Pontifícia Universidade Católica
do Rio de Janeiro

Antonio Pedro Santos Alves

Requirements Engineering for ML-Enabled Systems: Status Quo and Problems

Dissertação de Mestrado

Dissertation presented to the Programa de Pós–graduação em Informática of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Informática.

Advisor : Prof. Marcos Kalinowski
Co-advisor: Prof. Daniel Méndez

Rio de Janeiro
December 2023

Pontifícia Universidade Católica
do Rio de Janeiro

Antonio Pedro Santos Alves

Requirements Engineering for ML-Enabled Systems: Status Quo and Problems

Dissertation presented to the Programa de Pós–graduação em Informática of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Informática. Approved by the Examination Committee:

Prof. Marcos Kalinowski
Advisor
Departamento de Informática – PUC-Rio

Prof. Daniel Méndez
Co-advisor
BTH, Sweden

Prof^a. Maria Teresa Baldassarre
Uniba, Italy

Prof. Helio Côrtes Vieira Lopes
PUC-Rio

Rio de Janeiro, December 6th, 2023

Antonio Pedro Santos Alves

Computer Technician from the National Industrial Learning Service of Petrópolis, Rio de Janeiro (SENAI - Petrópolis), and with bachelor degree in Computer Science from the Federal University of São João del-Rei, in São João del-Rei, Minas Gerais (UFSJ).

Bibliographic data

Alves, Antonio Pedro Santos

Requirements Engineering for ML-Enabled Systems: Status Quo and Problems / Antonio Pedro Santos Alves; advisor: Marcos Kalinowski; co-advisor: Daniel Méndez. – 2023.

52 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2023.

Inclui bibliografia

1. Informática – Teses. 2. Engenharia de Requisitos. 3. Aprendizado de Máquina. 4. Survey. I. Kalinowski, Marcos. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

To my parents, brother, and family for their unconditional love, support, and encouragement. To my friends and workmates for making the days lighter and more fun. And for my advisor, Marcos, for his valuable lessons.

Acknowledgments

First and foremost, I want to thank the Department of Informatics of PUC-Rio for giving me the opportunity to do research at a high level and the CNPq agency for sponsoring it. Without them, this would not have been possible.

I would like to thank my advisor, Prof. Dr. Marcos Kalinowski, for the lessons and assistance during my Master's program. His vision, attention, dynamism, and wisdom were of great value to me and inspired me to research even more. Thank you for the guidance and, above all, for the friendship we have built. Before seeing you as an advisor, I see you as a great friend.

Thanks to my co-advisor, Prof. Dr. Daniel Méndez, for the teachings and valuable tips for the completion of this work. I also appreciate the committee members for being available not only to evaluate this dissertation but also to contribute with all their experience to make it a high-quality research. Thank you, Prof. Dr. Helio Lopes and Prof. Dr. Maria Teresa Baldassarre.

I also want to express my gratitude to the friends who have been with me on this journey. Without the occasional beer, *açaí*, and hangouts, it would have been impossible to reach where I am. You made the days much lighter. I especially want to thank the friends I have from 'CUTUCA F.C' at UFSJ, my friends from volleyball and soccer, and lastly, my old workmates from MarkTech. Also, I would like to thank the new friends I made at PUC-Rio during the time of Americanas and ExACTa, especially the 'Clube do Quitute'. Special thanks to my English Teacher, Nathália, for her unconditional support and friendship. To my closest friends, thank you for being a source of support and trust in moments of difficulty and joy.

To my parents, brother, godparents, uncles, and grandmother, thank you for your unconditional support in the pursuit of the Master's degree. Without your sacrifices and help, I would not be here writing these words. Thank you.

Finally, I want to thank God for wisdom and for surrounding me with good and caring people on this journey. May it always be so.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Abstract

Alves, Antonio Pedro Santos; Kalinowski, Marcos (Advisor); (Co-Advisor). **Requirements Engineering for ML-Enabled Systems: Status Quo and Problems**. Rio de Janeiro, 2023. 52p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Systems that use Machine Learning (ML) have become commonplace for companies that want to improve their products, services, and processes. Literature suggests that Requirements Engineering (RE) can help to address many problems when engineering ML-Enabled Systems. However, the state of empirical evidence on how RE is applied in practice in the context of ML-enabled systems is mainly dominated by isolated case studies with limited generalizability. We conducted an international survey to gather practitioner insights into the status quo and problems of RE in ML-enabled systems. We gathered 188 complete responses from 25 countries. We conducted quantitative statistical analyses on contemporary practices using bootstrapping with confidence intervals and qualitative analyses on the reported problems involving open and axial coding procedures. We found significant differences in RE practices within ML projects, some of them have been reported on literature and some are totally new. For instance, (i) RE-related activities are mostly conducted by project leaders and data scientists, (ii) the prevalent requirements documentation format concerns interactive Notebooks, (iii) the main focus of non-functional requirements includes data quality, model reliability, and model explainability, and (iv) main challenges include managing customer expectations and aligning requirements with data. The qualitative analyses revealed that practitioners face problems related to lack of business domain understanding, unclear requirements, and low customer engagement. These results help to provide a better understanding of the adopted practices and which problems exist in practical environments. We put forward the need to adapt further and disseminate RE-related practices for engineering ML-enabled systems.

Keywords

Requirements Engineering; Machine Learning; Survey.

Resumo

Alves, Antonio Pedro Santos; Kalinowski, Marcos; . **Engenharia de Requisitos para Sistemas Integrados com Componentes de Aprendizado de Máquina: Status Quo e Problemas**. Rio de Janeiro, 2023. 52p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Sistemas que usam Aprendizado de Máquina, doravante *Machine Learning* (ML), tornaram-se comuns para empresas que deseajam melhorar seus produtos, serviços e processos. A literatura sugere que a Engenharia de Requisitos (ER) pode ajudar a explicar muitos problemas relacionados à engenharia de sistemas inteligentes envolvendo componentes de ML (*ML-Enabled Systems*). Contudo, o cenário atual de evidências empíricas sobre como ER é aplicado na prática no contexto desses sistemas é amplamente dominado por estudos de casos isolados com pouca generalização. Nós conduzimos um *survey* internacional para coletar informações de profissionais sobre o *status quo* e problemas de ER para *ML-Enabled Systems.* Coletamos 188 respostas completas de 25 países. Realizamos uma análise quantitativa sobre as práticas atuais utilizando *bootstrapping* com intervalos de confiança; e análises qualitativas sobre os problemas reportados através de procedimentos de codificação *open* e *axial.* Encontramos diferenças significativas nas práticas de ER no contexto de projetos de ML, algumas já reportadas na literatura e outras totalmente novas. Por exemplo, (i) atividades relacionadas à ER são predominantemente conduzidas por líderes de projeto e cientistas de dados, (ii) o formato de documentação predominante é baseado em *Notebooks* interativos, (iii) os principais requisitos não-funcionais incluem qualidade dos dados, confiança e explicabilidade no modelo, e (iv) os principais desafios consistem em gerenciar a expectativa dos clientes e alinhar requisitos com os dados disponíveis. As análises qualitativas revelaram que os praticantes enfrentam problemas relacionados ao baixo entendimento sobre o domínio do negócio, requisitos pouco claros e baixo engajamento do cliente. Estes resultados ajudam a melhorar o entendimento sobre práticas adotadas e problemas existentes em cenários reais. Destacamos a necessidade para adaptar ainda mais e disseminar práticas de ER relacionadas à engenharia de *ML-Enabled Systems.*

Palavras-chave

Engenharia de Requisitos; Aprendizado de Máquina; Survey.

Table of contents

List of figures

List of tables

List of Abreviations

BDD – Behavior-Driven Development

CNN – Convolutional Neural Network

CRISP-DM – Cross Industry Standard Process for Data Mining

GORE – Goal-Oriented Requirements Engineering

ILSVRC – ImageNet Large-Scale Visual Recognition Challenge

ML – Machine Learning

NFRs – Non-Functional Requirements

RE – Requirements Engineering

RQ – Research Question

SAS – Self-Adaptative-Systems

SE - Software Engineering

UML – Unified Modeling Language

If I have seen further, it is by standing on the shoulders of giants.

Isaac Newton, *Letter excerpt.*

1
Introduction

1.1
Context and Motivation

At the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) of 2012, the world was astonished by a Machine Learning (ML) model halving the second-best error rate on a very difficult computational task: image classification (KRIZHEVSKY; SUTSKEVER; HINTON, 2017). Nowadays, ML models have surpassed even human performance and reached a level where the classification task is essentially solved (LUNDERVOLD; LUNDERVOLD, 2019). In fact, with the increase of computational power, ML can nowadays deal with larger and more difficult problems, outperforming other approaches also in different tasks, such as natural language processing (PETERS et al., 2018), speech recognition, and synthesis tasks (XIONG et al., 2018). With companies noticing the potential that ML could have on their products and services, having ML components as a part of a larger system, ML-enabled systems became more and more commonplace.

However, even big tech companies like Google and Microsoft report challenges related to building reliable and maintainable ML-enabled systems, not only for the inherent ML coding complexity but also due to difficulties concerning requirements (SCULLEY et al., 2015; KIM et al., 2017). Indeed, the shift from engineering conventional software systems to ML-enabled systems comes with challenges regarding idiosyncrasies of such systems, such as addressing additional qualities properties (*e.g.,* fairness and explainability), dealing with a high degree of iterative experimentation, and facing unrealistic assumptions (VILLAMIZAR; ESCOVEDO; KALINOWSKI, 2021; NAHAR et al., 2023). Furthermore, the non-deterministic nature of ML-enabled systems poses challenges from the viewpoint of Software Engineering (SE) (GIRAY, 2021).

Literature suggests that Requirements Engineering (RE) can help address problems related to engineering ML-enabled systems (VOGELSANG; BORG, 2019; VILLAMIZAR; ESCOVEDO; KALINOWSKI, 2021; AHMAD et al., 2021). However, research on this intersection mainly focuses on using ML techniques to support RE activities rather than exploring how RE can improve the development of ML-enabled systems (DALPIAZ; NIU, 2020). It is worth mentioning that, on traditional software, RE is intrinsically volatile and complex (FERNÁNDEZ et al., 2017), and in the context of ML-enabled systems development, it is typically the most difficult activity (ISHIKAWA; YOSHIOKA, 2019). This difficulty is increased by the current state of empirical evidence on how RE is applied in practice in the context of such systems, which is still weak and dominated by isolated studies.

1.2 Goal

The goal of this dissertation is to investigate the state of practice and problems of RE for ML-enabled systems, helping to steer future research on the topic. In particular, we aim to provide insights regarding (i) roles that are typically in charge of requirements, (ii) which requirements are typically elicited and documented, (iii) which non-functional requirements typically play a major role in ML-enabled systems development, (iv) which RE activities are perceived as most difficult, and lastly (v) what RE-related problems do ML practitioners face.

1.3 Research Method

In order to achieve our goal, we analyze data from an international survey conducted to characterize the pain in developing ML-enabled systems. In this dissertation, we focused on the survey's RE-related questions. In total, 188 practitioners from 25 countries completely answered the survey. Based on practitioners' responses, we conducted quantitative and qualitative analyses, where we adopted bootstrapping techniques with confidence intervals to strengthen statistical validity (LUNNEBORG, 2001; WAGNER et al., 2019) on quantitative analysis, and we used open and axial coding procedures from Grounded Theory (STRAUSS A.; CORBIN, 2009) on qualitative analysis.

