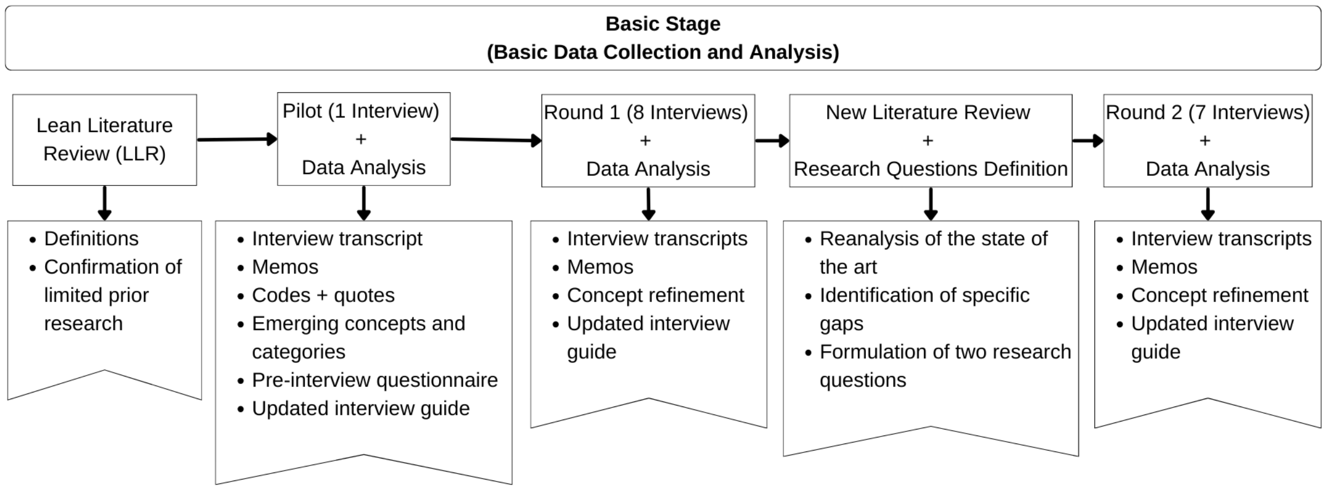

Figure 1: Socio-Technical Grounded Theory method applied in basic stages.

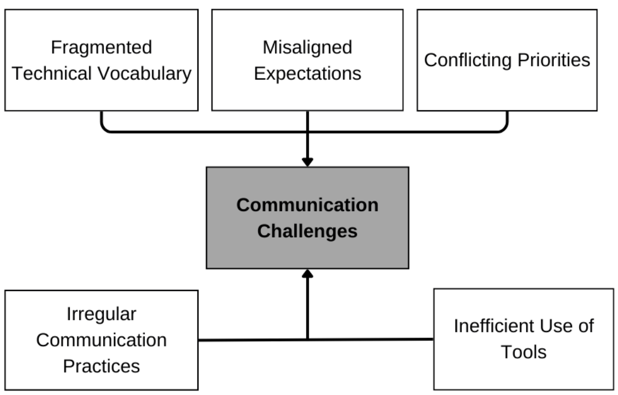

Figure 2: Main communication challenges in the integration of AI-based systems into software systems.

3.2 Cycle 2: In-Depth Study

The second cycle sought to investigate, in greater detail, the communication aspects involved in the integration of AI-based systems. The focus was on understanding the challenges faced in communication between specialists from different fields and the practices that support effective collaboration in these contexts.

The transition from the first to the second cycle was guided by preliminary findings from Cycle 1, which revealed that communication barriers were among the most recurrent and impactful challenges faced by teams integrating AI-based systems. Although the initial focus was on software engineering practices more broadly, the prominence of communication-related issues across interviews highlighted the need for a more focused investigation. Therefore, Cycle 2 was designed to deepen the analysis specifically on interdisciplinary communication, allowing a more refined understanding of how these challenges affect the integration process.

