How AI Developers Overcome Communication Challenges in a Multidisciplinary Team: A Case Study

DAVID PIORKOWSKI, IBM Research, United States
SOYA PARK, Massachusetts Institute of Technology, United States
APRIL YI WANG, University of Michigan, United States
DAKUO WANG, IBM Research, United States
MICHAEL MULLER, IBM Research, United States
FELIX PORTNOY, IBM, United States

The development of AI applications is a multidisciplinary effort, involving multiple roles collaborating with the AI developers, an umbrella term we use to include data scientists and other AI-adjacent roles on the same team. During these collaborations, there is a knowledge mismatch between AI developers, who are skilled in data science, and external stakeholders who are typically not. This difference leads to communication gaps, and the onus falls on AI developers to explain data science concepts to their collaborators. In this paper, we report on a study including analyses of both interviews with AI developers and artifacts they produced for communication. Using the analytic lens of shared mental models, we report on the types of communication gaps that AI developers face, how AI developers communicate across disciplinary and organizational boundaries, and how they simultaneously manage issues regarding trust and expectations.

CCS Concepts: • **Human-centered computing → Empirical studies in collaborative and social computing**.

Additional Key Words and Phrases: multidisciplinary collaboration; artificial intelligence; machine learning; data science; shared mental models

ACM Reference Format:
David Piorkowski, Soya Park, April Yi Wang, Dakuo Wang, Michael Muller, and Felix Portnoy. 2021. How AI Developers Overcome Communication Challenges in a Multidisciplinary Team: A Case Study. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW1, Article 131 (April 2021), 25 pages. https://doi.org/10.1145/3449205

1 INTRODUCTION

As machine learning (ML) continues to transform the world around us, data science teams are becoming a standard fixture at companies and organizations. These data scientists provide the expertise necessary to incorporate ML and artificial intelligence (AI) solutions[1] into a wide variety of business problems (e.g., healthcare [15], HR [30], and education [69]). New technologies and processes for solving problems bring not only new opportunities, but new challenges as well.

[1] In this paper, we use the term Artificial Intelligence (AI) to include both machine learning and artificial intelligence.

Authors' addresses: David Piorkowski, IBM Research, United States, djp@ibm.com; Soya Park, Massachusetts Institute of Technology, United States, soya@mit.edu; April Yi Wang, University of Michigan, United States, aprilww@umich.edu; Dakuo Wang, IBM Research, United States, dakuo.wang@ibm.com; Michael Muller, IBM Research, United States, michael_muller@us.ibm.com; Felix Portnoy, IBM, United States, fportno@us.ibm.com.

2573-0142/2021/4-ART131 $15.00
https://doi.org/10.1145/3449205

Correspondingly, a healthy amount of research has emerged looking at a broad range of topics including the challenges of building AI software [2, 5, 57, 65] and how data scientists collaborate [58, 59, 72].

Prior work on data scientists' collaborative practices have provided high-level accounts on how data scientists work, and suggested that data science teams face intra-role communication gaps in code reading, code reuse, and code documentation activities [22, 48, 72]. In this study, we focus on the inter-role communication gap. We ran a case study of a data science team at IBM whose role is to provide AI solutions for other product teams in IBM. For the purposes of this paper, we term members of this team *AI developers*, since they include members who are AI-adjacent in that they are not data scientists but still require AI knowledge to do their job. To help frame our understanding of both the challenges and the solutions, we used *shared mental models* (SMMs) theory [13]. At the highest level, SMMs provide an abstraction for team members' common understanding of task responsibilities and what the corresponding information needs are. This allows them to more readily anticipate each others' needs and work together more efficiently [36]. Effective communication is a core tenet of SMMs and the aspect we were interested in using to understand our findings. In particular, we focus in on the principles proposed by Scheutz et al. for realizing SMMs, which are described in detail later in the paper [49]. For simplicity in this first paper, we focused only on the AI developer's mental models *within* a data science team. Subsequent research should examine the "other side" or "sides" of the communications gap(s), namely the perspectives of other stakeholders in the broader AI product context.

Accordingly, we conducted a semi-structured interview study with four AI expert participants over multiple sessions (in total 10 sessions) to construct a detailed understanding of the communications gaps they faced in the AI lifecycle [61] and how they overcame them. Interviewees who participated in two or more sessions additionally shared and discussed communication artifacts with us. As a result, we had ten interview sessions in total. Given the tight focus and small participant count, we consider this as a formative, yet in-depth, look at the following questions.

(1) What are the key communication gaps AI developers faced?
(2) How do AI developers overcome those gaps and communicating across roles?

In this paper, we detail how AI developers overcome communication gaps using the tools and techniques that are currently available. We explain why these approaches are successful by viewing them through the lens of SMMs and bring out the most problematic communication gaps that AI developers face. In doing so, we provide a glimpse into the difficulties faced by people in this role. This understanding can then serve future work that aims to address the communication challenges that they face and the complementary challenges that their colleagues face across the communication gap, in roles that are outside of the data science team.

We found that data scientists are concerned about communication gaps stemming from mismatched expertise between the team and their stakeholders. To address the gap, the teams hold informal education sessions and have continuous sync-up sessions to establish trust. Finally, we discuss best practices in AI development according to SMMs.

2 RELATED WORK

Our research contributes to the growing discussion of collaborative data science in Human-Computer Interaction (HCI) and Computer-Supported Cooperative Work (CSCW) (e.g., [3, 24, 37, 46]). We summarize related work into four parts: (1) software engineering for AI systems, (2) emerging roles and collaboration practices of an AI team, (3) communication challenges in AI teams and software engineering teams, and (4) SMM theory.

2.1 Software Engineering for AI Systems

The advances of AI techniques open the opportunities for companies and developers to build software products that handle tasks intelligently, improve accuracy and efficiency, and provide personalized experience. AI has been widely applied to many domains such as helping diagnose patients [15, 62], building chatbots to improve customer service [30, 68], and enhancing the education experience with intelligent tutoring systems [69].

Building AI products involves not only the process of extracting insights from data (data-centric), but also constructing models from insights and building software products (model-centric or product-centric). The process of extracting insights from data has been well-studied and categorized into three high-level stages [63]: data preparation, model building, and model deployment. While some studies in HCI and CSCW investigated the workflow from a data-centric perspective [25, 38, 42], others discussed the process of integrating models and building AI products in detail [2, 5, 57, 65], Amershi et al. [2] summarized nine stages of machine learning workflow as data-oriented (e.g., collection, cleaning, and labeling) and model-oriented (e.g., model requirements, feature engineering, training, evaluation, deployment, and monitoring). This workflow is iterative and contains many feedback loops. In practice, AI developers often struggled to establish a repeatable process, as Hill et al. found in their interview study with AI developers from various expertise levels [17]. In this paper, we use the AI development lifecyle as proposed by Amershi et al.'s work [2].

Prior research investigated the technical challenges of software engineering for AI systems, such as difficulties in problem formulation and specifying desired outcome, lack of critical analysis of training data, and lack of evaluation of models with business-centric measures [31, 64]. In addition, building AI products requires the development team to consider many facets, such as interactions, performance, cross-platform experience, and social good [17, 53]. Thus, the AI development team must embrace collaboration between different roles and leverage expertise from each other—despite the relatively low usage of documentation for this purpose [44, 48, 72]. Our work aims to reveal the hows and whys of the AI development teams communicating across roles and stages.

2.2 Emerging Roles and Collaboration Practices of AI Development Teams

Prior research has shed light on emerging roles and collaboration practices in a data-centric workflow. We use the phrase "data scientist" referring to people who have the technical skills to find trends and manage data in a variety of domains, such as decision sciences and business intelligence, product and marketing analytics, fraud and risk analytics, data services and operations, and data engineering and infrastructure [43]. Building AI systems falls into the criteria of data engineering and infrastructure, where development and implementation skills are critical for data scientists. With the growing impact of data science, many jobs across all industries continue to be changed by data scientists [56]. For example, many software developers are now learning machine learning through informal resources (e.g., interactive machine learning tools [70]), hoping to adopt machine learning into their own practices [8].

Effective collaboration can help data science teams to leverage expertise from each other and to improve quality of work [58]. Studies have investigated collaboration models among different team settings, such as professional data scientists [58, 59], civic data scientists [19], domain experts [32, 41], and software-oriented data analysts [23]. Zhang et al. used a survey to study the general collaboration practices of data science teams among both technical team members and non-technical team members [72]. Building on top of Zhang et al.'s work, our work takes a deeper focus and specifically examines kinds of communication gaps and collaboration practices in AI development teams. Through in-depth interviews, we discovered that mismatched expertise between data scientists and business experts hampers their collaboration, and we learned how data scientists address the issue.

2.3 Communication Challenges in AI and Software Engineering Teams

Prior work has also investigated the communication challenges that happen in AI and software engineering teams. For example, Storey et al. found that engineer users on GitHub have a wide variety of "socially enabled communication channels" and social media to exchange information along with their coding project, but they still faced communication challenges such as miscommunication, language barriers, fear of intimidation and poor attitudes [52].

In the data science domain, the communication challenge is known to be broad and fraught with difficulties. Mao et al. [32] reported that in scientific discovery projects, data scientists and bio-medical scientists often have different motivations. Such mismatch of their the motivations complicated their communication behavior in that bio-medical scientists users kept asking data scientists new questions, even though the data scientists had not yet found an answer for their previous question. Hou et al. reported in their interview study that subject matter experts and data scientists "speak a different language" [19]. Sometimes the data scientists do not know how to translate a subject matter expert's business problem into a well-defined data science problem. Thus, it may need a third party to be the bridge or the "broker role" [67] to do the translation for the communication.