1.4 Results

Our results revealed important differences in how RE is being performed in the ML-enabled system context. For instance, requirements engineers do not play a representative role within the ML-enabled system context in which RE-related activities are typically performed by project leaders and data scientists. Furthermore, requirements are being mainly documented within interactive Notebooks, followed closely by user stories and simple requirements lists. The prevalence of Non-Functional Requirements (NFRs) within these systems includes specific ML-NFRs, such as data quality, model reliability, and model explainability. Moreover, we also revealed difficulties in managing customer expectations and in aligning requirements with data. The main reported problems are comparable to those in traditional RE, including the lack of problem and business domain understanding, unclear goals and requirements, low customer engagement, and problems with managing expectations and communication.

1.5 Outline

The remainder of this work is organized as follows. Chapter 2 provides the background and an overview of related work.

Chapter 3 describes the goal and research questions that guided this research. Therefrom details the methodology behind our analyses and how the bootstrapping technique, axial and open coding were applied.

Chapter 4 presents the results, while Chapter 5 presents the threats to the validity of our study and discusses the obtained results and their practical implications.

Finally, Chapter 6 contains the concluding remarks and describes future work possibilities.

2
Background and Related Work

2.1
Introduction

Software Engineering plays a fundamental role when developing reliable, maintainable, and functional systems. In this context, Requirements Engineering is extremely important due to its contribution to the success of software, although its customer dependency, volatility, and interdisciplinary nature turn it into a very difficult discipline to investigate (FERNÁNDEZ et al., 2017). RE has been characterized by its uncertainty and the participation of interdisciplinary stakeholders in order to meet the customers' needs on the developed software (WAGNER et al., 2019).

The usage of ML-enabled systems has grown considerably in recent years, resulting in increasing demands for high-quality within this context (VILLAMIZAR; ESCOVEDO; KALINOWSKI, 2021), despite the difficulty of building reliable and maintainable ML-enabled systems (SCULLEY et al., 2015). ML involves algorithms that analyze data to create models capable of making predictions on unseen data (MITCHELL, 1997). Thus, unlike traditional systems, ML-enabled systems learn from data instead of being programmed with predefined rules. This shift changes the way of designing such kind of system once poor-quality requirements (*e.g.*, data, customer expectations, quality metrics) can lead to inaccurate results, and RE is an important tool in this scenario. An example comes from Microsoft, where the main actors in ML development are Data Scientists, and despite their high education level - more than 40% professionals have a master's degree - they struggle not only with algorithms and scale, which are inherent to ML-enabled systems but mostly with RE challenges (KIM et al., 2017).

Therefore, in this chapter, we present an overview of the difficulties and challenges reported on RE for ML-enabled systems, as well as the main contributions on the topic.

2.2
RE for ML-Enabled Systems

RE and ML have a special connection. An ML model can be seen as a requirements specification based on training data since the data can be seen as a learned description of how the ML model shall behave (KAESTNER, 2020). In this manner, when developing ML models, we need to identify relevant and representative data, validate models, and balance model-related user expectations (*e.g.*, accuracy versus inference time), just as in RE for traditional systems where we need to identify representative stakeholders, validate specifications with customers, and address conflicting requirements. Thus, despite being known and having the same essence when we think about traditional software, handling requirements on intelligent systems is still the most difficult activity for ML development (ISHIKAWA; YOSHIOKA, 2019), and the reasons for it vary.

First of all, ML-enabled systems assume that there will be an ML model that solves a problem, but how do professionals should evaluate it? Source code review and exhaustive coverage tests, which are commonly used in testing phases, are not appropriate to it once they are intrinsically hard to interpret (BORG et al., 2019). Moreover, the expected steps for handling requirements in traditional software, such as analysis and specification in preliminary development phases and acceptance at the final phases, are not possible in such systems given that this would demand an estimate of different metrics (*e.g.*, accuracy) in advance (ISHIKAWA; YOSHIOKA, 2019). Another challenge is regarding NFRs for these systems. There is no holistic view on NFRs for ML-enabled systems as most of the research focuses on individual ones, such as privacy, processing time, and data quality (HORKOFF, 2019). ML-enabled systems also face problems regarding specification, elicitation, validation, and documentation (KUWAJIMA; YASUOKA; NAKAE, 2020).

Agent-based SE and Goal-Oriented Requirements Engineering (GORE) research are potential contributors to overcome these difficulties (BELANI; VUKOVIĆ; CAR, 2019), as well as contributions from RE for SAS, which are data-based systems and face some similar data difficulties, such as delayed software decisions due to data availability (KEPHART; CHESS, 2003; MORANDINI et al., 2017). Nevertheless, the similarities regarding some common difficulties, RE for ML still faces its own challenges (CHALLA; NIU; JOHNSON, 2020; LWAKATARE et al., 2020; MARTÍNEZ-FERNÁNDEZ et al., 2022). In this sense, we can decompose the main contributions of RE for ML-enabled systems into theoretical studies and industrial findings.

2.2.1 Theoretical Studies

It is worth mentioning how ML is useful to SE activities, however, it is less common to see studies tackling the opposite, how SE can be useful to engineer ML-enabled systems (KUMENO, 2019; NASCIMENTO et al., 2020; LORENZONI et al., 2021; MARTÍNEZ-FERNÁNDEZ et al., 2022). Focusing on RE, we observe the same behavior. Challa *et al.* (CHALLA; NIU; JOHNSON, 2020) is one of the studies that showed concern regarding requirements and project success, in fact, how faulty requirements led to unexpected results in deep learning systems.

Ahmad *et al.* (AHMAD et al., 2021) and Villamizar *et al.* (VILLAMIZAR; ESCOVEDO; KALINOWSKI, 2021) provided insightful results about practices and challenges regarding RE for ML-enabled systems through systematic literature reviews. Ahmad *et al.* (AHMAD et al., 2021) found 27 papers discussing major modeling languages in use to handle requirements, application domains, and main limitations on RE for ML. Papers that were empirical studies were focused on Autonomous Driving, Computer Vision, Fraud Detection, and the Medical domain, while non-empirical studies (*e.g.*, theoretical studies and not yet evaluated) tackled ethics, trust, and explainability domains. Regardless of the domain, most studies mainly use UML and GORE as modeling languages to handle requirements. Finally, the main limitations and outstanding challenges reported on RE for ML were in terms of the overconfidence when using AI, vagueness and complexity when defining requirements, and the trade-off between NFRs that must be prioritized (*e.g.*, model reliability versus model transparency).

Villamizar *et al.* (VILLAMIZAR; ESCOVEDO; KALINOWSKI, 2021) found 35 papers discussing requirements for ML. Most of them focused their contributions on analysis and approaches to deal with requirements, with special attention on the challenges of defining business goals and problem understanding. Also, fundamental NFRs for ML-enabled systems that are not commonly addressed in traditional software development were presented, such as security, explainability, data quality, fairness, and transparency. Lastly, challenges that are preventing progress in the area of RE for ML were discussed, such as the lack of validation techniques and difficulty in dealing with customer expectations.

The difference between what is being studied by researchers and what is being used in practice tends to be harmful for SE practical adoption (JACOBSON; MEYER; SOLEY, 2009; JACOBSON; SPENCE, 2009), and this extends to the RE context. Besides highlighting difficulties and gaps to be filled on RE for ML-enabled systems, both Ahmad *et al.* (AHMAD et al., 2021) and Villamizar *et al.* (VILLAMIZAR; ESCOVEDO; KALINOWSKI, 2021) warned about most papers being validation research or opinion papers, and few of them proposing practical solutions. Having a wider picture of current industrial practices and challenges could help to overcome this lack by clarifying the current needs of practitioners engineering ML-enabled systems (LWAKATARE et al., 2020).

2.2.2 Industrial Findings

From an industry perspective, ML coding is a small part of several other steps of ML-enabled system engineering (SCULLEY et al., 2015), and RE is a potential tool to accelerate the process of maturing ML development and industrial AI adoption (SCHARINGER et al., 2022). In this way, gathering empirical evidence from the industry is essential to bridge the gap between theory and practice. Collecting practitioners' insights becomes imperative to identify real-world challenges and current practices accurately. Such studies can provide a better understanding of the practical problems that can guide the advancement of new RE for ML techniques and their effective implementation in real-world scenarios. In the following, we present studies conducted within industry settings involving practitioners to understand RE for ML.

Vogelsang and Borg (VOGELSANG; BORG, 2019) conducted interviews with four data scientists to find out the current practices and what should be done to handle and surpass the challenges regarding requirements. They suggest the need for new RE for ML solutions or at least the adaptation of existing ones. Habibullah *et al.* (HABIBULLAH; GAY; HORKOFF, 2023) conducted interviews and a survey to understand how NFRs are perceived among ML practitioners. They identified the degree of importance practitioners place on different NFRs, explored how NFRs are defined and measured, and identified associated challenges.

Recently, Scharinger *et al.* (SCHARINGER et al., 2022) revealed the worries at Siemens regarding problems that any ML project is susceptible to, listing *ML Pitfalls*, such as lack of decision quality baselines and underestimating costs. They believe that RE is the key to both avoiding these pitfalls and ripening ML development. Lastly, Nahar *et al.* (NAHAR et al., 2023) identified challenges in building ML-enabled systems through a systematic literature survey aggregating existing studies involving interviews or surveys with practitioners of multiple projects. With respect to RE, they reported challenges related to unrealistic expectations from stakeholders, vagueness in ML problem specifications, and additional requirements such as regulatory constraints.

2.3 Concluding Remarks

In this chapter, we presented an overview of the difficulties and challenges regarding RE for ML-enabled systems and what are the main contributions within this area. We observed valuable theoretical contributions, but we recognize the need for industrial ones to bridge the gap between theory and practice. Unfortunately, empirical evidence on how RE is applied in practice in the context of ML-enabled systems is still weak and dominated by isolated studies.

3
Research Method

3.1
Introduction

In this chapter, we first define our goal and research questions, considering the importance of industrial insights for future research on RE for ML-enabled systems. Thereafter, we detail the employed research methodology, providing details on the survey design and on the methods for data collection and analysis.

3.2
Goal

Empirical evidence is fundamental to overcome the gap between theory and practice. In this sense, the goal of this dissertation is **to investigate the state of practice and problems of RE for ML-enabled systems**, sharing our findings with the community to help steer future research on the topic.

3.3
Research Questions

In order to achieve the presented goal, we set up two research questions presented in Table 3.1.

In ***RQ1***, we aim to reveal how practitioners are currently approaching RE for ML, identifying trends, prevalent methods, and the extent to which the industry aligns with established practices. In ***RQ2***, we aim to identify the ML-enabled systems challenges that are crucial in this investigation, once they inform the development of strategies to mitigate difficulties, helping to guide future research on the topic in a problem-driven manner.

In Table 3.2, we refine ***RQ1*** into more specific questions in order to better detail contemporary practices. These questions address who is in charge

Table 3.1: Dissertation research questions

ID	Description
RQ1	What are the contemporary practices on RE for ML-enabled systems?
RQ2	What are the main problems faced during the problem understanding and requirements ML life cycle stage?

Table 3.2: Details of research question 1

ID	Description
RQ1.1	Who is addressing the requirements of ML-enabled system projects?
RQ1.2	How are requirements typically elicited in ML-enabled system projects?
RQ1.3	How are requirements typically documented in the ML-enabled system projects?
RQ1.4	Which Non-Functional Requirements do typically play a major role in terms of criticality in the ML-enabled system projects?
RQ1.5	What activities are considered to be most difficult when defining requirements for ML-enabled systems?

of requirements, how requirements are elicited and documented, critical NFRs, and particularly challenging activities.