3.2.1 Research Questions. The second cycle was guided by the following research questions:

- **RQ1:** What are the main communication challenges faced between AI-based system specialists and software development specialists during the integration of AI-based systems into traditional systems?
- **RQ2:** What practices have been adopted or recommended to improve communication and support the integration of AI-based systems into traditional systems?

3.2.2 Data Collection. Seven semi-structured interviews were conducted with professionals directly involved in the integration of AI-based systems. Participants were selected by convenience, ensuring a diversity of roles, experiences, and organizational contexts. The interviews were conducted remotely, lasting between 30 and 45 minutes, recorded with consent, and transcribed using TurboScribe, an AI-based transcription service[2], followed by manual review to ensure accuracy.

The interview guide was built upon two main sources: the literature on interdisciplinary collaboration in ML projects and the empirical categories identified during the first cycle. The thematic blocks (i) shared technical vocabulary, (ii) alignment of expectations, and (iii) alignment of priorities were initially identified as emerging categories in the Cycle 1 analysis. In contrast, blocks (iv) structured communication practices and (v) communication tools were inspired by prior literature reviews on coordination strategies and the use of collaborative tools in software engineering teams [3, 8]. Block (vi) participant recommendations was included to capture spontaneous suggestions from professionals. This combination allowed for a deeper exploration of key topics in interdisciplinary communication.

The participants of Cycle 2 included four software engineers, three data scientists, and one engineering manager, with 4 to 10 years of experience. Most held graduate degrees (master's level) in computer science, statistics, or systems modeling. All participants were involved in interdisciplinary AI-software projects across different teams and companies.

[2]https://www.turboscribe.ai

3.2.3 Data Analysis. The analysis was conducted iteratively using MAXQDA, applying open coding and theoretical integration. Coding was performed collaboratively by two researchers, involving regular discussions to resolve discrepancies and ensure interpretive validity. Analytical memos documented preliminary insights and supported the constant comparison process, essential in STGT.

The analytical process was conducted iteratively until reaching theoretical saturation, progressively refining and expanding existing categories. Following STGT principles, the categories presented in this paper represent the consolidated findings from the two cycles conducted.

The emerging categories reveal recurring barriers and effective communication practices, forming the groundwork for theoretical and practical recommendations aimed at enhancing the sociotechnical integration of AI-based systems into software systems.

4 Results

This section presents the partial results obtained from the qualitative analysis conducted in two complementary cycles. The first cycle, of an exploratory nature, involved eight interviews with professionals from the fields of Software Engineering and Data Science, aiming to map the practices adopted and challenges faced in integrating AI-based systems into traditional software systems. The findings of this cycle revealed, among other aspects, that communication failures between interdisciplinary teams are among the main barriers to the effective integration of AI-based systems into software systems, providing both empirical and theoretical support for the in-depth exploration carried out in the subsequent cycle.

In the second cycle, which constitutes the main focus of this study, seven new interviews were conducted with a specific focus on the communication aspects emerging from the first analysis. The results of this cycle are presented in the following subsections, organized to highlight both the challenges faced and the practices adopted or recommended by the teams in integrating AI-based systems into software systems.

4.1 RQ1 — Communication Challenges in Integrating AI-based Systems

The analysis reinforced the presence of the five dimensions previously identified: the absence of a shared technical vocabulary, misalignment of expectations and priorities, lack of structured communication practices, and inadequate use of collaborative tools. These categories, initially outlined in the first cycle, were confirmed and enriched with new empirical examples, providing further detail on the challenges faced by the teams. Each of these dimensions is presented and illustrated below with excerpts from the interviews.

Shared Technical Vocabulary. The absence of a common language between ML and Software Engineering (SE) teams was widely identified as a source of noise and rework. Participant 13 reported a misunderstanding about how ML models work: *"The software team thought that the model had to be trained every time a prediction was made. They didn't know the model was already trained."*. Participant 10 added an example of incompatibility between expectations and outputs: *"The developer created an API expecting a data format that was incompatible with what the model produced. This led to a conflict over data formats."* These reports reinforce the findings from the first cycle, which already indicated the absence of glossaries or terminological alignment as barriers to mutual understanding.