To remediate these communication challenges in an AI or software development team, prior works in HCI have proposed novel collaboration systems to aid the teamwork [7, 51, 54, 59]. For example, Wang et al. built a system that allows AI developers to chat alongside their code in a Jupyter Notebook environment, and the system can conceptually link those informal communication messages back to the code [59]. Storey et al. investigated shared waypoints and social tagging in collaborative software development [51]. But most of systems were built for supporting communications between the technical AI developers, not considering the inter-role communications between an AI developer and a domain expert.

Thus, we conducted this interview study to investigate AI development teams' communication practices, with a focus on the communication challenges that happened between the AI developers and the domain expert collaborator. By reflecting on the tools and techniques that AI developers use to communicate with other stakeholders, we hope to open opportunities for future tool designers to build better collaboration tools.

2.4 Shared Mental Models

Effective team collaboration, particularly multidisciplinary collaboration, requires team members to hold SMMs on task requirements, procedures, and role responsibilities [13]. Studies revealed that in multidisciplinary collaboration, professionals are unwilling to accept overlapping duties over roles [20] and rely on collaborative work tools to establish common ground [1].

Inspired by the general literature on common ground [11], our work uses SMM theories as a lens to discuss the communication and collaboration practices between AI developers and other stakeholders. In this approach, an SMM provides one means to establish common ground [55]. The SMM literature distinguished between two models of shared understandings across team members [29, 47, 49]. The *task model* consists of equipment (e.g., equipment functioning, operating procedures) and task (e.g., task procedures, task strategies), while the *team model* consists of team interaction (e.g., roles, communication channels) and team (e.g., knowledge, skills). Yu et al. [71] summarized four stages for developing an SMM within agile software development teams from prior literature: knowing meta-knowledge of the team [27, 33], learning and building the team's transactive memory system [6], understanding and reaching consensus [55], and executing team

Table 1. A summary of the principles of how humans realize shared mental models [49]. This framework is helpful for describing collaborations activities in the domain of AI.

Principle	**Definition**
Consistency	Ability to maintain stability by resolving conflicts due to differing perceptions, differing knowledge states, asynchronous information, and missing updates.
Reactivity	Ability to effectively address unanticipated events or state changes by informing team members of the changes and adapting goals and plans to account for the new situations.
Proactivity	Ability to anticipate problems, bottlenecks and failures and take proactive actions, such as asking for clarification or offering assistance.
Coordination	Ability to work together via overall cooperative attitudes, such as establishing joint goals and plans, transparent task assignments, and truthful information sharing
Knowledge Stability	Ability to understand the staleness of information over time and adjust sampling rates according to their confidence in the information's validity.

goals. They identified several activities in agile software development practices that can improve collaboration using SMM theory (e.g., planning, reflexivity, leader briefing – which are integrated in our coding schema in Table 4). Scheutz et al. extended SMMs into the domain of AI, using SMMs as the theoretical foundation behind frameworks describing collaborations activities of humans and agents [49]. Grounded in prior work on SMMs, they proposed five principles of how humans realize SMMs, shown in Table 1. We leverage these principles as the lens through which we explain and understand of how AI developers work. Our work aims to uncover the unique practices and challenges that AI developers have when building SMMs with other stakeholders.

3 METHODOLOGY

To identify the communication gaps that AI developers faced, we interviewed AI developers and qualitatively analyzed interview transcripts them using a SMM lens. What follow are the details of both the participants and the study design.

3.1 Participants

Our study involved real projects in IBM. We wanted to understand difficulties and pain points of production-targeted data science work-practices. Participants described issues and experiences that could only be communicated with agreed protections of privacy and confidentiality. Therefore, our sample was restricted to within the company.

Using our access as employees of the same company, we recruited participants through an established working relationship with the data science team represented in this study. We will refer to this team as the *AI team*. We sent out emails to IBM's mailing lists detailing the selection criteria. The criteria were:

- the participant had to be working on a project involving Machine Learning (ML) or AI
- the project was nearing completion or complete
- ML or AI was an important part of their role on the project

Our recruitment resulted in four participants from the same department in the company, but from different teams. Table 2 shows participants' roles, experience and expertise. The brackets after each role refer to the labels we use to identify each participant's quotes.

We established a working relationship with these teams specifically because of a couple of key aspects of how they work. First, they work on a wide variety of problems with a variety of teams, providing them with a perspective that is wider than a data science team working on a single domain. Furthermore, they do not maintain any of the AI applications they build. Instead, they hand off all the projects they create for other teams to maintain. Therefore, the AI team should be incentivized to provide enough information to others for maintainability's sake. Thus, we hoped that this aspect would bring to light useful collaboration practices.

3.2 Interview Protocol

To answer our research questions we ran a total of 10 hours of semi-structured interview sessions over four participants. We split interviews into two different phases lasting from 45 minutes to one hour:

- The *project-overview phase* consisted of one interview to describe the AI project and roles involved. It also served to scope future interviews. We conducted four project-overview sessions over four participants.
- The *artifact-in-depth phase* consisted of interviews to discuss one or more artifacts brought by the participant to demonstrate how they communicate across roles. We conducted six artifact-in-depth sessions over four participants, with interviewees sharing one to three artifacts per interview.

In the project-overview interviews, participants recalled an ML or AI project they had recently finished or were about to finish. In these interviews, we asked participants about details of the project, the composition of the team, and how ML or AI fit into the project. At the conclusion of the project overview interview, the researcher and participant jointly built a visualization in Mural[2] which aimed to map the various roles involved on the project to the phases of the AI development lifecycle. Figure 1 shows a consolidated view of these joint visualization activities. In this task,

[2] https://mural.co

Table 2. Interview participants, labeled based on their role in their team. Expertise and the number of project collaborators are provided by the participants. * We reviewed artifacts with Data3, but due to confidentiality, we are unable to share information from these interviews in this work. Data3's project overview interview data is included.

Role [*Label*]	**Years of experience**	**Expertise**	**Project collaborators**	**# Artifacts shared**
Data scientist [*Data1*]	4	Machine learning, unsupervised learning, natural language processing, and statistics	8 internal, 10 external	2
Data scientist [*Data2*]	4	Building data science models in a business setting to help business transformation	10 internal, about 10 external	1
Data scientist [*Data3*]	2	Machine learning, deep learning, statistical analysis	7 internal, 2-5 external	0*
Strategy consultant [*Strat1*]	2	Business strategy and project implementation	3 internal, 8 external	5

participants were shown a proposed starting AI development lifecyle and were invited to modify it to make it more closely match the project they described. Once they finished modifying the lifecycle, participants were asked to place the roles that were mentioned during the interview into any of the development phases that were represented in the Mural. Roles included members from the participant's team, and also other teams who were involved in the project. In our case, all four participants were working with at least one "client" team, whom we term the *Stakeholder team*, that would eventually take on ownership and maintenance of the final product.

In the artifact-in-depth interviews, participants shared artifacts that had previously or currently served as a mechanism for facilitating communication across roles. We asked participants to share artifacts used in either of the "Model evaluation" or "Model deployment and integration" AI lifecycle stages if possible. We chose these two stages because, according to the project-overview interviews, these two stages tended to involve lots of different roles across multiple teams, and thus, were likely to reveal communication challenges. Additionally, AI developers indicated in the project-overview interviews that these stages required them to produce explanatory artifacts. Participants shared from one to five artifacts with researchers, which were discussed in detail in the artifact in-depth interview sessions. Artifacts consisted of digital objects such as slide presentations, documentation, software repositories, and README files. For each one of these artifacts, we ran an artifact-in-depth interview that uncovered how the artifact was created, how it was used, how successful it was, and its limitations. One participant's artifacts and corresponding interviews were removed from the study due to confidentiality concerns, but data from the project-overview interview remains.

Interviews were conducted remotely using WebEx[3], were recorded and transcribed for qualitative analysis.

3.3 AI Team Working Practices

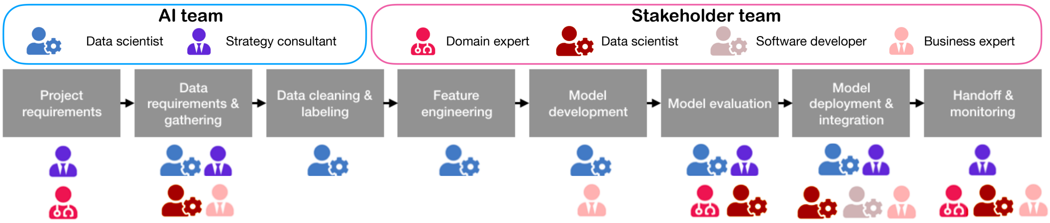


Fig. 1. Aggregated view of our interviewees' different roles involved in each stage of AI workflow. Each team's composition slightly varies, and can be found in the Appendix. An AI team consults and implements AI solutions. It normally consists of technical and business representatives. A Stakeholder team requests help from a AI implementation team. They are knowledgeable in their domain, however lack knowledge in ML or AI. We presented the stages linearly, but in reality the AI development process is highly iterative.

Participants reported that in general, two teams are involved in the AI project: an AI team and a Stakeholder team. The AI team is responsible for gathering the Stakeholder team's needs, formulating the problem into the data science domain, and building an AI model to solve the problem. The Stakeholder team often has deep expertise and domain knowledge in a particular field, and is responsible for helping the AI team to understand the real-world problem, providing guidance, and verifying and consuming the final model result.

The AI team has two main roles, the data scientist and the strategic consultant. The data scientists build and evaluate models. The strategic consultants shepherd the project and keep up the

[3] https://www.webex.com

Table 3. The coding book (9 categories and 27 concepts) generated from iterative axial coding.