3.4 Survey Design

We designed our survey aiming to gather wide world reports on the current practices and problems of ML-enabled systems. In order to have a solid tool, we followed the best practices of survey research (WAGNER et al., 2020) and carefully conducted the following steps:

Step 1. Initial Survey Design. We conducted a literature review on RE for ML-enabled systems (VILLAMIZAR; ESCOVEDO; KALINOWSKI, 2021) and combined our findings with previous results on traditional RE problems (FERNÁNDEZ et al., 2017) and the RE status quo on traditional software (WAGNER et al., 2019) to provide the theoretical foundations for questions and answer options. Therefrom, the initial survey was drafted by Software Engineering and Machine Learning researchers from PUC-Rio (Brazil) with experience in R&D projects involving ML-enabled systems.

Step 2. Survey Design Review. The survey was reviewed and adjusted based on online discussions and annotated feedback from Software Engineering and Machine Learning researchers from BTH (Sweden). Thereafter, the survey was also reviewed by the other co-authors.

Step 3. Pilot Face Validity Evaluation. This evaluation involves a lightweight review by randomly chosen respondents. It was conducted with 18 Ph.D. students taking a Survey Research Methods course at UCLM (Spain) (taught by my advisor). They were asked to provide feedback on the clearness of the questions and to record their response time. This phase resulted in minor adjustments related to usability aspects and unclear wording. The answers were discarded before launching the survey.

Step 4. Pilot Content Validity Evaluation. This evaluation involves subject experts from the target population. Therefore, we selected five experienced data scientists developing ML-enabled systems, asked them to answer the survey, and gathered their feedback. The participants had no difficulties in answering the survey, and it took an average of 20 minutes. After this step, we considered the survey ready to be launched.

The survey was implemented using the Unipark Enterprise Feedback Suite. It started with a consent form describing the purpose of the study and stating that it is conducted anonymously. The remainder was divided into 15 demographic questions (D1 to D15) followed by three specific parts (P1 to P3) with 17 substantive questions (Q1 to Q17):

[**P1**] Q1 - Q7: Questions regarding ML life cycle stages and their problems

[**P2**] Q8 - Q12: Questions regarding requirements on ML-enabled systems

[**P3**] Q13 - Q17: Questions regarding deployment and monitoring of ML-enabled systems

In this dissertation, we focus on the demographics questions that help us characterize the participants; on **P1** questions regarding the perceived difficulty and relevance over ML life cycle stages, and the main problems faced on *Problem Understanding and Requirements* stage; and, lastly, the **P2** questions, focused on RE. The complete survey instrument is available in our open science repository (ALVES et al., 2023a). An excerpt of the substantive questions related to this dissertation is shown in Table 3.3.

Table 3.3: Research questions and related survey questions

RQ	Survey No.	Description	Type
-	...	...	...
RQ2	Q4	According to your personal experience, please outline the main problems or difficulties (up to three) faced during the Problem Understanding and Requirements ML life cycle stage.	Open
-	...	...	...
RQ1.1	Q8	Who is actively addressing the requirements of ML-enabled system projects in your organization?	Closed (MC)
RQ1.2	Q9	How were requirements typically elicited in the ML-enabled system projects you participated in?	Closed (MC)
RQ1.3	Q10	How were requirements typically documented in the ML-enabled system projects you participated in?	Closed (MC)
RQ1.4	Q11	Which Non-Functional Requirements (NFRs) typically play a major role in terms of criticality in the ML-enabled system projects you participated in?	Closed (MC)
RQ1.5	Q12	Based on your experience, what activities do you consider most difficult when defining requirements for ML-enabled systems?	Closed (MC)
-	...	...	...

3.5 Data Collection

Our target population concerns professionals involved in building ML-enabled systems, including different activities, such as management, design, and development. Therefore, it includes practitioners in positions such as project leaders, requirements engineers, data scientists, and developers. We used convenience sampling, sending the survey link to professionals active in our partner companies, and also distributed it openly on social media. We excluded participants that informed having no experience with ML-enabled system projects. Data collection was open from January 2022 to April 2022. In total, we received responses from 276 professionals, out of which 188 completed all four survey sections. The average time to complete the survey was 20 minutes. We conservatively considered only the 188 fully completed survey responses.

3.6 Data Analysis Procedures

For data analysis purposes, given that all questions were optional, the number of responses varies across the survey questions. Therefore, we explicitly indicate the number of responses when analyzing each question by informing N = *number of responses to that question*. Once we considered only 188 responses, the maximum N is 188.

3.6.1 Quantitative Data Procedures

Research questions *RQ1.1 - RQ1.5* were related to closed questions in our survey, so we decided to use inferential statistics to analyze them. Our population has an unknown theoretical distribution (*i.e.*, the distribution of ML-enabled system professionals is unknown). In such cases, resampling methods, like bootstrapping, have been reported to be more reliable and accurate than inference statistics from samples (LUNNEBORG, 2001; WAGNER et al., 2020). Hence, we use bootstrapping to calculate confidence intervals for our results, similar as done in (WAGNER et al., 2019).

In short, bootstrapping involves repeatedly taking samples with replacements and then calculating the statistics based on these samples. For each question, we take the sample of n responses for that question and bootstrap S resamples (with replacements) of the same size n. We assume n as the total valid answers of each question (EFRON; TIBSHIRANI, 1993), and we set 1000 for S, which is a value that is reported to allow meaningful statistics (LEI; SMITH, 2003). Figure 3.1 summarizes the adopted bootstrapping method.

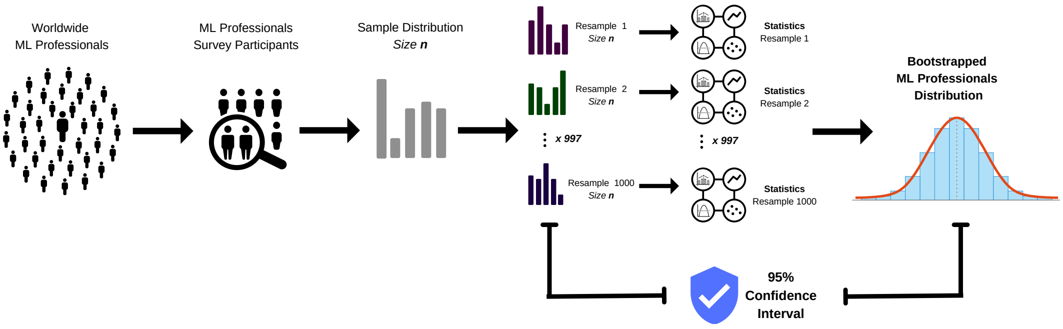


Figure 3.1: Bootstrapping technique

3.6.2 Qualitative Data Procedures

For research question *RQ2*, which seeks to identify the main problems faced by practitioners involved in engineering ML-enabled systems related to *Problem Understanding and Requirements* stage, the corresponding survey question was designed to be open text. We conducted a qualitative analysis using open and axial coding procedures from grounded theory (STOL; RALPH; FITZGERALD, 2016) to allow the problems to emerge from the open-text responses reflecting the experience of the practitioners. The qualitative coding procedures were conducted by one researcher and reviewed independently by four additional researchers from Brazil (1), Sweden (2) and Turkey (1).

3.7 Concluding Remarks

In this chapter, we presented our goal, research questions, and research method, including the survey design, data collection, and data analysis procedures for quantitative and qualitative data. The questionnaire, the collected data, and the quantitative and qualitative data analysis artifacts, including Python scripts for the bootstrapping statistics and graphs, and the peer-reviewed qualitative coding spreadsheets, are available in our open science repository (ALVES et al., 2023a). The study results will be presented in the next chapter.

4
Results

4.1
Introduction

In this chapter, we first summarize the study population in terms of demographic and professional profile. Thereafter, we present practitioners' perceptions about the difficulty and relevance of each ML life cycle stage. Finally, we delve into the results of our research.

4.2
Study Population

The survey's participants came from all parts of the world. Each continent has at least one respondent. In Figure 4.1, we have an overview of respondents nationality. Brazil, Turkey, Austria, Germany, Italy, and Sweden were the countries with more participants in the survey, which is expected once the survey was shared in a convenience sampling strategy where these countries with the most responses match with the researchers involved.

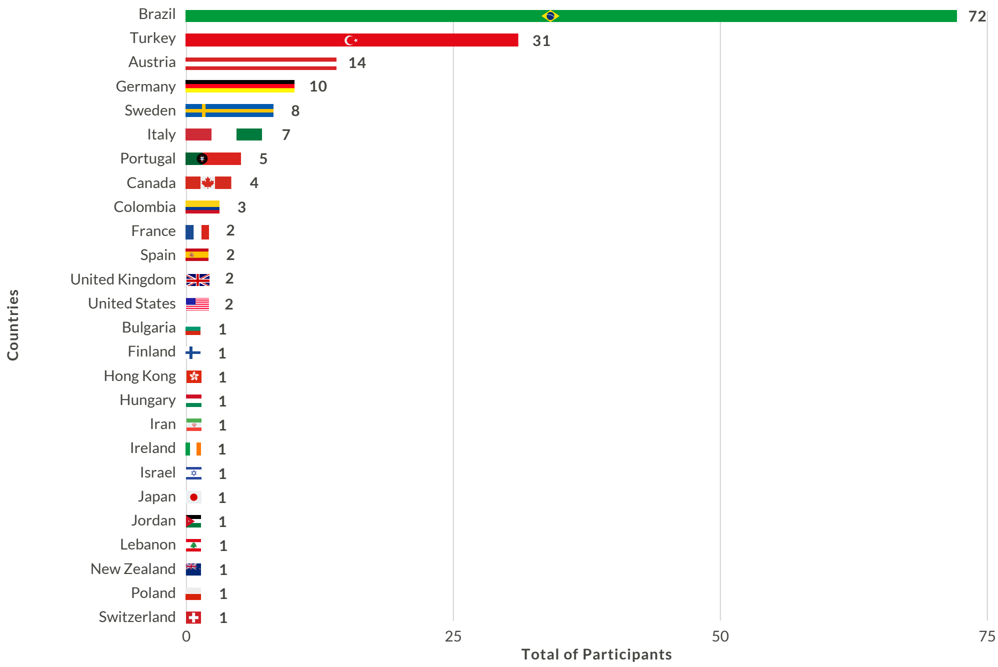


Figure 4.1: Participants nationality (N = 175)

Once the survey focused on an industry scenario, we extracted which company size the participants were currently working in, as presented in Figure 4.2. Most of the professionals work in big companies (more than 2000 employees) and the other significant part work in small to medium companies (from 50 to 250 employees). The fact of big companies leading this statistic suggests the trend of companies internalizing technologies and processes in order to prioritize business goals and domains.