Alignment of Expectations. Divergences in scope, deadlines, and expected results were frequent. Participant 13 stated that there was confusion between deployment and validation: *"The manager thought that putting the model into production was just about deploying it. But we were still testing and validating it. He didn't understand that."* Participant 11 emphasized the need to explain the process to stakeholders: *"We had to sit down with the product team and explain that the value of the model wasn't immediate. There was a whole process until it reached that point."*

Alignment of Priorities. Conflicts over what should be prioritized also hindered integration. Participant 15 noted that infrastructure priorities often overruled ML work: *"The product team gave top priority to bug fixes. No one wanted to stop to deal with technical debt in data."* Participant 14 described a recurring dilemma in task prioritization: *"We constantly debated what was more important: improving the model or resolving customer tickets."*

Lack of Structured Communication Practices. Informality or the absence of systematic communication processes contributed to frequent misalignments. Participant 9 reported a lack of communication between scheduled meetings: *"We only communicated during the bi-weekly meeting. Until then, each person made decisions on their own."* Participant 10 described the surprise factor in uncoordinated efforts: *"Many times, I showed up at the meeting and discovered that something had been done that no one had mentioned before."*

Inadequate Communication Tools. Disorganization of channels and the absence of a standardized platform contributed to information dispersion. Participant 10 commented on the inefficiency of task tracking in spreadsheets: *"Managing tasks in spreadsheets was very confusing. After we moved to Jira, it was much easier to track."* Participant 11 emphasized how centralized documentation helped team alignment: *"We used Notion to write everything before the meetings. It really helped align things and avoid losing information."*

4.2 RQ2 — Practices Adopted or Recommended to Improve Communication and Integration

Based on the reported challenges, participants shared practices that were effective in mitigating communication problems. The emerging strategies align with the identified dimensions and reinforce the importance of institutionalizing collaborative mechanisms. The following paragraphs describe these practices, grouped according to the five dimensions previously identified.

Shared Glossaries and Leveling Sessions. Participant 11 described an initiative to align terminology: *"We created a shared glossary with the most common ML terms and how they applied to the system."* This practice helped develop a common vocabulary among the teams, promoting conceptual alignment and avoiding misunderstandings.

Alignment Rituals. Participant 15 emphasized the positive impact of structured meetings: *"When we started having weekly meetings with minutes and decision points, communication improved."* Regular meetings, with agendas and documented decisions, were cited as significantly improving communication.

Standardized Use of Collaborative Tools. Participant 10 reported improvements after switching tools: *"We switched from spreadsheets to Jira."* Participant 11 highlighted the usefulness of documentation tools for team alignment: *"We used Notion to write everything before the meetings. It really helped align things and avoid losing information."*

Negotiation of Priorities. Some teams began adopting formal forums to negotiate priorities. Although not all participants reported the existence of structured mechanisms, those who did observed improvements in clarity and productivity.

The recommendations derived from these empirical findings are summarized in Table 1, organized based on the five central categories identified.

Table 1: Practical recommendations based on the analyzed categories

Category	Recommendation
Shared technical vocabulary	Develop collaborative glossaries and promote technical leveling sessions
Expectation alignment	Establish regular alignment meetings focusing on deliverables and scope
Priority alignment	Create formal mechanisms for negotiating priorities across domains
Structured communication practices	Institute communication rituals, such as meetings with decision logs
Communication tools	Adopt and standardize the use of collaborative and tracking platforms

5 Discussion

The empirical findings from this preliminary qualitative study provide answers to the proposed research questions, RQ1 and RQ2, reaffirming previously identified issues such as communication challenges, misaligned expectations, and conflicting priorities among interdisciplinary teams [9]. Our empirical data enrich these discussions with concrete examples, enabling specific and actionable practices to enhance interdisciplinary dialogue. Furthermore, we complement recommendations from other works, such as Piorkowski et al.'s [12] advocacy for cross-training, by explicitly focusing on immediate, day-to-day communication practices as complements to broader organizational strategies. Ultimately, effective integration requires not only technical coordination but also a fundamental reevaluation of project management methodologies to accommodate uncertainties inherent to ML components.