Category	**Concept**	**Definition (Utterances about...)**
Data Scientists Provided Information	Objective	... project objectives
	Algorithm selection	... deciding between algorithms and final algorithm selection
	Model performance and results	... evaluating model quality, model performance or model results.
	Model improvements	... model improvements when compared to a previous version or an existing baseline
	Model validation	... validating model results
Stakeholders Requested Information	Stakeholder requested information	... information needs requested by the Stakeholder team
Data Scientists' Rationale for Providing Certain Type of Information	Missing info	... reasons to address missing information
	Data science education	... reasons to educate others about data science concepts
	Contextualization	... reasons to contextualize the results in the problem domain
	Legal and regulatory	... reasons regarding legal or regulatory compliance
Information Conveyance Selection	Examples	... using a data instance as an example
	Visualizations and plots	... building visualizations and plots
	Listing numbers and metrics	... providing numerical data such as summary statistics or metrics
	Explanatory story	... creating an analogous story to provide explanation
	Definitions	... defining terms, typically in a less-technical way
Data Science Project Workflow	Data Science Project Workflow	... steps, tasks, timeline, and other details about the workflow
Data Science Collaboration Roles	Stakeholder team roles	... roles and responsibilities of members in the Stakeholder team
	ML team roles	... roles and responsibilities of members in the ML team
Trust Between ML Teams and Stakeholder Teams	Trust development	... how trust was built between the two teams
	Distrust	... instances of distrust between the two teams
Communication	Communication challenges	... communication difficulties faced by the Stakeholder team
	Communication artifacts	... digital creations to aid communication between teams
	Communication frequency	... how often communication happened between roles
	Efforts in preparation for communication	... timelines or work done to prepare communication artifacts
Technology	Communication technology	... technologies used to prepare communication artifacts
	Data science technology	... technology used for data science

communication between the two teams. On the Stakeholder team are the domain experts, data scientists, software developers and business experts. Domain experts are the subject matter experts in the problem domain that the AI team is trying to help solve. Although data scientists exist on the Stakeholder teams, their role is the project is to maintain the solution that is built by the AI team. Stakeholder team data scientists tend to be more specialized to their specific business domain then the data scientists on the AI team. Software developers are responsible for helping package the AI solution into an application or service. Finally, the business expert is a management role that oversees an entire line of business that would be represented by multiple domain experts.

An aggregated view of how the roles are sorted and related to the AI development lifecycle is presented in Figure 1. This figure was generated from the Murals that participants created as part of the project-overview interview. This figure shows how the roles drop in and out of different phases of the lifecycle as the project matures. As a result, much of the work happens outside the view of others, hinting as to why maintaining a shared knowledge state across all the different roles is challenging.

3.4 Qualitative Coding

To help map the current practices for achieving SMMs to the communication gaps that AI developers faced, we needed to identify the relevant qualitative data in the transcripts. To do so, we conducted multiple interview sessions with our participants. As a result, we had 10 sessions. We followed a consensus-coding protocol, as recently reviewed by McDonald et al. [34]. We conducted axial coding of the 10 interview transcripts with an open-coding protocol. Two of the researchers started with one interview and each developed a code book independently to generate concepts and organize these concepts into categories. The concepts were chosen to be broadly about topics related to the types of information exchanged, how information was exchanged, and difficulties around information exchange. The two coders then sat together and discussed the disparity of the code book and their understanding and definition. Then, they split and independently coded transcripts again based on the agreed upon concept and category definitions. This process was iterated upon for 4 times, until they come to an agreement about the concepts, categories, and general themes. Once agreement was reached, they divided and conquered the other nine interview transcripts. The final code book has nine categories and 27 concepts and is summarized in Table 3.

4 RESULTS

Participants faced a varied assortment of challenges related to communication gaps. The results of our qualitative analysis are presented in two parts. First, in Section 4.1, we bring to focus the three communication gap themes that AI developers faced. We also elaborate on the role that education plays in the communication gaps we uncovered. Second, in Section 4.2, we dive into the AI developers' world and highlight specific examples of challenges that they faced. Those examples are organized using the principles for realizing SMMs presented in Table 1. These vignettes highlight the communication gap and the techniques and tools participants chose to address them. We also provide real-life examples from slide decks that demonstrate the techniques that AI developers used to get their points across.

4.1 RQ1: Kinds of Communication Gaps

Participants' reflections on communication challenges centered on three major themes: knowledge gaps across roles, establishing trust, and setting expectations.

4.1.1 Knowledge Gaps between Roles due to Mismatched Expertise. Team members came from different technical backgrounds and expertise. This configuration makes the collaboration challenging. ML and AI involve several domain specific concepts such as model algorithms, evaluation metrics and advanced statistical measures that make it difficult for software engineers to learn, let alone business stakeholders [8]. AI teams struggle to teach these concepts to stakeholders without getting into too much detail. The lack of AI knowledge caused stakeholders to misinterpret model performance: *"For defects, ... a lot of times it's because people don't understand how the model works,"* [Data2].

Teams tend to go through an *education phase* at the beginning of the project to address this issue. The ultimate goal of such education is to bridge the knowledge gap and to lay the groundwork for upcoming collaboration. Echoing findings from previous work, one of the common practices is to hold AI education sessions for software engineers to overcome the knowledge barrier [2]. AI teams' education efforts on the fundamentals of machine learning extended beyond software engineers, to other non-AI savvy roles such as business users. For example, one kind of education specifically focused on mapping machine learning terminology to business terminology. One data scientist said: *"Some of the languages came from them [business stakeholders]. Like, they had an idea of the near match, which is from their perspective,"* [Data1]. However, such education sessions are done in an informal manner and there is no effective way to teach the team, resulting in lots of trial and error. Strat1 said: *"If one way [of explaining] doesn't work, ... where there's a (need for) common understanding or common communication. We... really try to keep building examples and education materials along those lines."*

This education session is *bi-directional*. Domain experts also educate AI teams by explaining domain knowledge and the desired behavior of the model. Domain knowledge is hard to capture in one session. Also topics change as patterns and findings emerge from the data as the teams progress in the project. Thus, AI teams repeatedly ask questions and confirm to domain experts throughout the entire collaboration.

4.1.2 Establishing Trust Across Disciplines. Given the nature of a multi-disciplinary team without a shared work history, it is difficult to establish trust between team members [1, 20, 26]. Due to these fundamental differences, when business stakeholders and data scientists work together, stakeholders do not understand the hurdles and contribution of data scientists and vice versa. According to Strat1, *"We also even schedule additional time beyond that specifically on particular topics that we find difficult just because we know this is a new area and it's kinda key."* Without building enough trust upfront, conflicts are inevitable. The same participant said: *"There's a lot of trust and understanding upfront and when we kinda give these high-level structures and architectures of what we can deliver."*

Trust building took on many forms, not only from repeated contact with regular meetings but also in the varied efforts that the project team undertook to explain particular model concepts or metrics. Additional effort was spent in addressing differences between how AI is developed versus more traditional software. An important example of this was when the AI implementation team explained to the stakeholders that the model's performance does not necessarily improve as they make incremental improvements on a model: *"[When] we didn't have any particular performance increase, we had to give an explanation (and to let) them know that it's kind of a trial and error... That was a real kind of struggle to let them know, we weren't actually doing anything wrong and nothing was broken,"* [Strat1].

4.1.3 Setting and Managing Stakeholder Expectations. Setting expectations played a key role to clients who were unfamiliar with the uncertainties inherent to machine learning: *"They [stakeholders] had made it very clear previously, that they're not comfortable moving forward with what they deem as a '[closed] box'. So we need to be as transparent as possible in the modeling"* [Strat1].

For instance, at the early stage of collaboration, AI teams demonstrate a sample model to stakeholders and show its prediction on certain data points. The teams have to persuade stakeholders that AI will outperform current techniques being used in their service:

> *"This team is very used to rigid logic rules or logic trees in terms of making decisions, ... but a big miscommunication... difficulty [was] they would see one particular data point or factor and say well, this always should give a positive match. And we have to really explain that."* [Strat1]

Additional challenges of such persuasion stem from the fact that each stakeholder is looking for different performance metrics, hence there is no general way of communication but it makes the AI experts work on a case-by-case basis: *"A lot of times depending on their subjectivity. I think that's the hard part to really assess the benefit— how this model has improved,"* [Data2].

4.1.4 Communication Gaps from a Shared Mental Models Lens. SMM theory's principles provide a framework for interpreting how AI developers addressed the communication gaps identified above. By framing the issues from the lens of theory, we begin to uncover potential reasons *why* certain strategies are successful and thus, can better support AI developers in the communication challenges that they face. We summarize the findings above from this perspective as follows.

From a *consistency* point of view, the desired outcome from setting expectations and education is to get both the AI and Stakeholder teams on the same page working towards the same purpose. From a *proactivity* perspective, the AI team's past experiences and expectations lead the team to address potential misunderstandings early and also, help build and maintain trust. Finally, from a *reactivity* perspective, part of managing expectations is reconciling misunderstandings when they occur. In such situations, the AI team has to be quick on their feet as further explained in the next section.

4.2 RQ2: How AI Developers Cross Communication Gaps

Prior research has shown the kinds of problems faced in developing AI software [2, 17] and the tools used in this space [72], but to our knowledge, there is little known beyond survey results [72] about the contextualized specifics of how AI developers communicate with other roles or the motivations behind that communication. In this section, we shed light on these questions by providing specific examples of participants' responses to information needs. We loosely frame the discussion using the principles of how people realize SMMs from Table 1 as they neatly capture participants' motivations from the interviews.

The coordination principle is omitted due to a lack of data from participants about it. The coordination principle emphasizes joint goals and plans and information sharing (Table 1). However, part of the answer to RQ1, above, was that the two teams' mismatched expertise (section 4.1.1), resulting in different perspectives and languages about goals and plans, especially in terms of expectations (section 4.1.3). Those differences may have reduced the effectiveness of their information-sharing. Thus, the absence of discussions related to the coordination principle is consistent with what we learned about communication gaps in relation to RQ1.