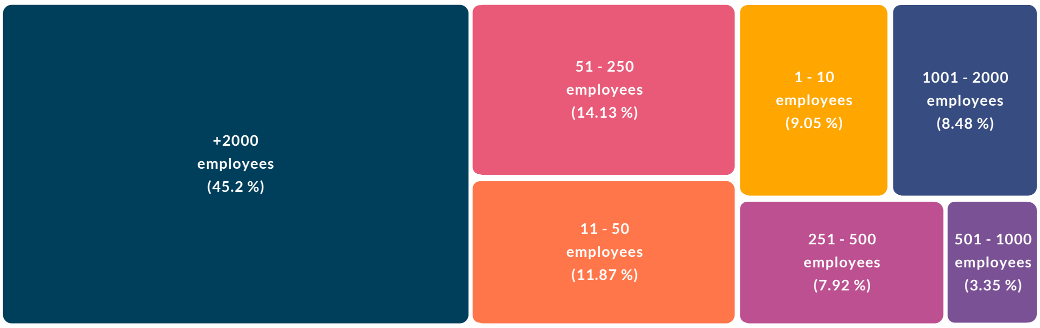


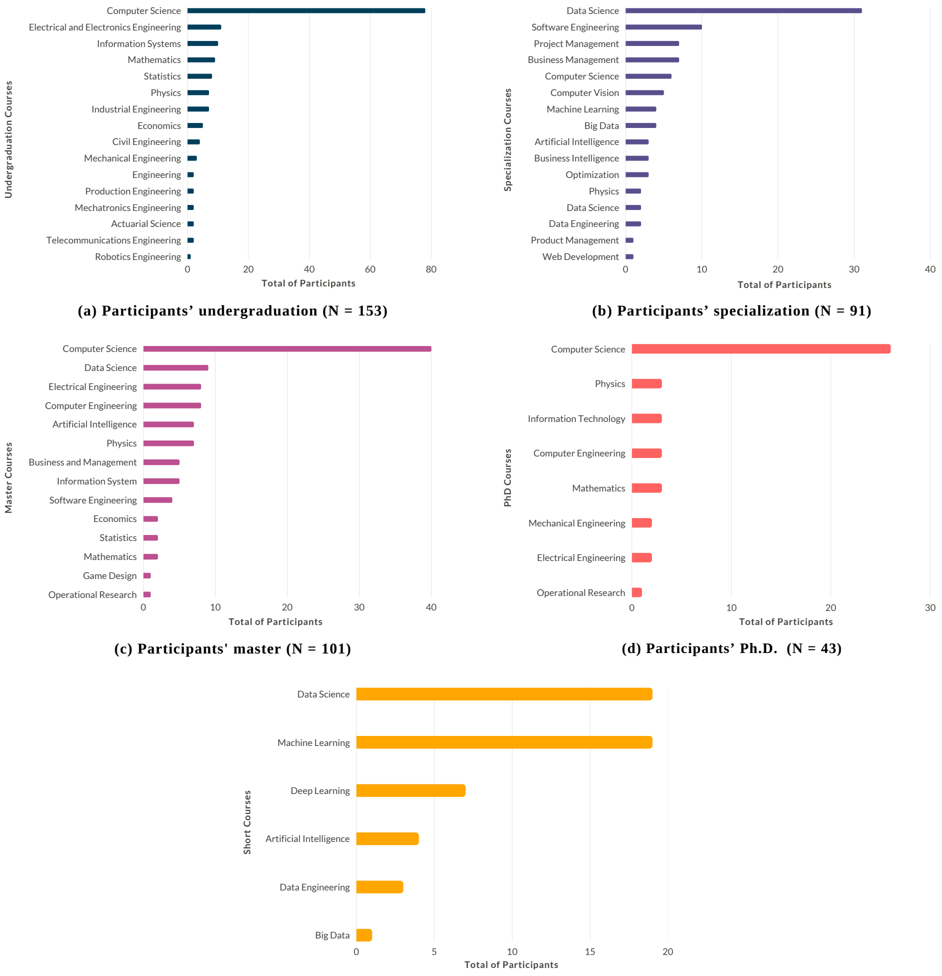
Figure 4.2: Current participants' company size (N = 177)

Regarding participants' background, we were able to detail their educational background in terms of Short Courses, Specializations, Undergraduate, Master, and Ph.D. degrees. We present an overview of it in Figure 4.3.

In terms of short-term courses (Figure 4.3 (e)), generally available on online educational platforms such as Coursera[1], Udemy[2], CodeAcademy[3], and Udacity[4], there is a significant focus on topics like Machine Learning, Data Science, Deep Learning, and Artificial Intelligence. Data Management and Data Governance also appear to be relevant, especially for the certifications offered by big tech companies on their products and services, such as Cloud Management and Data Engineering certifications on Google Cloud Platform and Azure, from Google and Microsoft, respectively.

Unlike short courses, participants were also asked if they had taken specialization courses, and the results are presented in Figure 4.3 (b). In this sense, professionals focused on specializations related to Computer Science topics, such as Data Science, Software Engineering, Project Management, Big Data, Business Intelligence, and Web Development. Other topics not directly related to technology were also present, such as courses focused on Physics and Optimization. However, these topics, in general, use technology to achieve and improve results. Other specializations that were less representative included Econometrics, Marketing, Psychology, and Autonomous Driving.

[1] https://www.coursera.org/
[2] https://www.udemy.com/
[3] https://www.codecademy.com/
[4] https://www.udacity.com/



(a) Participants' undergraduation (N = 153)

(b) Participants' specialization (N = 91)

(c) Participants' master (N = 101)

(d) Participants' Ph.D. (N = 43)

(e) Participants' short courses (N = 53)

Figure 4.3: Participants educational background

Regarding formal education, most participants have an undergraduate degree, as shown in Figure 4.3 (a). Computer Science is the most chosen field, although many professionals have a background in Electrical, Mechanical, and Civil Engineering. In addition to engineers, there were also many mathematicians and statisticians. Other degrees with fewer responses were informed, such as Biology, Logistics, Aerospace Engineering, Nuclear Engineering, and Robotics, showcasing the diversity of professionals working with Machine Learning in the industry.

When we look at graduate courses, such as Master's and Ph.D. degrees, the profile is predominantly related to Technology, Mathematics, Engineering, and Economics. In Figure 4.3 (c), we show that out of the top 6 choices for Master's degrees, 5 were related to computer science: Computer Science, Data Science, Artificial Intelligence, Computer Engineering, and Information Systems. In addition to these focuses, participants also emphasized socio-economic and mathematical topics, such as Statistics, Operational Research, Business Management, and Economics. Professionals with Ph.D. degrees were less frequent, but in Figure 4.3 (d), we show that for those who had a Ph.D., their research focus was mainly in the areas of Computer Science, and the remainder was divided among Mathematics, Physics, and Computer, Electrical, and Mechanical Engineering.

It can be said that the majority of professionals who participated in the survey have an analytical profile with backgrounds in technology, engineering, and mathematics. This predominant profile is reflected in the roles that these professionals have in their companies. In Figure 4.4, it is possible to observe that most participants work as Data Scientists. Other significant roles are Project Leader, Developer, Solution Architect, and Business Analyst. Lastly, we have a few practitioners assuming the Requirements Engineer and Test Manager roles. Despite being representative in the 'Others' field, isolated positions were mentioned, such as CEO, CTO, Operational Research Analyst, and Machine Learning Engineer.

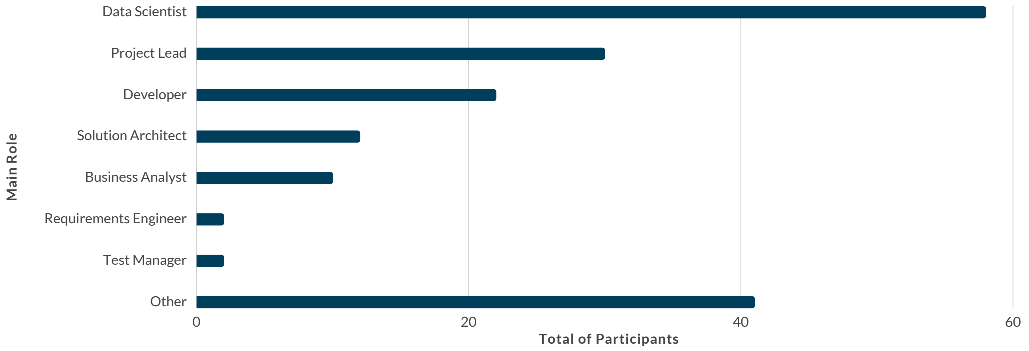
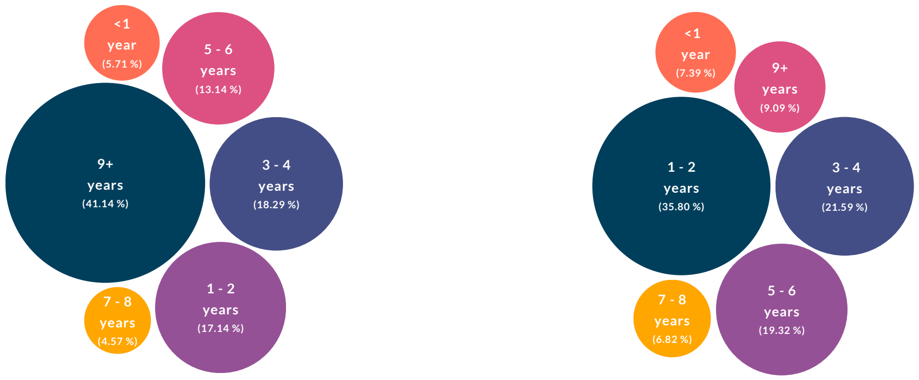


Figure 4.4: Participants' main role (N = 177)

Our respondents are senior professionals in terms of software development, as shown in Figure 4.5 (a). However, these professionals have relatively less experience with Machine Learning projects, as depicted in Figure 4.5 (b). Despite Machine Learning being a relatively new field, the median experience of the professionals who answered the survey was three years.



(a) Participants' Software Experience (N = 175)

(b) Participants' ML Experience (N = 176)

Figure 4.5: Participants' experience on software and ML-enabled systems

4.3 Problem Understanding and Requirements ML Life Cycle Stage

In the survey, similar to what was done by Kalinowski *et al.* (KALINOWSKI et al., 2023), based on the nine ML life cycle stages presented by Amershi *et al.* (AMERSHI et al., 2019) and the Cross Industry Standard Process for Data Mining (CRISP-DM) industry-independent process model phases (SCHRÖER; KRUSE; GÓMEZ, 2021), we abstracted seven generic life cycle stages as shown in Figure 4.6 and asked about their perceived relevance and difficulty.

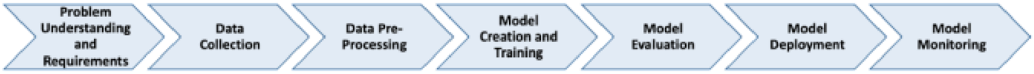
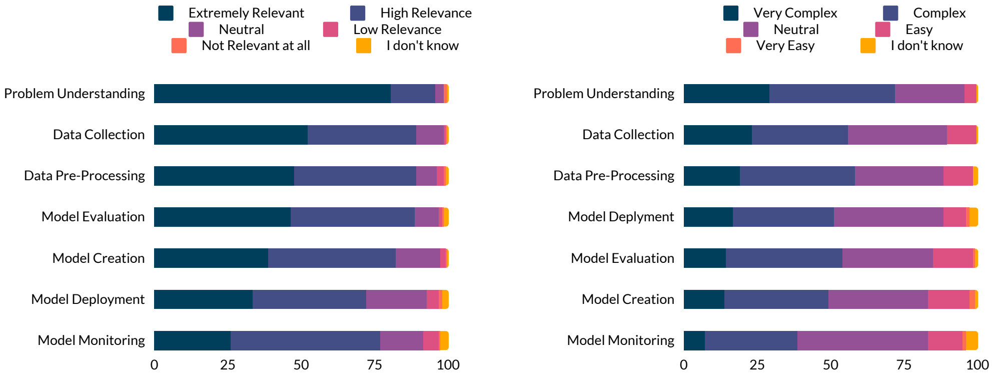


Figure 4.6: Seven stages of ML life cycle

The answers are presented in Figure 4.7 and reveal that ML practitioners are extremely worried about requirements. The *Problem Understanding and Requirements* stage is clearly perceived as the most relevant and most complex life cycle stage.