The persistence of a fragmented technical vocabulary emerged as a primary communication barrier, directly impacting integration efficiency and quality. As exemplified by participant observations regarding model training misconceptions and incompatible data formats, terminological misunderstandings lead to rework and conflict. Unlike previous studies, our research extends this by proposing practical communication strategies explicitly oriented toward technical integration, emphasizing the need for proactive glossary development and leveling sessions.

Misalignment of expectations and conflicting priorities were consistently identified as significant impediments. The dissonance between the iterative, experimental nature of ML development and the often more linear expectations of traditional software project management, as noted by Kästner [7] and Serban et al. [13], was a recurring theme. Similarly, the struggle to balance bug fixes with technical debt in data highlights the organizational friction caused by divergent priorities. These findings align with Nahar et al.'s [9] observations on managerial noise and the impact of siloed functions on clarity of responsibilities [7], reinforcing the need for continuous negotiation and alignment.

The observed lack of structured communication practices and inefficient use of collaborative tools could worsen these challenges. Reliance on infrequent, unstructured meetings and uncommunicated decisions directly impacts coordination, task tracking, and deliverable alignment. Although prior work advocates various communication strategies and tools [3, 8], our findings highlight the need for a disciplined, standardized approach when adopting platforms such as Jira or Notion, integrated into regular communication rituals to foster transparency and traceability.

Furthermore, our findings highlight the specific challenge of communicating the probabilistic nature and inherent uncertainties of ML models to software engineers, as also discussed by Ozkaya [10]. The difficulty in discussing model validity and data dependencies highlights a critical gap in mutual understanding that extends beyond mere vocabulary. This suggests a need for communication strategies that address the epistemological differences between deterministic software logic and probabilistic ML outcomes.

While Piorkowski et al. [12] advocate cross-training to bridge technical gaps, our findings illustrate practical barriers like resource constraints. Thus, our study expands on their work by focusing on immediate, operational-level communication practices as alternatives or complements to broader cross-training efforts. This suggests that while theoretically beneficial, its real-world application requires significant organizational commitment and strategic planning to overcome resistance and logistical hurdles.

These findings also build upon the first cycle, which established the empirical foundation for refining categories and further exploring communication practices. The ongoing research, with additional interviews planned until theoretical saturation (the advanced phase of STGT), aims to consolidate both explanatory and prescriptive theories for multidisciplinary teams working on AI-based systems.

6 Limitations

This study presents limitations that should be taken into account when interpreting the results. Firstly, the research is at an initial stage and is based on a small sample participants, selected by convenience, which may limit the generalizability to other organizational contexts or to teams with different levels of maturity.

Additionally, data were obtained through semi-structured interviews, which, although appropriate to the qualitative approach and to STGT, may be subject to recall or perception bias on the part of the interviewees. Triangulation with other methods could strengthen the results in future stages of the research.

Finally, the current analysis presents partial results. Continuing data collection until theoretical saturation is reached is necessary for the consolidation of categories and for the development of a more robust theory regarding communication in ML contexts.

7 Related Work

Effective communication among members of multidisciplinary teams is crucial for the success of projects that integrate ML models into software systems. The literature distinguishes two main types of communication: formal (meetings and documentation) and informal (quick information exchange) [8]. In our research, we observed that both are essential for the project's progress, but failures in structuring either can hinder the integration of AI-based systems.

When teams have distinct backgrounds, such as data science and software engineering, communication barriers arise, mainly due to the lack of a shared technical vocabulary, which can lead to misunderstandings and integration failures [2, 8, 12]. Our findings support this view, showing that the lack of clarity in communication regarding the objectives of the ML model was a critical point.