4.2.1 Consistency Gaps: How Knowledge is Communicated. At its core, the consistency principle is about maintaining a shared knowledge state. The reasons for why SMMs fall into conflict vary, but in this case study, we observed effects from the asynchronous nature of the work and the imbalance of data science skill across roles. Recall that the AI team and Stakeholder teams mingle, separate and mingle again in the different phases of the lifecycle and that there are entire phases of development where data scientists' skills are put to use building models in (mostly) isolation. Consequently, there are abundant opportunities for mental models to differ. Therefore, in this section, we describe the different ways that AI developers shared information with others, via specific examples of the problems they faced and how they overcame them.

One recurring way that participants overcame the asynchronous nature of their work and maintained consistency was by answering questions from other roles. Many of the questions fielded by the participants are best described as quick explanations such as explaining what certain lines of code did, *"Most of the questions were in reference to the actual code like, 'in this (Python) file'. Like, 'I don't know what line two hundred is doing,"'* [Data1] or specific requests for code documentation, *"I mean after several iterations of just communication back and forth I understood that they needed* *to know kind of more code documentation for the error codes in particular,"* [Strat1]. Participants described these types of requests as straightforward to answer as they could be addressed after just a quick glance at the code in question. Questions such as these were answered during meetings using screen-sharing technologies, via message services such as Slack, or email.

As questions started to repeat, participants pivoted to shared information spaces such as cloud content management tools (e.g., Box, Dropbox) or GitHub repositories' README files, Data1 recalled such an instance:

> *"It started out more with just them asking questions and, you know, I would have meetings to respond to the questions, things like that. But after maybe two or three of those meetings, I started to realize that we're kind of getting some questions repeated... so I thought it would just be easier to get all the one document, then we can share that."* [Data1]

The consolidation of information into a shared resources can be viewed as a way to reduce the cost of interruption by making the questions publicly available to others (e.g., [39]).

AI developers' data science expertise often left them as the only ones in the room who could explain what a model was doing and how to interpret it. Knowing this, AI developers took the task of educating others seriously, often spending lots of time preparing materials. Strat1 recalled one example where he spent time over "2 to 3 days" preparing three PowerPoint slides to explain the concepts of precision and recall to the Stakeholder team. Although the team understood the concepts, they were unable to sufficiently map those concepts to the business problem being solved. The interviewee lamented, *"They (Stakeholder team) understood recall and precision better, but that still didn't really resonate with them in terms of business impact and more,"* [Strat1].

The situation above is indicative of the overarching educational challenge that all the AI developers talked about: how to map a problem from the business domain to the AI domain. This problem is particularly thorny since each group seems to express the problem in their own language (again impacting the substrate on which the coordination principle depends), and this complicates communication regarding how a business problem is translated to a model's algorithm and outputs.

> *"I guess for this audience, they had some concept of how— what this means conceptually, right? Some of the language came from them. They had an idea of the near match, which is from their perspective... This was a way to kind of explicitly say how we're identifying it (the concept of a near match) in the code... I don't think they understood how we were combining all these different elements into this measure of closeness."* [Data1]

Data1's presentation attempted to address this gap by mapping the content of his presentation to stakeholder roles' perceptions. He did so by leveraging a specific use case that stakeholders were very interested in solving.

> *"I think these examples were chosen intentionally because for (the Stakeholder team) this (use case) is a really big one for them... So it is something that was on the forefront for them because we thought this was a relevant example for this audience because that was something they were focused on."* [Data1]

We observed that AI developers have a tremendous influence on decisions made by the Stakeholder team due to this knowledge imbalance. Successes and failures could be traced back to how effectively an AI developer was able to educate others on how a model worked.

> *"I guess the major outcome, which I think we achieved, just try to get management up to speed on how we had improved this algorithm and hopefully also get them excited about it... Our ultimate (goal) is to actually implement this in their business so they can start to leverage it and hopefully make some improvements."* [Data1]

4.2.2 Addressing Gaps: The Role of Reactivity and Proactivity. Uncertainty is a given when building an intelligent system, so it becomes an important aspect of maintaining an SMM. Dealing with that uncertainty unfolds in two ways captured in the reactivity and proactivity principles. Uncertainty leads to changes which require teams to respond. Reactivity is about responding to *unanticipated* problems whereas proactivity is about responding to *anticipated* problems.

Unexpected problems that participants handled reactively included explaining model performance and understanding end users. One of the well established difficulties of building AI systems is that it is difficult to predict a model's performance without first building the model and evaluating it. Techniques for estimating how likely a model is going to meet stakeholder expectations often boils down to AI developers' experience. This lack of predictability is often at odds with stakeholders' preexisting experiences with working with traditional software teams. In Section 4.1.2, we described Strat1's difficulties in explaining why models were "underperforming" compared to their best shared predictions and they had to explain the discrepancy to their business counterparts through explaining how machine learning evaluations work.

In such instances, participants exhibited patterns of reactive communication, where they would decipher possible causes for under-performance or prepare some alternative approaches to try in the next iteration. This allowed the team to focus on other issues in the project, as a plan was already in place for the underperforming model.

In another example, Data2 talked about how users were unwilling to use the system that was created because they were unwilling to trust the prediction it made. They summarized the problem:

> *"The biggest challenge for us, I believe, is we need to communicate with business stakeholders and with all the users, to persuade them, educate them, or prove to them that the [prediction] does make sense."* [Data2]

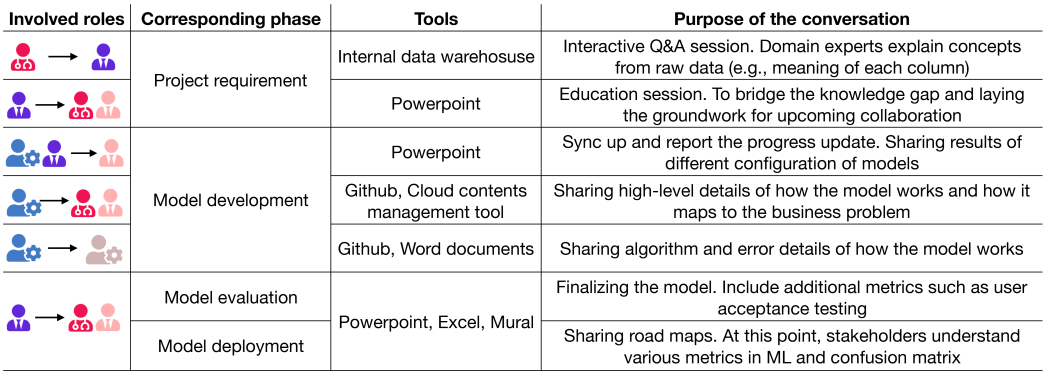
We now turn from reactivity to strategies for proactivity. Some communication problems can be anticipated in advance, and documentation is a prime example.

Data scientists are experts at statistics and AI modeling but are typically novices to the business domain, whereas stakeholders are the opposite. To bridge the gap, one of the practices we observed was when data scientists proactively documented model information to inform their collaborators. Data1 succinctly summarized the intended audiences for these documents and the goal, *"So I guess (the documentation) is high level business and high level technical. So hopefully regardless of who the person is they can get some kind of a sense of what's happening here."*

Incorporating domain knowledge into a model is a key to successful modeling. To do so, data scientists ought to understand domain knowledge embedded in their data. Our participants maintained documentation per project to capture domain knowledge. Unlike AI-concept one-directional documentations for stakeholders, these domain documentations take a role bi-directional channel to be revisited and reviewed with stakeholders. Through this documentation, stakeholders can provide feedback and data scientists can confirm their observations.

Education combined aspects of reactive and proactive strategies. Although participants were aware of the communication gaps about education, and were thoughtful about how to respond to them, there were still instances where those efforts failed. And in the iterations that followed, what started as a proactive activity effectively became a reactive one as participants tailored their content based on the feedback that they got. It was in learning what stakeholders knew that led to well-designed examples that could be reused effectively.

4.2.3 Effective Knowledge Sharing. There are lots of different types of information requests that occur, challenges that they carry and approaches to addressing those needs as effectively as possible. Making sure that the information is shared effectively as it is updated is particularly important as much of the work is done independently of others. As we have covered the specific details



Involved roles	Corresponding phase	Tools	Purpose of the conversation
👤 → 👤	Project requirement	Internal data warehosuse	Interactive Q&A session. Domain experts explain concepts from raw data (e.g., meaning of each column)
👤 → 👤👤		Powerpoint	Education session. To bridge the knowledge gap and laying the groundwork for upcoming collaboration
👤👤 → 👤	Model development	Powerpoint	Sync up and report the progress update. Sharing results of different configuration of models
👤 → 👤👤		Github, Cloud contents management tool	Sharing high-level details of how the model works and how it maps to the business problem
👤 → 👤		Github, Word documents	Sharing algorithm and error details of how the model works
👤 → 👤👤	Model evaluation	Powerpoint, Excel, Mural	Finalizing the model. Include additional metrics such as user acceptance testing
	Model deployment		Sharing road maps. At this point, stakeholders understand various metrics in ML and confusion matrix

Fig. 2. Different tools used for inter-team communication and the reason of conversation.