(a) Perceived Relevance (N = 171)

(b) Perceived Difficulty(N = 169)

Figure 4.7: Perceived relevance and complexity of each ML life cycle stage

4.4 RQ1: Contemporary RE practices for ML-enabled systems

4.4.1 RQ1.1: Who is addressing the requirements of ML-enabled system projects?

The proportion of roles reported to address the requirements of ML-enabled system projects within the bootstrapped samples is shown in Figure 4.8 together with the 95 % confidence interval. The N in each figure caption is the number of participants that answered this question, as previously mentioned. We report the proportion P of the participants that checked the corresponding answer and its 95% confidence interval in square brackets.

It is possible to observe that the project lead and data scientists were most associated with requirements in ML-enabled systems with **P = 56.664 [56.435, 56.893]** and **P = 54.618 [54.376, 54.859]**, while Requirements Engineers and Business Analysts had a much lower proportion **P = 11.022 [10.875, 11.17]** and **P = 29.558 [29.347, 29.768]**, respectively. Developers and Solution Architects were also significant with **P = 21.832 [21.625, 22.04]** and **P = 14.074 [13.911, 14.236]**. Testers were the least option checked with **P = 1.191 [1.138, 1.245]**. Several isolated options were mentioned in the “Others" field (*e.g.*, Product Owner, Machine Learning Engineer, and Tech Lead), altogether summing up 11% and not significantly influencing the overall distribution (**P = 11.021 [10.865, 11.177]**).

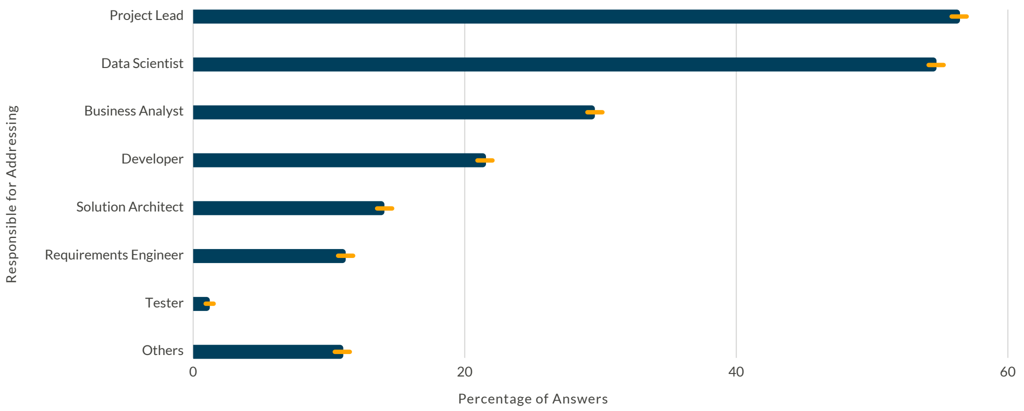


Figure 4.8: Roles addressing requirements of ML-enabled systems (N = 170)

4.4.2 RQ1.2: How are requirements typically elicited in ML-enabled system projects?

As presented in Figure 4.9, respondents reported Interviews as the most commonly used technique (**P = 55.615** [**55.382, 55.849**]), followed (or complemented) by Prototyping (**P = 43.7** [**43.433, 43.967**]), Scenarios (**P = 43.238** [**42.997, 43.479**]), Workshops (**P = 42.663** [**42.441, 42.885**]), and Observation **P = 36.826** [**36.584, 37.069**]. A few options were mentioned in the “Others" field (*e.g.*, simulating system’s behavior, and simple meetings with stakeholders), with a small proportion **P = 6.45** [**6.337, 6.563**].

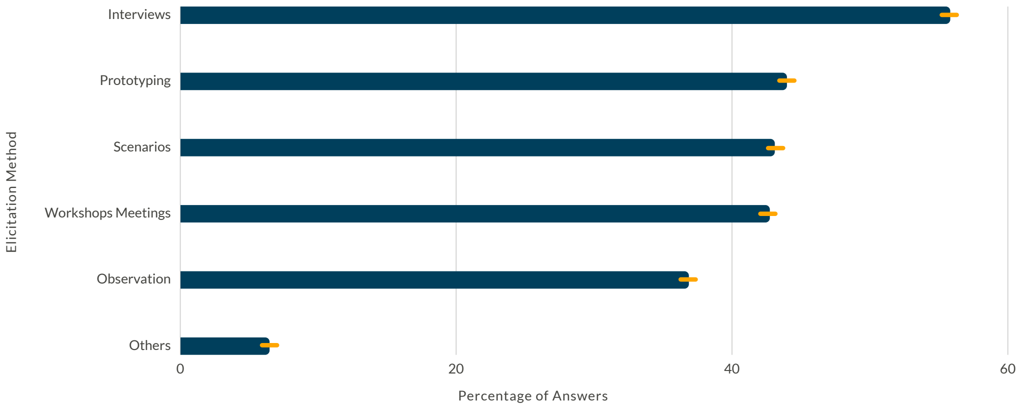


Figure 4.9: Requirements elicitation techniques of ML-enabled systems (N = 171)

4.4.3 RQ1.3: How are requirements typically documented in the ML-enabled system projects?

Figure 4.10 shows Notebooks as the most frequently used documentation format with **P = 37.448** [**37.213, 37.683**], followed by User Stories (**P**

= **36.169** [**35.929, 36.409**]), Requirements Lists (**P = 29.773** [**29.539, 30.007**]), Prototypes (**P = 24.26** [**24.031, 24.489**]), Use Case Models (**P = 21.734** [**21.546, 21.922**]), and Data Models (**P = 19.99** [**19.791, 20.189**]). Surprisingly, almost 17% mentioned that requirements are not documented at all **P = 16.845** [**16.672, 17.018**]. A similar proportion uses Vision Documents (**P = 16.96** [**16.791, 17.13**]) and Goal Models (**P = 16.919** [**16.742, 17.097**]). The least used options were ML Canvas (**P = 8.757** [**8.62, 8.893**]) and Behavior-Driven Development (BDD) Scenarios (**P = 2.276** [**2.202, 2.35**]). Several options were mentioned in "Others" (*e.g.*, Wiki tools, Google Docs, Jira) altogether summing up 8.8% (**P = 8.784** [**8.643, 8.925**]).

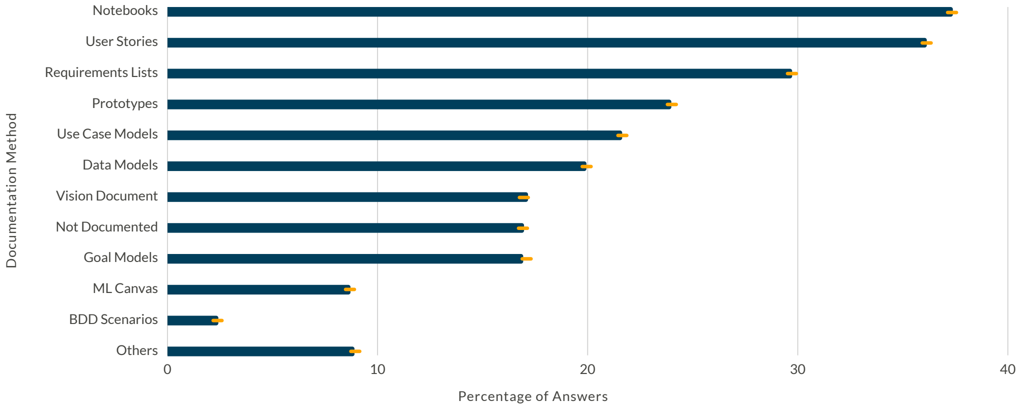


Figure 4.10: Requirements documentation of ML-enabled systems (N = 171)

4.4.4 RQ1.4: Which Non-Functional Requirements do typically play a major role in terms of criticality in the ML-enabled system projects?

Regarding NFRs, practitioners show a significant concern with some ML-related NFRs, such as Data Quality (**P = 69.825** [**69.603, 70.048**]), Model Reliability (**P = 42.788** [**42.584, 42.991**]), and Model Explainability (**P = 37.97** [**37.741, 38.2**]). Some NFR regarding the whole system were also considered important, such as System Performance (**P = 40.667** [**40.435, 40.9**]), and Usability (**P = 29.501** [**29.272, 29.731**]). A significant amount of participants informed that NFRs were not at all considered within their ML-enabled system projects (**P = 10.667** [**10.518, 10.817**]). Lastly, in the "Others" field (**P = 1.749** [**1.685, 1.814**]), a few participants also mentioned that they did not reflect upon NFRs as presented in Figure 4.11.

The remainder system-NFRs consist in System Reliability (**P = 20.841** [**20.637, 21.046**]), System Maintainability (**P = 20.64** [**20.437, 20.844**]), System Security (**P = 18.859** [**18.684, 19.034**]), System Compatibility (**P = 15.951** [**15.778, 16.124**]), System Privacy (**P = 13.058** [**12.91, 13.206**]), System Safety (**P = 12.409 [12.255, 12.563]**), and System Portability (**P = 7.586 [7.456, 7.716]**). In the other hand, the remainder ML-related NFRs were Model Accountability (**P = 15.559 [15.39, 15.727]**), Model Ethics Fairness (**P = 13.007 [12.857, 13.158]**), Model Interactiveness (**P = 8.867 [8.723, 9.012]**), and Model Transparency (**P = 19.021 [18.829, 19.214]**).

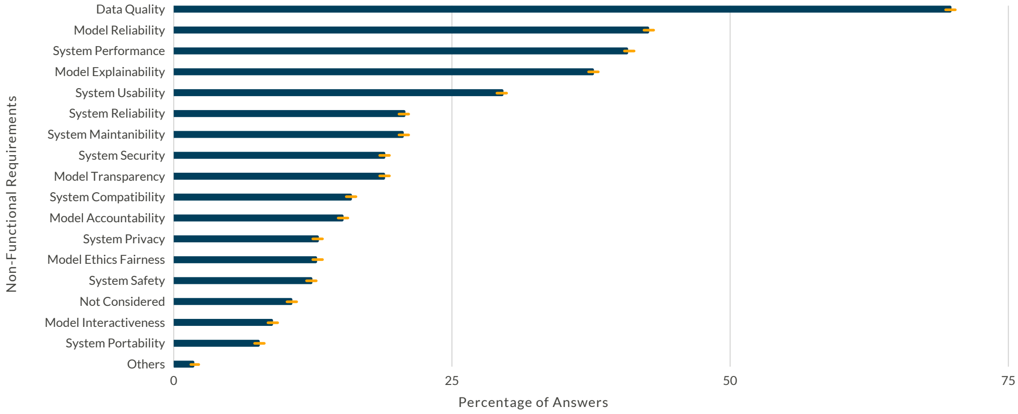


Figure 4.11: Critical Non-Functional Requirements of ML-enabled systems (N = 169)

4.4.5 RQ1.5: Which activities are considered most difficult when defining requirements for ML-enabled systems?

We provided answer options based on the literature on requirements (WAGNER et al., 2019) and requirements for machine learning (VILLAMIZAR; ESCOVEDO; KALINOWSKI, 2021), leaving the “Other" option to allow new activities to be added. As shown in Figure 4.12, respondents considered that managing customer expectations is the most difficult task (**P = 66.684 [66.464, 66.904]**), followed by aligning requirements with data (**P = 57.446 [57.22, 57.673]**), resolving conflicts (**P = 38.535 [38.303, 38.766]**), managing changing requirements (**P = 35.649 [35.402, 35.895]**), selecting metrics (**P = 33.992 [33.752, 34.231]**), elicitation and analysis (**P = 29.292 [29.062, 29.523]**), documentation (**P = 15.805 [15.637, 15.973]**), new quality attributes (**P = 14.167 [14.003, 14.331]**), and verification task (**P = 12.886 [12.733, 13.039]**). In the “Others" field (**P = 1.761 [1.7, 1.822]**), few participants mentioned their difficulty with customer expectations, but others informed a not listed difficulty, the requirements traceability.