Nahar et al. [9] noted that technical heterogeneity generates linguistic noise, affecting team collaboration. Unlike their study, our research explicitly analyzes this phenomenon through the lens of communication practices in technical integration processes, proposing concrete recommendations to address it.

Kästner [7] points out that siloed structures and unclear responsibilities lead to conflicts, and also highlights the lack of consolidated best practices for cross-team integration. Our findings reinforce these issues by showing that silos hinder alignment between teams and that, despite initiatives like structured documentation, there is still no consensus on integrated workflows.

Various strategies to improve communication have been adopted, such as collaborative tools and glossaries [1, 3, 4]. Our findings confirm the importance of these tools, such as Jira[3] and Notion[4], but also highlight that the implementation of regular meetings and cross-training is insufficient to bridge the communication gaps.

Ozkaya [10] discusses how ML models differ from traditional software engineering, primarily in aspects such as explainability and handling uncertainty. In our research, we found that communication regarding uncertainties and data dependencies remains an ongoing challenge, especially since the probabilistic nature of ML models is not easily understood by software engineers, leading to difficulties in discussions about the validity of the models.

Studies also highlight difficulties in communicating complex results to stakeholders [3]. In our case, we observed that the limited interaction between modeling and implementation teams hindered the rapid adaptation, impeding the progress of project phases.

Cross-training between software engineers and data scientists is commonly recommended to reduce technical gaps between disciplines [8, 12, 14]. Piorkowski et al. [11] emphasize cross-training as a primary strategy to improve interdisciplinary collaboration by bridging technical knowledge gaps between software engineers and data scientists. In contrast, our work explicitly investigates everyday communication practices employed by teams during the technical integration of AI-based systems, offering practical recommendations that complement Piorkowski's broader organizational and educational approach.

[3]Jira: https://www.atlassian.com
[4]Notion: https://www.notion.com

8 Conclusion

This study investigated how communication practices across interdisciplinary teams impact the integration of AI-based systems. Using a qualitative approach grounded in STGT, the study addressed two research questions related to communicational barriers, providing empirical insights into the challenges and practices that shape interdisciplinary collaboration.

The findings indicate that communication gaps—such as lack of shared vocabulary, misaligned expectations and priorities, unstructured practices, and inadequate tool usage—undermine the quality of ML model integration. In contrast, practices like collaborative glossaries, regular meetings, and standardized documentation emerged as effective strategies to address these issues.

These findings reinforce the importance of addressing not only technical aspects but also social and organizational dimensions in the development of ML-enabled systems. The main contribution of this research lies in empirically demonstrating the centrality of interdisciplinary communication as a determining factor for the technical success of such systems.

As next steps, we plan to expand data collection until reaching theoretical saturation, progressively advancing the application of the STGT. The continuation of data collection will be guided by theoretical sampling, prioritizing participants who can contribute to the deepening or expansion of emerging categories. The study will seek professionals with technical profiles, experiences, or organizational contexts that offer relevant variation to the dimensions already identified, in accordance with STGT guidelines.

As the research progresses, we aim to move beyond the descriptive refinement of categories toward a more abstract level of theorization. This includes identifying core categories and exploring their relationships to support the development of an explanatory theory of communication in AI-based software projects. In addition, we intend to translate these insights into actionable frameworks or practical guidelines for engineering teams, contributing both to academic understanding and to the improvement of real-world software development practices involving machine learning systems.

ARTIFACT AVAILABILITY

All artifacts related to this study including anonymized interview transcripts, data collection protocols, coding structures, and analysis diagrams are publicly available via Zenodo: https://doi.org/10.5281/zenodo.15860313.

ACKNOWLEDGMENTS

We acknowledge the support of the Coordination for the Improvement of Higher Education Personnel (CAPES) for funding this research. We also extend our sincere thanks to all the professionals who voluntarily participated in the interviews. Their insights and contributions were essential to the development of this study. Special thanks to the members of the research group from SPLAB, the Software Practices Laboratory at the Federal University of Campina Grande, for their valuable contributions throughout the process.