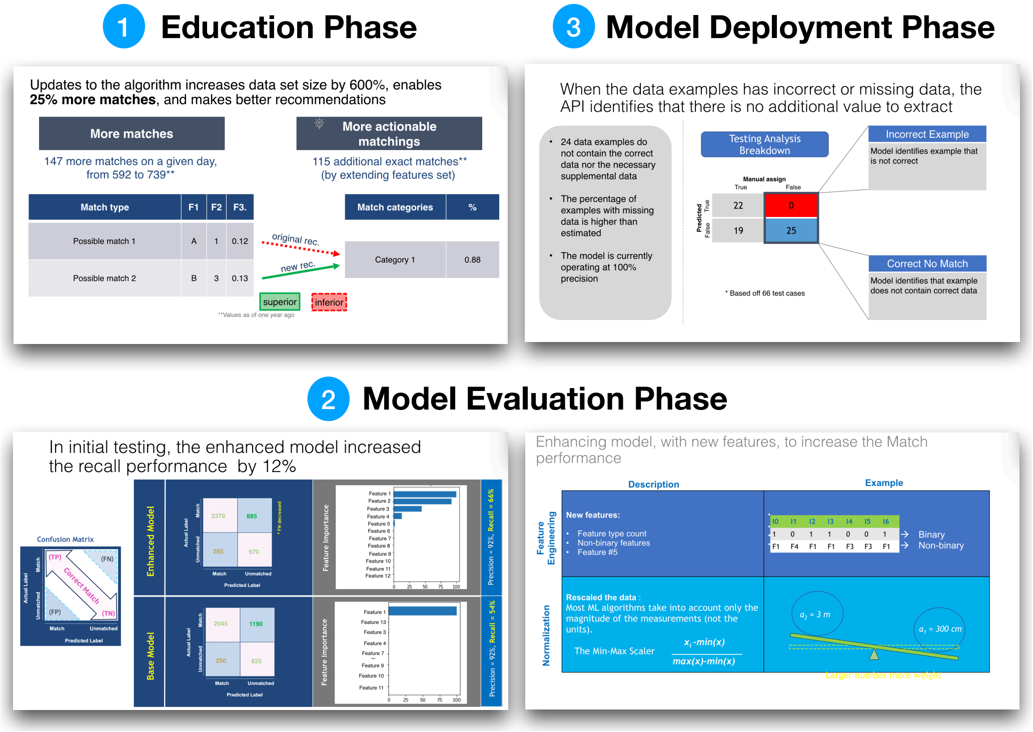
in the sections above, here instead we provide a summary of our findings. Figure 2 summarizes the inter-role communications that our participants discussed in the interviews. In the project requirement phase, domain experts, business experts, and strategy consultants gathered together to bridge the knowledge gap and lay the groundwork for upcoming collaboration. Data scientists are usually involved in the model development phase and report the progress on how the model works to domain experts and business experts. They also report low-level technical issues to software developers. Finally, the strategy consultants would report back to the domain experts and business experts to evaluate and deploy the models. In particular, they prefer to use PowerPoint, Excel, and Mural to visually represent information.

4.3 Artifact Analysis: Real-World Examples

To triangulate what participants described in the interviews, we also performed an analysis on the artifacts that they shared with us. In this analysis, we reused the same codes from Table 4 under the Information Conveyance Selection and identified where those approaches were used in the artifacts. Additionally, we wanted to provide real-life examples to the research community of the communication approaches taken to overcome communication gaps across roles.

Figure 3 shows a few example slides drawn from slide deck artifacts that participants shared with us. These decks were created by AI developers to educate stakeholders about AI concepts and to provide status updates on model development progress. We selected these slides to share because they contain examples of how AI developers convey information in different ways. Note that we have modified these slides from their original versions due to confidentiality concerns. Slides are referenced according to the AI development stage in which they occur and which slide from the set they are in. For example, in Figure 3, Slide 2-2 describes the second slide from the Model Evaluation Phase. The collection of the slides is not comprehensive and does not include all the stages with involving different roles across multiple teams (i.e., data requirements, handoff). We focused on only the three stages because those are the stages that data scientists were engaged to share *their* artifact instead of collecting requirements or dealing with logistics.

In the interviews, participants described the effort they took in creating specialized visualizations to express various kinds of information. One way was to explain data science concepts using multiple strategies. Slides 2-1 and 3 show how a confusion matrix can be used to explain the concepts of recall or true/false negatives. Additionally, in the upper right of Slide 2-2, a simple table visualization provides a quick explanation of how features in a model changed from binary to non-binary. Feature changes were also visualized in terms of how they influenced the quality of the model. For example, the right half of Slide 2-1 also compares how a different models features



1 Education Phase

Updates to the algorithm increases data set size by 600%, enables **25% more matches**, and makes better recommendations

More matches

147 more matches on a given day, from 592 to 739**

More actionable matchings

115 additional exact matches** (by extending features set)

Match type	F1	F2	F3.
Possible match 1	A	1	0.12
Possible match 2	B	3	0.13

Match categories	%
Category 1	0.88

**Values as of one year ago

3 Model Deployment Phase

When the data examples has incorrect or missing data, the API identifies that there is no additional value to extract

- 24 data examples do not contain the correct data nor the necessary supplemental data
- The percentage of examples with missing data is higher than estimated
- The model is currently operating at 100% precision

Testing Analysis Breakdown

		Manual assign	
		True	False
Predicted	True	22	0
	False	19	25

* Based off 66 test cases

Incorrect Example

Model identifies example that is not correct

Correct No Match

Model identifies that example does not contain correct data

2 Model Evaluation Phase

In initial testing, the enhanced model increased the recall performance by 12%

Enhancing model, with new features, to increase the Match performance

	Description	Example
Feature Engineering	**New features:** • Feature type count • Non-binary features • Feature #5	I0 I1 I2 I3 I4 I5 I6 1 0 1 1 0 0 1 → Binary F1 F4 F1 F1 F3 F3 F1 → Non-binary
Normalization	**Rescaled the data** : Most ML algorithms take into account only the magnitude of the measurements (not the units). The Min-Max Scaler $\frac{x_i\text{-}min(x)}{max(x)\text{-}min(x)}$	a_2 = 3 m a_1 = 300 cm

Fig. 3. Examples of presentation in AI development workflow: (1) Education Phase: The AI team explains basic level of AI knowledge and potential benefits that integrating AI will bring to stakeholders; (2) Model Evaluation Phase: The AI team reports their progress once in a while to stakeholders and reminds them information of AI metrics and presents their model's performance; (3) Model Deployment Phase: After a few months of collaboration and education, the stakeholders can now interpret AI performance easily. The AI team also shares plans moving forward (Figure has been modified by authors to remove identifiable and confidential information).

changed, and also showcases how much impact each feature has in the final prediction. Finally, visualization can aid comparison as in Slide 1 which presents two examples from two different datasets and explains how they both ultimately generate the same prediction.

In some cases, a simple picture conveyed the information for the AI developer. In Slide 2-1, on the bottom-right, the analogy of a scale provides an abstraction for the role that normalization plays in this particular model. Images can likewise be an effective recall mechanism and help trigger memories. The small descriptive variant of the confusion matrix on the left side of Slide 2-1 was used in such a way. Strat1 explained to us that he would intentionally repeat it in his slides so that people recalled prior discussions about it. These creative uses of images highlight ways the AI team's strategies in overcoming communication gaps.

Outside the specific visual aspects of the slide, we noticed that each slide was highly targeted. Slides were designed to only deliver a single message that was typically contextualized in the problem domain. This stands out in headings of the slides and the other explanatory text that they contain as in Slide 3.

The variety of approaches on display over these four slides helps to demonstrate how precisely targeted and carefully crafted information is by AI developers, before it is used as a tool to cross a communication gap. The team must mold its approach to best suit its particular audience.

Our findings for the artifact analysis align with what participants talked about in the interviews. There is no one-size-fits-all approach here. Designing information content was time-consuming, yet necessary work as it is critical to building an effective SMM and to the project's success. Not all content needed to be bespoke, however. As is the case in Slide 2-1 with the reminder version of the confusion matrix, we also identified some evidence of reuse. For example, the confusion matrices themselves were used across multiple presentations, even if they were used to answer slightly different questions. Participants also stated that they would reuse explanatory content from past artifacts, adapting the content to fit their current context. Yet, such reuse relied on participants recalling ideas from the past, as none of the tools provided affordances to search for these kinds of explanations. Given the small scope of this case study, we cannot generalize beyond these participants, but at least for these observations, AI developers tended to iteratively customize a piece of information until the communication gap is crossed, and then if they remembered, reuse it if possible in the future.

5 DISCUSSION

Our research found that AI developers face various communication challenges and devise their own specific responses to these challenges. We discuss what are the best current practices that we observed during our study, where tools can help improve the collaboration experience, and how machine teaching and automated ML can lead to efficient communication among teams. We also describe the limitations and future work of this study.

5.1 Best Practices in AI Development Workflow

In this section, we consolidate observations regarding the best practices that AI developers observed during our study. To understand best practices, we explain emergent practices through a lens of SMM. To better understand why the best practices were successful, we borrow from Yu and Petter's work on how Agile development practices map to successful mental models [71] which are defined in the top half of Figure 4. We then frame AI developers' best practices from this perspective to provide additional explanation for why these practices are successful as shown in the bottom half of Figure 4.

5.1.1 Longer Planning Period than Actual Implementation. Software development teams tend to spend extensive time on planning and brainstorming [40]. Although AI application development is known to be unique and different from software engineering [2], we discovered a commonality between two domains at a high-level. Both ML and software engineering spend a great deal of time on planning and brainstorming. Through this period, AI teams are able to learn about domain knowledge and adjust their implementation accordingly (self-correction):

> *"We spent around four months just doing the query work, understanding those business scenarios, and then we held a couple interviews with sales rep to confirm our observations. Based on the knowledge, ... we started to do initial models... it's more like an incubation. Then, we held what we called garage session— a three-day workshop with assets team."* [Data3]

5.1.2 Shared Documentation Effort to Build External Knowledge. As we stated earlier, managing shared information storage for team members is key to successful communication. It was shown

Table 4. A list of codes used for best practices (Top) and identified practices and corresponding SMM practices and components (Bottom)

SMM practice	**Definition**
Planning	In the beginning of the project, team members spend time to brainstorm and define a goal of the project
Team-interaction training	Team members share embedded information in team tasks
Reflexivity	*Team members* work together to reflect upon previous experience and mistakes
Self-correction training	*Individual team member* works to reflect upon previous experience and mistakes

Practice	**SMM practice**	**SMM components**
Spending time to understand the problem and iterate	planning, self-correction training	task procedure, task strategies, environmental constraints
Capturing frequently asked questions by stakeholders	team-interaction training	information sources
Documenting domain knowledge in data	team-interaction training	information flow, communication channel
Running small pilot studies to evaluation models in development	self-correction training, team-interaction training	task procedure, task strategies
Rolling from one business sector to another	reflexivity	task procedure, task strategies

that (1) it is helpful to answer repeated questions from stakeholders (2) it becomes an effective communication channel to bridge between teams.