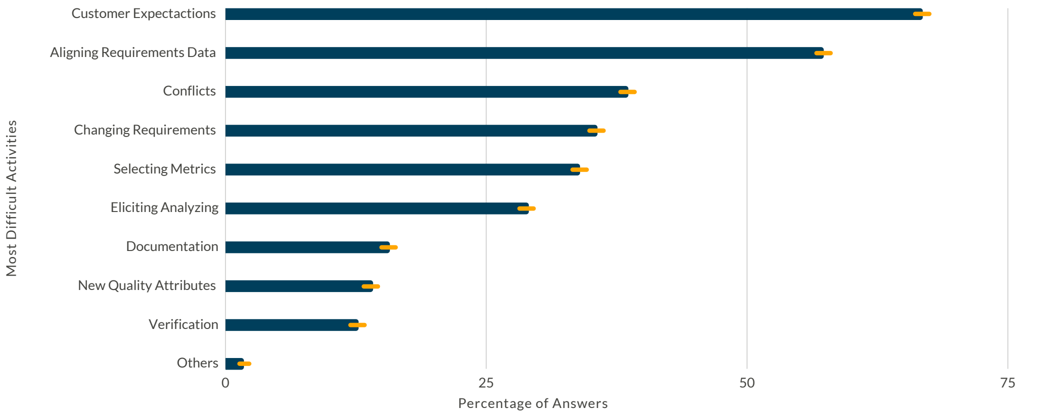


Figure 4.12: Most difficult RE activities in ML-enabled systems (N = 171)

4.5 RQ2: Main problems faced during the problem understanding and requirements ML life cycle stage

Regarding the main problems faced by the participants during the *Problem Understanding and Requirements* stage, they emerged from open coding applied to free text answers. Participants could inform up to three problems related to each ML life cycle stage. In total, 262 open-text answers were provided for problems related to problem understanding and requirements.

We incorporated axial coding procedures to provide an easily understandable overview, relating the emerging codes to categories. We started with the sub-categories *Input*, *Method*, *Organization*, *People*, and *Tools*, as suggested for problems in previous work on defect causal analysis (KALINOWSKI; CARD; TRAVASSOS, 2012). Based on the data, we merged the *Input* and *People* categories, as it was difficult to separate between the two, given the concise answers provided by the participants. We also renamed the *Tools* category into *Infrastructure* and identified the need to add a new category related to *Data*. It is noteworthy that these categories were identified considering the overall coding for the seven ML life cycle stages, while in this paper, we focus on the *Problem understanding and Requirements* stage.

Figure 4.13 presents an overview of the frequencies of the resulting codes using a probabilistic cause-effect diagram, which was introduced for causal analysis purposes in previous work (KALINOWSKI et al., 2010; KALINOWSKI; MENDES; TRAVASSOS, 2011). While this representation provides a comprehensive overview, the percentages are just frequencies of occurrence of the codes (*i.e.*, the sum of all code frequencies is 100%). Also, the highest frequencies within each category are organized closer to the middle.

It is possible to observe that most of the reported problems are related to the *Input* category, followed by *Method* and *Organization*. Within the

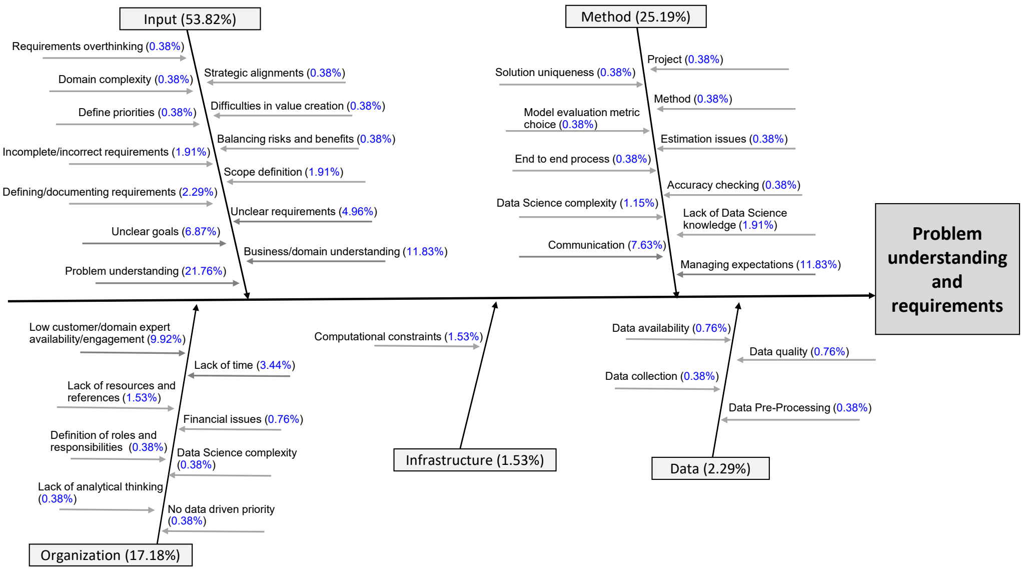

Input (53.82%)
Requirements overthinking (0.38%)
Domain complexity (0.38%)
Define priorities (0.38%)
Incomplete/incorrect requirements (1.91%)
Defining/documenting requirements (2.29%)
Unclear goals (6.87%)
Problem understanding (21.76%)
Strategic alignments (0.38%)
Difficulties in value creation (0.38%)
Balancing risks and benefits (0.38%)
Scope definition (1.91%)
Unclear requirements (4.96%)
Business/domain understanding (11.83%)
Method (25.19%)
Solution uniqueness (0.38%)
Model evaluation metric choice (0.38%)
End to end process (0.38%)
Data Science complexity (1.15%)
Communication (7.63%)
Project (0.38%)
Method (0.38%)
Estimation issues (0.38%)
Accuracy checking (0.38%)
Lack of Data Science knowledge (1.91%)
Managing expectations (11.83%)
Problem understanding and requirements
Low customer/domain expert availability/engagement (9.92%)
Lack of resources and references (1.53%)
Definition of roles and responsibilities (0.38%)
Lack of analytical thinking (0.38%)
Lack of time (3.44%)
Financial issues (0.76%)
Data Science complexity (0.38%)
No data driven priority (0.38%)
Organization (17.18%)
Computational constraints (1.53%)
Infrastructure (1.53%)
Data availability (0.76%)
Data collection (0.38%)
Data quality (0.76%)
Data Pre-Processing (0.38%)
Data (2.29%)

Figure 4.13: Main problems faced during Problem Understanding and Requirements phase

Input category, the main problems concern difficulties in understanding the problem and the business domain, and unclear goals and requirements. In the *Method* category, the prevailing reported problems concern difficulties in managing expectations and establishing effective communication. Finally, in the *Organization* category, the lack of customer or domain expert availability and engagement, and the lack of time dedicated to requirements-related activities were mentioned. While we focus our summary on the most frequently mentioned problems, it is noteworthy that the less frequent ones may still be relevant in practice. For instance, computational constraints or a lack of data quality (or availability) can directly affect ML-related possibilities and requirements.

4.6 Concluding Remarks

In this chapter, we presented our study results, describing, firstly, the practitioners' demographics and perception about the *Problem Understanding and Requirements* stage. Thereafter, we described the analyses regarding each of our research questions, including both quantitative and qualitative data analysis procedures.

5
Discussion

5.1
Introduction

In this chapter, we discuss the threats to the validity of our survey design and analysis. Furthermore, we also discuss our main findings, their implications for researchers and practitioners, and their relation to previous literature. Finally, we investigate a potential country-related bias.

5.2
Threats to Validity

We identified some threats while planning, conducting, and analyzing the survey results. Hereafter, we list these potential threats, organized by the survey validity types presented in (LINAKER et al., 2015).

5.2.1
Face and Content Validity

Face and content validity threats include bad instrumentation and inadequate explanation of constructs. To mitigate these threats, we involved several researchers in reviewing and evaluating the questionnaire with respect to the format and formulation of the questions, piloting it with 18 Ph.D. students for face validity and with five experienced data scientists for content validity.

5.2.2
Criterion Validity

Threats to criterion validity include not surveying the target population. We clarified the target population in the consent form (before starting the survey). We also considered only complete answers (*i.e.*, answers of participants that answered all four survey sections) and excluded participants that had no experience with ML-enabled system projects.

5.2.3
Construct Validity

We ground our survey's questions and answer options on theoretical background from previous studies on RE (FERNÁNDEZ et al., 2017; WAGNER et al., 2019) and a literature review on RE for ML (VILLAMIZAR; ESCOVEDO;

KALINOWSKI, 2021). A threat to construct validity is inadequate measurement procedures and unreliable results. To mitigate this threat, we follow recommended data collection and analysis procedures (WAGNER et al., 2020).

5.2.4 Reliability

One aspect of reliability is statistical generalizability. We could not construct a random sample systematically covering different types of professionals involved in developing ML-enabled systems, and there is yet no generalized knowledge about what such a population looks like. Furthermore, as a consequence of convenience sampling, the majority of answers came from South America and Europe. Nevertheless, the experience and background profiles of the subjects are comparable to the profiles of ML teams, as shown in Microsoft's study (KIM et al., 2017). To deal with the random sampling limitation, we used bootstrapping and employed confidence intervals, conservatively avoiding null hypothesis testing. Another reliability aspect concerns inter-observer reliability, which we improved by including independent peer review in all of our qualitative analysis procedures and making all the data and analyses openly available online in our open science repository (ALVES et al., 2023a).

5.3 Main findings and implications

The survey findings reveal an important aspect within the ML-enabled systems context, which is the distribution of roles in RE activities. In traditional software projects, the role of Requirements Engineer is not prominent once its duties mix with some other positions such as developers, architects, project managers, or consultants (HERRMANN, 2013). Moreover, in some North American and European companies, it is commonplace to see the job title *Business Analyst* instead of *Requirements Engineer* to handle requirements (WANG et al., 2018).

However, in ML-enabled systems, a notable transition is observed, with project leaders and data scientists taking the lead in RE efforts. Contrary to expectations, the roles of business analysts, developers, solution architects, and requirements engineers are less associated when addressing requirements. It is worth mentioning that despite having a significant proportion of participants identifying themselves as *Business Analysts*, we still have a predominant association between addressing requirements with Project Leaders and Data Scientists.

The literature suggests that RE can help to address problems related to engineering ML-enabled systems, but the software engineering practices are not yet well established within this domain in practice. This could be pointing to the low participation of traditional roles on RE for such systems. Nevertheless, the involvement of project leaders and data scientists as key RE contributors could reflect the nature of ML projects, where domain expertise and data-driven insights are pivotal. This shift in responsibilities raises questions about the evolving dynamics of cross-functional collaborations within ML endeavors and prompts further exploration into how such roles influence the shaping of ML-enabled systems.