REFERENCES

[1] C. Bogart, C. Kästner, J. Herbsleb, and F. Thung. 2021. When and how to make breaking changes: Policies and practices in 18 open source software ecosystems. *ACM Transactions on Software Engineering and Methodology* 30, 4 (2021), 1–56.

[2] Simon Brandstädter and Kh. Sonntag. 2016. Interdisciplinary Collaboration – How to Foster the Dialogue Across Disciplinary Borders? In *Advances in Ergonomic Design of Systems, Products and Processes*, Barbara Deml, Patricia Stock, Ralf Bruder, and Christopher M. Schlick (Eds.). Springer, 395–409. doi:10.1007/978-3-662-48661-0_35

[3] L. Dabbish, C. Stuart, J. Tsay, and J. Herbsleb. 2012. Social Coding in GitHub: Transparency and Collaboration in an Open Software Repository. In *Proc. Conf. Computer Supported Cooperative Work (CSCW)*. 1277–1286.

[4] C.R.B. de Souza and D.F. Redmiles. 2008. An Empirical Study of Software Developers' Management of Dependencies and Changes. In *Proc. Int'l Conf. Software Engineering (ICSE)*. 241–250.

[5] Rashina Hoda. 2022. Socio-Technical Grounded Theory for Software Engineering. *IEEE Transactions on Software Engineering* 48, 10 (2022), 3808–3832. doi:10.1109/TSE.2021.3106280

[6] F. Ishikawa and N. Yoshioka. 2019. How Do Engineers Perceive Difficulties in Engineering of Machine-Learning Systems? - Questionnaire Survey. In *2019 IEEE/ACM Joint 7th International Workshop on Conducting Empirical Studies in Industry (CESI) e 6th International Workshop on Software Engineering Pesquisa de Engenharia e Prática Industrial (SERIP)*. IEEE, 2–9. doi:10.1109/CESSER-IP.2019.00009

[7] Christian Kästner. 2025. *Machine Learning in Production: From Models to Products*. MIT Press, Cambridge, MA. https://mitpress.mit.edu/9780262049117/machine-learning-in-production All author royalties donated to Evidence Action. Released under Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International Public License. Corresponds to CMU course 17-645 / 11-695..

[8] I. Mistrík, J. Grundy, A. van der Hoek, and J. Whitehead. 2010. *Collaborative Software Engineering*. Springer.

[9] Nadia Nahar, Shurui Zhou, Grace Lewis, and Christian Kästner. 2022. Collaboration challenges in building ml-enabled systems: Communication, documentation, engineering, and process. In *Proceedings of the 44th international conference on software engineering*. 413–425.

[10] Ipek Ozkaya. 2020. What Is Really Different in Engineering AI-Enabled Systems? *IEEE Software* 37, 4 (July/August 2020), 3–6. doi:10.1109/MS.2020.2993662

[11] Sheila Maria Mendes Paiva. 2025. Artifacts of the Study on Communication Practices in AI-Based Software Engineering Projects (v1.0). Dataset. doi:10.5281/zenodo.15860313 39th Brazilian Symposium on Software Engineering (SBES 2025).

[12] D. Piorkowski et al. 2021. How AI Developers Overcome Communication Challenges in a Multidisciplinary Team: A Case Study. In *Proc. ACM on Human-Computer Interaction*, Vol. 5. 1–25.

[13] Alex Serban, Koen van der Blom, Holger Hoos, and Joost Visser. 2020. Adoption and Effects of Software Engineering Best Practices in Machine Learning. In *Proceedings of the 14th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)* (Bari, Italy) *(ESEM '20)*. Association for Computing Machinery, New York, NY, USA, Article 3, 12 pages. doi:10.1145/3382494.3410681

[14] J. Tranquillo. 2017. The T-Shaped Engineer. *Journal of Engineering Education Transformations* 30, 4 (2017), 12–24.