It is worth noting how well documentation is utilized across different teams in our AI development teams — unlike previous literature claimed. The research literature revealed that data scientists do not use documentation to record their data-management or project plans [44], and that they do not use documentation systematically throughout a data science lifecycle [72], and that they do not use documentation resources enough [48].

We believe this difference occurs due to the team structures of our participants: two independent teams that have not previously collaborated, but they form a team in an ad-hoc manner to integrate ML and AI in their business model. Therefore, the two teams have not fully understood each others' expertise or established collaboration protocols. By using documentation as a conversation tool, teams can have contextualized and informative communication, and at the same time conduct knowledge management (team-interaction training).

5.1.3 Rolling from Small Pilot Studies to Bigger Projects. We observed that AI teams follow agile practices in software engineering. Data scientists try to minimize time to deliver an "executable" version of the model to stakeholders. The rationale appear to be that this helps AI teams to keep stakeholders in-the-loop and gain insights to tweak model behavior (self-correction). Additionally, it is also an opportunity to showcase their progress and the ability of the model: *"The goal is to understand their experience and make sure they [stakeholders] feel like our optimum (value) is correct and feel they are confident using the model,"* [Data2].

Once an AI team successfully launches a model, they slowly expand the model from one sector of the business to others. As the model is stable in one sector, they can streamline the expanding process based on previous experiences in modeling (reflexivity).

5.2 Design Implications for Collaboration Tools

Our case study reveals unique challenges in the ability of AI teams/devs to communicate across project roles. Consequently, we propose the following improvements to current collaborative tools.

5.2.1 Customizing Documentation to Enable Cross-Role Communication. Information and how it is presented needs to be highly tailored for the roles receiving that information or, simply put, one size does not fit all. Participants spent a lot of their time creating visualizations, selecting examples and creating stories to share information effectively. Documentation tools that capture and share information to different roles along the AI lifecycle likewise need to tailor their content and presentation to the needs of the role. Existing efforts around documenting aspects of the AI lifecycle in dataset documentation [16] and model service documentation [4, 35] are fairly static and do not consider questions of how to dynamically vary the content that is displayed for the role in question. One exception is Hind et al.'s formative work on difficulties users have creating such documentation, which also advocated for user-centric output [18]. Providing a tables of statistics or visualizations without context is not going to address the needs of these users. To facilitate cross-role communication on these teams, documentation tools need to be aware of how these roles process information, what information they care about and the best ways to share this information with them.

Currently, data scientists are left adapting imperfect tools to their needs. For example, each participant used PowerPoint for communication or educational needs. Yet participants also made clear that PowerPoint is not well-suited to creating and maintaining complete and timely documentation. Instead, participants described using it to capture a snapshot of the current state of the work. Recall that creating that snapshot can be a non-trivial amount of work. Strat1 said it took "2–3 days" to create slides.

So why do they use PowerPoint given its high cost? Although we did not ask explicitly, the interviews with participants suggest that the flexibility and familiarity of PowerPoint may be part of the reasons why. This may be particularly important for the transfer of knowledge from AI Developers to Stakeholders - especially if the Stakeholders are planning to re-use content from the AI Developers' presentation. We remember that Data1 reported that, *"...the major outcome [was] just try to get management up to speed... and hopefully also get them excited about it..."* If materials are going to management, who may send those materials on to other management, then the use of a generic presentation tool may be strategic.

We speculate that despite the sometimes high cost of translating content to slides, the benefit of using slides is much higher still. This high-cost situation suggests a need for closer integration between data science tools and communication tools, echoing prior literature [72]. One example of such integration is Callisto which brings together chat and computational notebooks [59], but more research is needed in this direction.

5.2.2 Towards Automatically Shared Mental Model Updates. Our interview study found that AI developers mainly shared the implementation progress through an external medium (e.g., PowerPoint, PDF document) other than their working environment (e.g., code editors, computational notebooks). Not only is it a time-consuming process for them to decide what to present and how to best deliver the results, it is impossible for them to keep up to date with the rapid changes that occur during the AI development pipeline. One possible solution to address this issue is through the use of automatic summarization tools built on recent advances on code summarization and explainable artificial intelligence. Such tools could monitor inter-team changes to code and other information repositories to keep team members up to date with the latest changes. SMMs suggests that enhanced visibility may enable better collaboration since everyone would be more aware of the recent changes; however, there would still be open questions on perceptions of trust, role-specific approaches to filtering and presenting information, and how such tooling can be designed as to not be yet another distraction that teams have to deal with.

5.2.3 Beyond "Game of Telephone": Re-imagine Efficient Communication without Mediators. We observed that communications among AI teams often involve indirect feedback for AI models. Conversation involve many different roles.[4] For example, in the project requirement stage, domain experts explain the domain to strategy consultants, which strategy consultants later *translate* to data scientists. Hou et al. [19] identified this role and referred to this human intermediary as a *coordination broker*, who bridges between technical and non-technical roles.

This generates a situation of the *Game of Telephone* – i.e., additional mediators have to be involved to incorporate their domain knowledge [10, 21, 45]. We envision that with the right tools and education, we can make the conversation more efficient and avoid the game of telephone. Recent work in AutoML and machine teaching makes promising progress [9, 60, 63, 66]. With this technology, non-ML experts can build and provide insight into models without additional mediators.

5.3 Threats to Validity, Limitations, and Future Work

Given the small scope of this study and its qualitative research nature, the findings from this paper are preliminary. We caution the readers to consider the following limitations, if they plan to generalize the paper's findings outside the investigated context.

We recruited informants only within a single company (IBM), where the first author works. Because our interviews involved private and confidential project details, the informants needed this privacy protection in participating in this study, as we have noted earlier in section 3.1. The results reported in this paper may be biased by this restriction of informants recruited. AI teams at other organizations may be structured differently; have a different composition of roles; or may be part of the Stakeholder team themselves. Such teams may not face some of the communication gaps we described or may face different ones entirely. Future work may explore whether the same types of issues would be seen in other types of organizations outside the team that we studied.

Another limitation is that we had only four interviews and 10 interview sessions. Despite the depth of each of those interviews, our findings should be considered formative, until further studies with a larger and different informant sample reaffirm our findings. Ideally, we would also like to triangulate through methods such as large-scale surveys of teams and/or on-site or online ethnography method. If we collect enough data on both successful and failed communication practices, maybe we can even apply a machine learning algorithm to learn what practices lead to successful communication.

Furthermore, we acknowledge that communication in an AI project is bi-directional, where both AI developers and stakeholders may drive and participate different parts of communication in an AI project [72]. However, the people whom we interviewed represent only one perspective on the projects that they discussed. Thus, we remind the reader that we presented a one-sided perspective

[4] We attached the team composition in the Appendix A.

on the problems represented. Future work should focus on the complementary perspectives from the Stakeholder teams.

Last but not least, we are also aware that a theoretical lens both magnifies items in its focus, and marginalizes other items. Our choice of SMMs was useful to clarify certain issues, but we anticipate that a future study may be able to use multiple theory-lens for a broader view of communication issues. As an example of the selected model's limitation, our interviews did not explicitly tag along the SMM's coordination principal. This could have been an effect of our interview design or our focus on the model evaluation and deployment phases. We speculate that much of the coordination activities are established in earlier phases such as project requirements and data requirements and gathering as these phases represent the earliest interactions between the AI team and the Stakeholder team [72]. Had we focused our study on these earlier phases, it is reasonable to expect that more coordination-principle-related details would have emerged. In the future, we also plan to use common ground theory [12] and theories of small group activity, as recently reviewed by Lee and Paine [28] and by Stahl [50].

Despite these listed limitations, we believe that our results and our methodology in this paper will be useful for future work. As Paul Dourish argued [14], a qualitative work's validity and power lies together with the other qualitative works (e.g., [19, 32, 38]). These various accounts from different contexts and different perspectives can form a systematic understanding of the communication practices in AI teams.

6 CONCLUSION

In this paper, we presented a case study on how AI developers overcome communication challenges across roles in the AI development lifecycle. With the aid of a shared mental model lens, we identified the key communication gaps that AI developers faced. These gaps had an overarching theme of educating others and specifically were about gaps in knowledge, establishing trust and setting expectations. We also shed light on how and why AI developers cross communication gaps. We extracted rich details from the interviews that highlighted the many different information requests that AI developers face and also shed light on how they approach and fulfil them. Together these results informed some best practices in AI development workflow and provide important insights to researchers and designers who are interested alleviating some of the communication challenges that AI developers face. Until then, all participants can do is wait for a magic tool to help them overcome their communication gaps.

> *"I've been working with this project for so long, I could easily kind of dump [information about] it out. It's just a matter of how do you go about this? ... if there was some kind of template or tool to show me exactly what I need to make, ... that would have saved me a lot of brainstorming."* [Strat1]

REFERENCES

[1] Piotr D Adamczyk and Michael B Twidale. 2007. Supporting multidisciplinary collaboration: requirements from novel HCI education. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 1073–1076.

[2] Saleema Amershi, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, and Thomas Zimmermann. 2019. Software engineering for machine learning: A case study. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. IEEE, 291–300.

[3] Cecilia Aragon, Clayton Hutto, Andy Echenique, Brittany Fiore-Gartland, Yun Huang, Jinyoung Kim, Gina Neff, Wanli Xing, and Joseph Bayer. 2016. Developing a research agenda for human-centered data science. In *Proceedings of the 19th ACM Conference on Computer Supported Cooperative Work and Social Computing Companion*. ACM, 529–535.