The survey also revealed that practitioners typically use traditional requirements elicitation techniques (interviews, prototyping, scenarios, workshops, and observation), even with a text-free option available for practitioners to inform some missing or new elicitation methods. Comparing the results to the elicitation techniques reported for traditional RE (WAGNER et al., 2019), a noticeable difference is that requirements workshops are slightly less commonly used in ML-enabled system contexts. This could be related either to the absence of traditional RE positions in the elicitation phase (*e.g.*, requirements engineer or business analyst), who would be typically familiar with conducting such workshops, or to the lack of specific adaptations on such workshop formats for ML-enabled systems.

With respect to requirements documentation, Notebooks, which are interactive programming environments that can be used to process data and create ML models, appear as the most used tool for documenting requirements. Again, this could be a symptom of the absence of a requirements engineer and the lack of awareness of RE specification practices and tools. Furthermore, a proportion of almost 17% mentioned that requirements were not documented at all. Given that in conventional contexts, problems related to requirements are common causes of overall software project failure (FERNÁNDEZ et al., 2017), this apparent lack of RE-related maturity regarding documentation may also be causing pain in ML-enabled system contexts.

Traditional artifacts, such as user stories, requirements lists, prototypes, and use case models, are also used in the ML-enabled systems context, but significantly less than in the conventional software context (WAGNER et al., 2019). Goal-oriented models are a common tool for documenting Self-Adaptative-Systems (SAS) projects (BELANI; VUKOVIĆ; CAR, 2019). However, within the ML-enabled system context, it is as important as not documenting requirements. Even specific approaches, such as the ML Canvas, do not relevantly represent a current practice for documenting the requirements of ML-enabled systems.

Regarding NFRs, practitioners express considerable concerns with specific ML-related NFRs, such as data quality, model reliability, and model explainability, which is something previously discussed in the literature (VOGELSANG; BORG, 2019; HORKOFF, 2019; HABIBULLAH; GAY; HORKOFF, 2023). In spite of these expected model concerns, we found practitioners recognizing the significance of overall system-related NFRs. In other words, the system which includes the ML model may be invariant to problems on this specific part, it needs to be reliable, useful, secure, and maintainable. Besides systems and model concerns, more than 10% of practitioners do not even consider NFRs in their ML-enabled system projects. Again, given the potential negative impacts of missing NFRs on software-related projects (FERNÁNDEZ et al., 2017), this can be seen as another indicator of the lack of overall awareness of the importance of RE in the industrial ML-enabled systems engineering context.

The survey also revealed the most difficult activities perceived by practitioners in defining requirements for ML-enabled systems. The difficulties reported by practitioners mix with previous literature, but now comes in a wider industrial scope. Managing customer expectations (ISHIKAWA; YOSHIOKA, 2019), aligning requirements with data (NAHAR et al., 2023; VILLAMIZAR; ESCOVEDO; KALINOWSKI, 2021), changing requirements (KHALAJZADEH et al., 2018), and selecting proper metrics (VOGELSANG; BORG, 2019) were previously reported as difficulties, which highlight the importance of effective communication, a deep understanding of customer needs, and domain and technical expertise to bridge the gap between aspirations and technological feasibility.

Finally, we contributed to the RE-related problems faced by practitioners in ML-enabled system projects. The main issues relate to difficulties in problem and business understanding, managing expectations, and low customer/-domain expert availability/engagement. These issues clearly have comparable counterparts in the conventional RE problems (FERNÁNDEZ et al., 2017). In Table 5.1, we show this strong relationship between problems in ML-enabled systems and traditional contexts. As comparable problems may have comparable solutions, adopting established RE practices (or adaptations of such practices) may help improve ML-enabled system engineering.

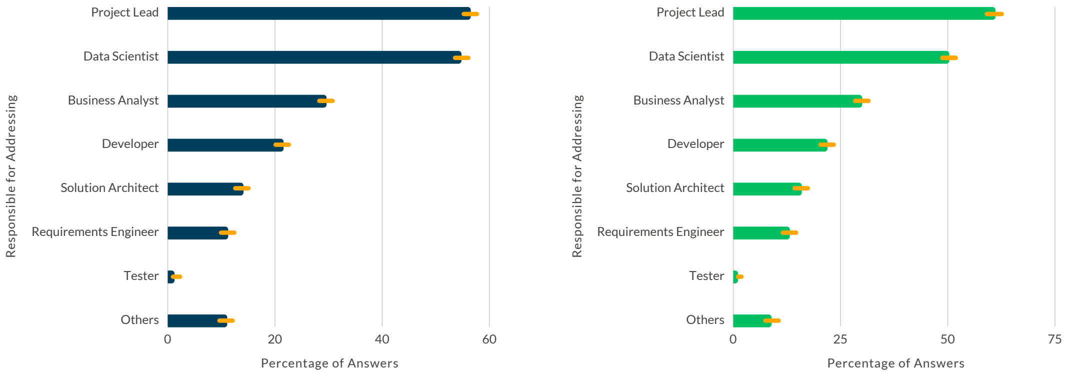
Table 5.1: Comparison between problems on ML-enabled and traditional systems

Traditional RE Problem	ML RE Problem
Incomplete and/or hidden requirements	[Input] Incomplete/incorrect requirements
Communication flaws between project team and customer	[Method] Communication
Moving targets (changing goals, business processes, and/or requirements)	[Input] Unclear goals
Underspecified requirements that are too abstract	[Input] Unclear requirements
Timeboxing/Not enough time in general	[Organization] Lack of time
Communication flaws within the project team	[Organization] Definition of roles and responsibilities
Stakeholders with difficulties in separating requirements from known solution designs	[Organization] Lack of analytical thinking
Insufficient support by customer	[Organization] Low client/domain expert availability/engagement
Inconsistent requirements	[Input] Requirements overthinking
Weak access to customer needs and/or business information	[Organization] Lack of resources and references

5.4 Brazil Bias Investigation

Our convenience sampling strategy led us to a significant amount of answers from Brazil. As informed in our Threats to Validity, our results regarding practitioners' backgrounds are comparable with previous literature, and we strongly believe that the ML-enabled system context is globalized. Thus, participants' nationality should not have a big impact on wider analyses. Furthermore, the purpose of this dissertation is not to check Brazil or any other country's influence on RE for ML-enabled systems. To analyze if the most frequent country was biasing our results, we also analyzed our data by applying blocking to remove the country with more participants (Brazil). Hence, we removed Brazilian answers from questions that involved quantitative analyses and verified how similar these results were to the previous ones.

Firstly, when removing Brazilian answers, we observed that the distribution of roles addressing requirements didn't change. *Projects Leaders* and *Data Scientists* still lead this task, with a small leading difference between them in the version without Brazil, as presented in Figure 5.1.

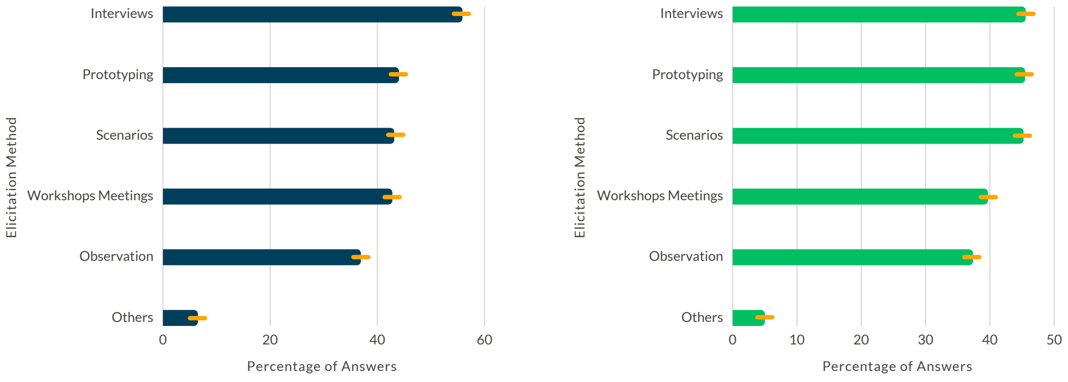


(a) All Valid Answers (N = 170)

(b) Valid Answers without Brazil (N = 100)

Figure 5.1: RQ1.1 with and without Brazil

In terms of elicitation methods, when discarding Brazilian answers, we kept the same distribution of methods, with an observable difference in the usage of *Interviews*. Without Brazil, *Prototyping*, and *Scenarios* are almost equally used, and with Brazil, these methods were slightly more distant, as presented in Figure 5.2. This may indicate that in Brazil *Interviews* seem to be more commonly used as the main elicitation method.

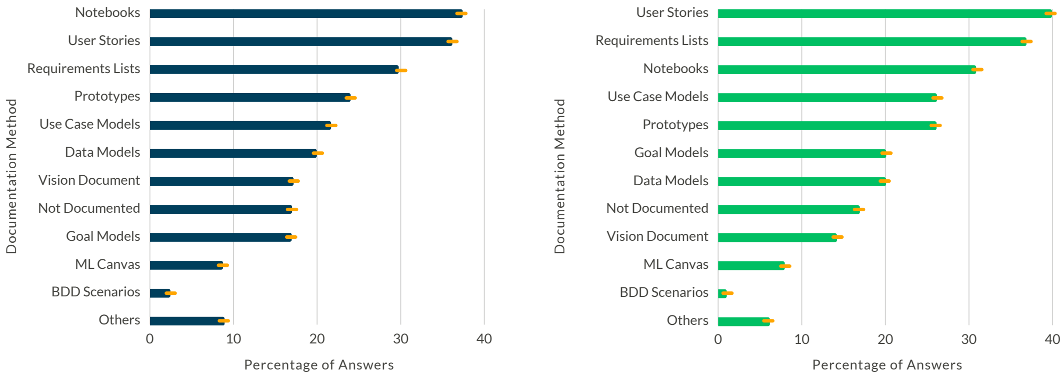


(a) All Valid Answers (N = 171)

(b) Valid Answers without Brazil (N = 99)

Figure 5.2: RQ1.2 with and without Brazil

Regarding RE documentation, when not considering Brazilian participants, we had some significant direct differences but preserved the underlying conclusions. *Notebooks* left the leading position but was still close to the new leading documentation artifact, *User Stories*. Other documentation tools were previously close, and without Brazil, minor switches happened, such as *Use Case Models* and *Prototypes*; and *Goal Models* and *Data Models*. *ML Canvas* and *BDD Scenarios* are still the least used artifacts.