[4] Matthew Arnold, Rachel KE Bellamy, Michael Hind, Stephanie Houde, Sameep Mehta, A Mojsilović, Ravi Nair, K Natesan Ramamurthy, Alexandra Olteanu, David Piorkowski, et al. 2019. FactSheets: Increasing trust in AI services through

supplier's declarations of conformity. *IBM Journal of Research and Development* 63, 4/5 (2019), 6–1.

[5] Anders Arpteg, Björn Brinne, Luka Crnkovic-Friis, and Jan Bosch. 2018. Software engineering challenges of deep learning. In *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, 50–59.

[6] John R Austin. 2003. Transactive memory in organizational groups: the effects of content, consensus, specialization, and accuracy on group performance. *Journal of applied psychology* 88, 5 (2003), 866.

[7] Anant Bhardwaj, Souvik Bhattacherjee, Amit Chavan, Amol Deshpande, Aaron J Elmore, Samuel Madden, and Aditya G Parameswaran. 2014. Datahub: Collaborative data science & dataset version management at scale. *arXiv preprint arXiv:1409.0798* (2014).

[8] Carrie J Cai and Philip J Guo. 2019. Software Developers Learning Machine Learning: Motivations, Hurdles, and Desires. In *2019 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 25–34.

[9] José P Cambronero and Martin C Rinard. 2019. AL: autogenerating supervised learning programs. *Proceedings of the ACM on Programming Languages* 3, OOPSLA (2019), 1–28.

[10] Arunima Chaudhary et al. 2021. AutoText: An End-to-End AutoAI Framework for Text. In *AAAI*. ACM.

[11] Herbert H Clark. 2006. Context and common ground. *Concise Encyclopedia of Philosophy of Language and Linguistics* (2006), 85–87.

[12] Herbert H Clark and Susan E Brennan. 1991. Grounding in communication. (1991).

[13] Sharolyn Converse, JA Cannon-Bowers, and E Salas. 1993. Shared mental models in expert team decision making. *Individual and group decision making: Current issues* 221 (1993), 221–46.

[14] Paul Dourish. 2014. Reading and interpreting ethnography. In *Ways of Knowing in HCI*. Springer, 1–23.

[15] Xiangmin Fan, Daren Chao, Zhan Zhang, Dakuo Wang, Xiaohua Li, and Feng Tian. 2020. Utilization of Self-Diagnosis Health Chatbots in Real-World Settings: Case Study. *Journal of Medical Internet Research* 22 (2020).

[16] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumeé III, and Kate Crawford. 2018. Datasheets for datasets. *arXiv preprint arXiv:1803.09010* (2018).

[17] Charles Hill, Rachel Bellamy, Thomas Erickson, and Margaret Burnett. 2016. Trials and tribulations of developers of intelligent systems: A field study. In *2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 162–170.

[18] Michael Hind, Stephanie Houde, Jacquelyn Martino, Aleksandra Mojsilovic, David Piorkowski, John Richards, and Kush R Varshney. 2020. Experiences with Improving the Transparency of AI Models and Services. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–8.

[19] Youyang Hou and Dakuo Wang. 2017. Hacking with NPOs: collaborative analytics and broker roles in civic data hackathons. *Proceedings of the ACM on Human-Computer Interaction* 1, CSCW (2017), 53.

[20] Adrian Jones. 2006. Multidisciplinary team working: Collaboration and conflict. *International Journal of Mental Health Nursing* 15, 1 (2006), 19–28.

[21] Yannis Katsis and Christine T. Wolf. 2019. ModelLens: An Interactive System to Support the Model Improvement Practices of Data Science Teams. In *Conference Companion Publication of the 2019 on Computer Supported Cooperative Work and Social Computing* (Austin, TX, USA) *(CSCW '19)*. Association for Computing Machinery, New York, NY, USA, 9–13. https://doi.org/10.1145/3311957.3359512

[22] Mary Beth Kery, Marissa Radensky, Mahima Arya, Bonnie E John, and Brad A Myers. 2018. The story in the notebook: Exploratory data science using a literate programming tool. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 174.

[23] Miryung Kim, Thomas Zimmermann, Robert DeLine, and Andrew Begel. 2016. The emerging role of data scientists on software development teams. In *Proceedings of the 38th International Conference on Software Engineering*. ACM, 96–107.

[24] Marina Kogan, Aaron Halfaker, Shion Guha, Cecilia Aragon, Michael Muller, and Stuart Geiger. 2020. Mapping Out Human-Centered Data Science: Methods, Approaches, and Best Practices. In *Companion of the 2020 ACM International Conference on Supporting Group Work*. 151–156.

[25] Georgia Kougka, Anastasios Gounaris, and Alkis Simitsis. 2018. The many faces of data-centric workflow optimization: a survey. *International Journal of Data Science and Analytics* 6, 2 (2018), 81–107.

[26] Penny Lacey. 2002. 9 Multidisciplinary Work Challenges and Possibilities. *Special Education Reformed: Inclusion-Beyond Rhetoric?* (2002), 157.

[27] James R Larson, Pennie G Foster-Fishman, and Christopher B Keys. 1994. Discussion of shared and unshared information in decision-making groups. *Journal of personality and social psychology* 67, 3 (1994), 446.

[28] Charlotte P Lee and Drew Paine. 2015. From The Matrix to a Model of Coordinated Action (MoCA) A Conceptual Framework of and for CSCW. In *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing*. 179–194.

[29] M Lee, Tristan Johnson, and Myung H Jin. 2012. Toward understanding the dynamic relationship between team and task shared mental models as determinants of team and individual performances. *International journal of information*

technology and business management 8, 1 (2012), 1–14.

[30] Q. Vera Liao, Muhammed Mas-ud Hussain, Praveen Chandar, Matthew Davis, Yasaman Khazaeni, Marco Patricio Crasso, Dakuo Wang, Michael Muller, N. Sadat Shami, and Werner Geyer. 2018. All Work and No Play?. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. ACM, New York, NY, USA, Article 3, 13 pages. https://doi.org/10.1145/3173574.3173577

[31] Lucy Ellen Lwakatare, Aiswarya Raj, Jan Bosch, Helena Holmström Olsson, and Ivica Crnkovic. 2019. A taxonomy of software engineering challenges for machine learning systems: An empirical investigation. In *International Conference on Agile Software Development*. Springer, 227–243.

[32] Yaoli Mao, Dakuo Wang, Michael Muller, Kush R Varshney, Ioana Baldini, Casey Dugan, and Aleksandra Mojsilović. 2019. How Data Scientists Work Together With Domain Experts in Scientific Collaborations: To Find The Right Answer Or To Ask The Right Question? *Proceedings of the ACM on Human-Computer Interaction* 3, GROUP (2019), 1–23.

[33] Sara A McComb. 2007. Mental model convergence: The shift from being an individual to being a team member. *Research in multi-level issues* 6 (2007), 95–147.

[34] Nora McDonald, Sarita Schoenebeck, and Andrea Forte. 2019. Reliability and Inter-Rater Reliability in Qualitative Research: Norms and Guidelines for CSCW and HCI Practice. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 72 (Nov. 2019), 23 pages. https://doi.org/10.1145/3359174

[35] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. ACM, 220–229.

[36] Susan Mohammed, Lori Ferzandi, and Katherine Hamilton. 2010. Metaphor no more: A 15-year review of the team mental model construct. *Journal of management* 36, 4 (2010), 876–910.

[37] Michael Muller, Melanie Feinberg, Timothy George, Steven J Jackson, Bonnie E John, Mary Beth Kery, and Samir Passi. 2019. Human-Centered Study of Data Science Work Practices. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, W15.

[38] Michael Muller, Ingrid Lange, Dakuo Wang, David Piorkowski, Jason Tsay, Q Vera Liao, Casey Dugan, and Thomas Erickson. 2019. How Data Science Workers Work with Data: Discovery, Capture, Curation, Design, Creation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–15.

[39] Michael J Muller, David R Milien, and Jonathan Feinberg. 2009. Information curators in an enterprise file-sharing service. In *ECSCW 2009*. Springer, 403–410.

[40] Brad A Myers and Mary Beth Rosson. 1992. Survey on user interface programming. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 195–202.

[41] Soya Park, April Wang, Ban Kawas, Q. Vera Liao, David Piorkowski, and Marina Danilevsky. 2021. Facilitating Knowledge Sharing from Domain Experts to Data Scientists for Building NLP Models. arXiv:2102.00036 [cs.HC]

[42] Samir Passi and Steven J Jackson. 2018. Trust in Data Science: Collaboration, Translation, and Accountability in Corporate Data Science Projects. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 136.

[43] DJ Patil. 2011. *Building data science teams.* " O'Reilly Media, Inc.".

[44] Kathleen H Pine and Max Liboiron. 2015. The politics of measurement and action. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 3147–3156.

[45] Claudio Pinhanez. 2019. Machine Teaching by Domain Experts: Towards More Humane, Inclusive, and Intelligent Machine Learning Systems. *arXiv preprint arXiv:1908.08931* (2019).

[46] Nadeem Qazi and BL William Wong. 2019. An interactive human centered data science approach towards crime pattern analysis. *Information Processing & Management* 56, 6 (2019), 102066.

[47] Christian J Resick, Toshio Murase, Wendy L Bedwell, Elizabeth Sanz, Miliani Jiménez, and Leslie A DeChurch. 2010. Mental model metrics and team adaptability: A multi-facet multi-method examination. *Group Dynamics: Theory, Research, and Practice* 14, 4 (2010), 332.

[48] Adam Rule, Aurélien Tabard, and James D Hollan. 2018. Exploration and explanation in computational notebooks. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 32.

[49] Matthias Scheutz, Scott A DeLoach, and Julie A Adams. 2017. A framework for developing and using shared mental models in human-agent teams. *Journal of Cognitive Engineering and Decision Making* 11, 3 (2017), 203–224.

[50] Gerry Stahl. 2013. Theories of collaborative cognition: Foundations for CSCL and CSCW together. In *Computer-supported collaborative learning at the workplace*. Springer, 43–63.

[51] Margaret-Anne Storey, Li-Te Cheng, Ian Bull, and Peter Rigby. 2006. Shared waypoints and social tagging to support collaboration in software development. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*. ACM, 195–198.

[52] Margaret-Anne Storey, Alexey Zagalsky, Fernando Figueira Filho, Leif Singer, and Daniel M German. 2016. How social and communication channels shape and challenge a participatory culture in software development. *IEEE Transactions on Software Engineering* 43, 2 (2016), 185–204.

[53] Jennifer Sukis. 2019. AI Design & Practices Guidelines. https://medium.com/design-ibm/ai-design-guidelines-e06f7e92d864

[54] Christoph Treude, Margaret-anne Storey, and Jens Weber. 2009. Empirical studies on collaboration in software development: A systematic literature review. (2009).

[55] Piet Van den Bossche, Wim Gijselaers, Mien Segers, Geert Woltjer, and Paul Kirschner. 2011. Team learning: building shared mental models. *Instructional Science* 39, 3 (2011), 283–301.

[56] Wil MP Van der Aalst. 2014. Data scientist: The engineer of the future. In *Enterprise interoperability VI*. Springer, 13–26.

[57] Zhiyuan Wan, Xin Xia, David Lo, and Gail C Murphy. 2019. How does Machine Learning Change Software Development Practices? *IEEE Transactions on Software Engineering* (2019).

[58] April Yi Wang, Anant Mittal, Christopher Brooks, and Steve Oney. 2019. How Data Scientists Use Computational Notebooks for Real-Time Collaboration. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–30.

[59] April Yi Wang, Zihan Wu, Christopher Brooks, and Steve Oney. 2020. Callisto: Capturing the "Why" by Connecting Conversations with Computational Narratives. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. ACM.

[60] Dakuo Wang, Josh Andres, Justin Weisz, Erick Oduor, and Casey Dugan. 2021. AutoDS: Towards Human-Centered Automation of Data Science. In *Proceedings of the CHI 2021*.

[61] Dakuo Wang, Q. Vera Liao, Yunfeng Zhang, Udayan Khurana, Horst Samulowitz, Soya Park, Michael Muller, and Lisa Amini. 2021. How Much Automation Does a Data Scientist Want?. In *in submission*.

[62] Dakuo Wang, Liuping Wang, Zhan Zhang, Ding Wang, Haiyi Zhu, Yvonne Gao, Xiangmin Fan, and Feng Tian. 2021. Brilliant AI Doctor in Rural China: Tensions and Challenges in AI-Powered CDSS Deployment. In *Proceedings of the CHI 2021*.

[63] Dakuo Wang, Justin D. Weisz, Michael Muller, Parikshit Ram, Werner Geyer, Casey Dugan, Yla Tausczik, Horst Samulowitz, and Alexander Gray. 2019. Human-AI Collaboration in Data Science: Exploring Data Scientists' Perceptions of Automated AI. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 211 (Nov. 2019), 24 pages. https://doi.org/10.1145/3359313

[64] Hironori Washizaki, Hiromu Uchida, Foutse Khomh, and Yann-Gaël Guéhéneuc. 2019. Studying software engineering patterns for designing machine learning systems. In *2019 10th International Workshop on Empirical Software Engineering in Practice (IWESEP)*. IEEE, 49–495.

[65] Yasuhiro Watanabe, Hironori Washizaki, Kazunori Sakamoto, Daisuke Saito, Kiyoshi Honda, Naohiko Tsuda, Yoshiaki Fukazawa, and Nobukazu Yoshioka. 2019. Preliminary Systematic Literature Review of Machine Learning System Development Process. *arXiv preprint arXiv:1910.05528* (2019).

[66] Bryan Wilder, Eric Horvitz, and Ece Kamar. 2020. Learning to Complement Humans. *arXiv preprint arXiv:2005.00582* (2020).

[67] Marian G Williams and Vivienne Begg. 1993. Translation between software designers and users. *Commun. ACM* 36, 6 (1993), 102–104.

[68] Anbang Xu, Zhe Liu, Yufan Guo, Vibha Sinha, and Rama Akkiraju. 2017. A New Chatbot for Customer Service on Social Media. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. ACM, New York, NY, USA, 3506–3510. https://doi.org/10.1145/3025453.3025496

[69] Ying Xu, Dakuo Wang, Penelope Collins, Hyelim Lee, and Mark Warschauer. [n.d.]. Same benefits, different communication patterns: Comparing Children's reading with a conversational agent vs. a human partner. *Computers & Education* 161 ([n. d.]), 104059.

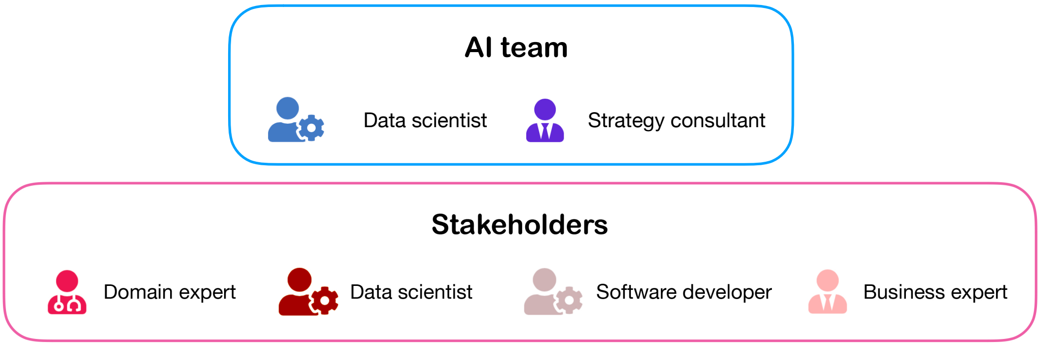
[70] Qian Yang, Aaron Steinfeld, and John Zimmerman. 2019. Unremarkable AI: Fitting Intelligent Decision Support into Critical, Clinical Decision-Making Processes. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 238.

[71] Xiaodan Yu and Stacie Petter. 2014. Understanding agile software development practices using shared mental models theory. *Information and software technology* 56, 8 (2014), 911–921.

[72] Amy X Zhang, Michael Muller, and Dakuo Wang. 2020. How do data science workers collaborate? roles, workflows, and tools. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW1 (2020), 1–23.

A TEAM COMPOSITION OF EACH TEAM

This shows our interviewees' team configurations and how different roles are involved in each stage of ML workflow. The team's composition varies due to special occasions, such as data availability, data already being labeled, and different role having responsibility across different teams. An aggregated view can be found in Figure 1.



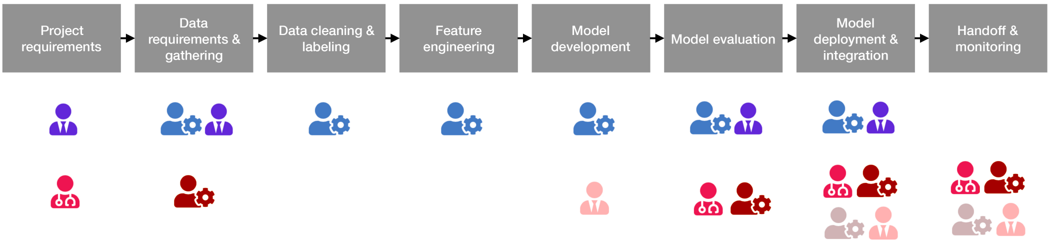


Fig. 4. Data1 team

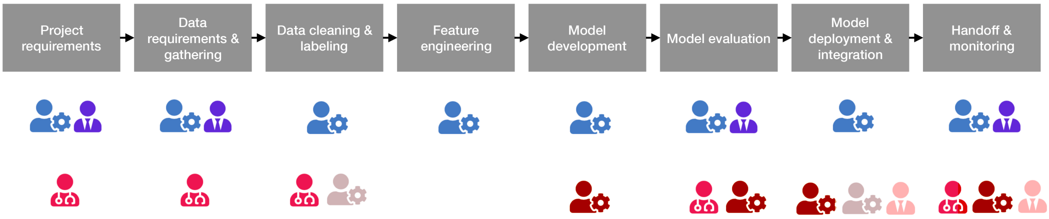


Fig. 5. Data2 team

Received June 2020; revised October 2020; accepted December 2020

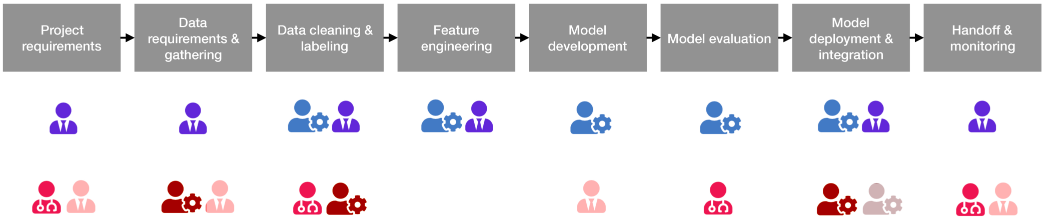


Fig. 6. Data3 team

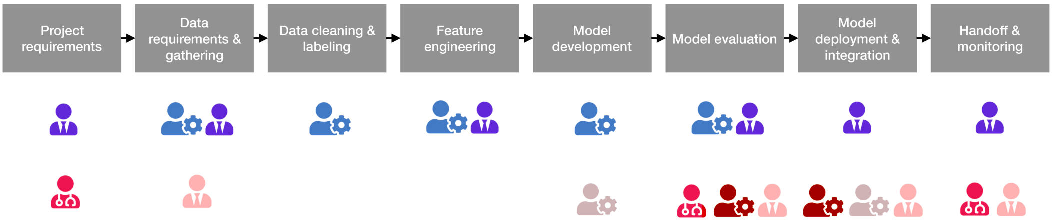


Fig. 7. Strat1 team