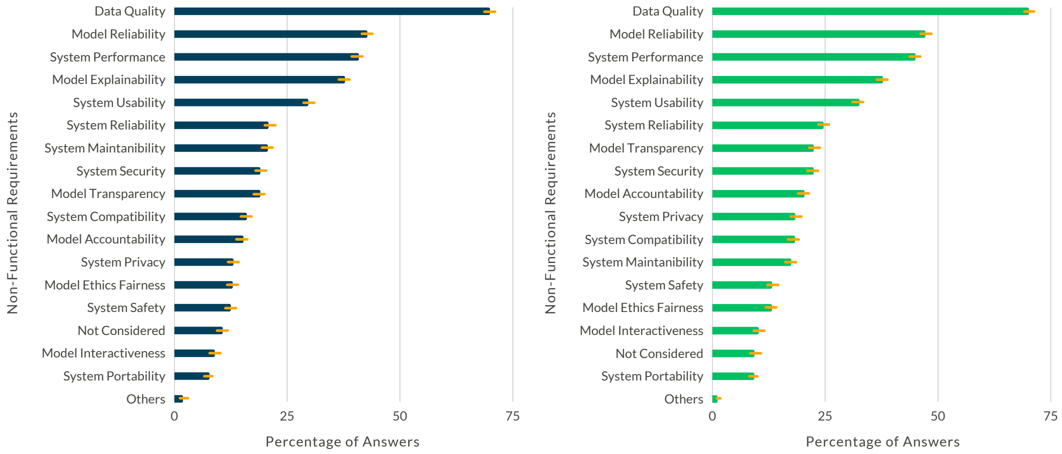


(a) All Valid Answers (N = 171)

(b) Valid Answers without Brazil (N = 100)

Figure 5.3: RQ1.3 with and without Brazil

With respect to NFRs, the main conclusion was still the same with or without Brazil, as presented in Figure 5.4. Few NFRs changed positions but were definitely close to what was previously found, such as Model Transparency, Model Accountability, System Maintainability, and System Compatibility. Despite these changes, the NFRs proportion between them is in a range of very close values, thus, these changes have little impact.

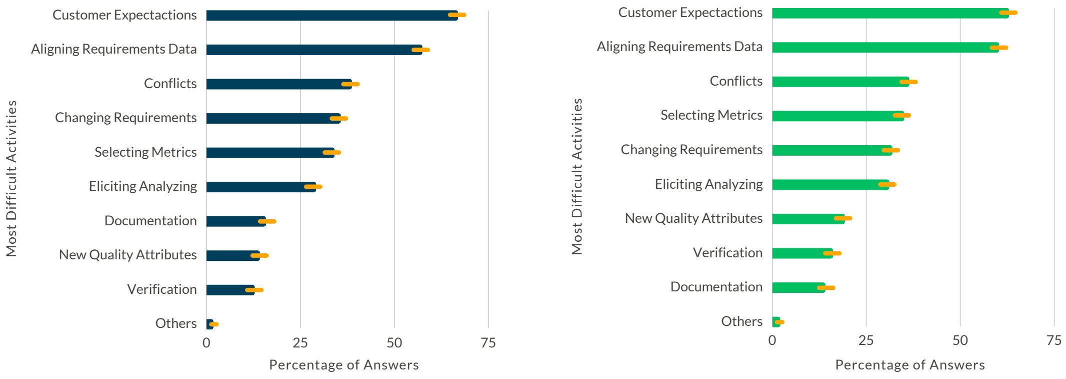


(a) All Valid Answers (N = 171)

(b) Valid Answers without Brazil (N = 98)

Figure 5.4: RQ1.4 with and without Brazil

Lastly, regarding the most difficult activities when dealing with RE, we also had minor changes. Without Brazil, the top three difficulties were preserved, however, we had direct switches between *Selecting Metrics* and *Changing Requirements*; and between *New Quality Attributes* and *Documentation*. Their previous proximity allows us to conclude that the latent difficulties were mostly similar.



(a) All Valid Answers (N = 171)

(b) Valid Answers without Brazil (N = 100)

Figure 5.5: RQ1.5 with and without Brazil

5.5 Concluding Remarks

In this chapter, we presented our main contributions, which shed light on some new aspects and reaffirmed others from the previous literature in a wider scope. We also discussed the main threats to the validity of our study and showed that Brazil's prevalence of participants' nationality didn't affect the answers' distribution, increasing the confidence in the generalizability of our findings.

6
Conclusion

6.1
Contributions

Literature suggests that RE can help to tackle challenges in ML-enabled system engineering (VILLAMIZAR; ESCOVEDO; KALINOWSKI, 2021). Recent literature studies (*e.g.*, (VILLAMIZAR; ESCOVEDO; KALINOWSKI, 2021; AHMAD et al., 2021; NAHAR et al., 2023)) and industrial studies (*e.g.*, (VOGELSANG; BORG, 2019; SCHARINGER et al., 2022)) on RE for ML-enabled systems have been important to help to understand the literature focus and industry needs. However, the studies on industrial practices and problems are still isolated and not yet representative.

We complement these studies, aiming at strengthening empirical evidence on current RE practices and problems when engineering ML-enabled systems, with an industrial survey that collected responses from 188 practitioners involved in engineering such systems. We applied bootstrapping with confidence intervals for quantitative statistical analysis, and open and axial coding for qualitative analysis of RE problems.

Our analyses confirmed some of the findings of previous ML-enabled system studies, such as the relevance NFRs related to data quality, model reliability, and explainability (VOGELSANG; BORG, 2019; HORKOFF, 2019; HABIBULLAH; GAY; HORKOFF, 2023), and challenges related to customer expectation management and vagueness of requirements specifications (VILLAMIZAR; ESCOVEDO; KALINOWSKI, 2021; NAHAR et al., 2023). However, we also shed light on some new and intriguing aspects. For instance, the survey revealed that project leaders and data scientists are taking the reins in RE activities for the ML-enabled systems and that interactive Notebooks dominate requirements documentation. With respect to the problems, the main issues relate to difficulties in problem and business understanding, difficulties in managing expectations, unclear requirements, and lack of domain expert availability and engagement.

Overall, when comparing RE practices and problems within ML-enabled systems with conventional RE practices (WAGNER et al., 2019) and problems (FERNÁNDEZ et al., 2017), we identified significant variations in the practices but comparable underlying problems.

6.2 Limitations

Improving sample size and representativeness is a common opportunity for improvement in survey research. We understand that our study could benefit from having a larger sample and a more meaningful representation of other countries that are prominent within the ML-enabled systems industry, such as the US, China, and India.

Furthermore, our data collection strategy (convenience sampling) led us to a prevalence of respondents from the survey's collaborators' nationalities, especially Brazil. Nevertheless, we showed that removing data from Brazil didn't affect the contemporary practices findings. However, we did not conduct a similar analysis for the qualitative analyses.

6.3 Future Work

Future work includes developing solutions to address reported problems and identified concerns, such as the lack of proper documentation methods and missing NFRs. The close relation of traditional RE problems to the ones in RE for ML-enabled systems can help to guide such solution proposals. As comparable problems may have comparable solutions, we put forward a need to adapt and disseminate RE-related practices for engineering ML-enabled systems.

6.4 Research Publications

Table 6.1 lists the research papers written that are related to this dissertation.

Table 6.1: Publications related to this dissertation

Paper Title	**Venue**	**Status**
Status Quo and Problems of Requirements Engineering for Machine Learning: Results from an International Survey (ALVES et al., 2023b)	PROFES 2023	Accepted
ML-Enabled Systems Model Deployment and Monitoring: Status Quo and Problems	SWQD 2024	Submitted

7
Bibliography

AHMAD, K. et al. What's up with requirements engineering for artificial intelligence systems? In: IEEE. **2021 IEEE 29th International Requirements Engineering Conference (RE)**. [S.l.], 2021. p. 1–12.

ALVES, A. P. S. et al. **Status Quo and Problems of Requirements Engineering for Machine Learning: Results from an International Survey**. [S.l.]: Zenodo, 2023. <https://doi.org/10.5281/zenodo.8248333>. [Data set].

ALVES, A. P. S. et al. Status quo and problems of requirements engineering for machine learning: Results from an international survey. In: **Product-Focused Software Process Improvement - 24th International Conference, PROFES 2023, Dornbirn, Austria, December 10-13, 2023. Proceedings**. [S.l.: s.n.], 2023. p. 1–16.

AMERSHI, S. et al. Software engineering for machine learning: A case study. In: **2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)**. [S.l.: s.n.], 2019. p. 291–300.

BELANI, H.; VUKOVIĆ, M.; CAR, Z. Requirements engineering challenges in building ai-based complex systems. **2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)**, p. 252–255, 2019. Disponível em: <https://api.semanticscholar.org/CorpusID:201698152>.

BORG, M. et al. Safely entering the deep: A review of verification and validation for machine learning and a challenge elicitation in the automotive industry. **Journal of Automotive Software Engineering**, v. 1, p. 1–19, 2019. ISSN 2589-2258. Disponível em: <https://doi.org/10.2991/jase.d.190131.001>.

CHALLA, H.; NIU, N.; JOHNSON, R. Faulty requirements made valuable: On the role of data quality in deep learning. In: IEEE. **2020 IEEE Seventh International Workshop on Artificial Intelligence for Requirements Engineering (AIRE)**. [S.l.], 2020. p. 61–69.

DALPIAZ, F.; NIU, N. Requirements engineering in the days of artificial intelligence. **IEEE software**, IEEE, v. 37, n. 4, p. 7–10, 2020.

EFRON, B.; TIBSHIRANI, R. J. **An Introduction to the Bootstrap**. [S.l.]: Chapman & Hall/CRC, 1993.

FERNÁNDEZ, D. M. et al. Naming the pain in requirements engineering: Contemporary problems, causes, and effects in practice. **Empirical software engineering**, Springer, v. 22, p. 2298–2338, 2017.

GIRAY, G. A software engineering perspective on engineering machine learning systems: State of the art and challenges. **Journal of Systems and Software**, v. 180, p. 111031, 2021. ISSN 0164-1212.

HABIBULLAH, K. M.; GAY, G.; HORKOFF, J. Non-functional requirements for machine learning: Understanding current use and challenges among practitioners. **Requirements Engineering**, Springer, v. 28, n. 2, p. 283–316, 2023.

HERRMANN, A. Requirements engineering in practice: There is no requirements engineer position. In: DOERR, J.; OPDAHL, A. L. (Ed.). **Requirements Engineering: Foundation for Software Quality**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 347–361. ISBN 978-3-642-37422-7.

HORKOFF, J. Non-functional requirements for machine learning: Challenges and new directions. In: IEEE. **2019 IEEE 27th international requirements engineering conference (RE)**. [S.l.], 2019. p. 386–391.

ISHIKAWA, F.; YOSHIOKA, N. How do engineers perceive difficulties in engineering of machine-learning systems? - questionnaire survey. **2019 IEEE/ACM Joint 7th International Workshop on Conducting Empirical Studies in Industry (CESI) and 6th International Workshop on Software Engineering Research and Industrial Practice (SER&IP)**, p. 2–9, 2019. Disponível em: <https://api.semanticscholar.org/CorpusID:198981285>.

JACOBSON, I.; MEYER, B.; SOLEY, R. The semat initiative: A call for action. **Dr. Dobb's Journal**, v. 10, 2009.

JACOBSON, I.; SPENCE, I. Why we need a theory for software engineering. **Dr. Dobb's Journal**, 2009.

KAESTNER, C. Machine learning is requirements engineering—on the role of bugs, verification, and validation in machine learning. **Medium post, Accessed June**, v. 25, 2020.

KALINOWSKI, M.; CARD, D. N.; TRAVASSOS, G. H. Evidence-based guidelines to defect causal analysis. **IEEE software**, IEEE, v. 29, n. 4, p. 16–18, 2012.

KALINOWSKI, M. et al. **Engenharia de Software para Ciência de Dados: Um guia de boas práticas com ênfase na construção de sistemas de Machine Learning em Python**. [S.l.]: Casa do Código, 2023.

KALINOWSKI, M. et al. Applying dppi: A defect causal analysis approach using bayesian networks. In: SPRINGER. **Product-Focused Software Process Improvement: 11th International Conference, PROFES 2010, Limerick, Ireland, June 21-23, 2010. Proceedings 11**. [S.l.], 2010. p. 92–106.

KALINOWSKI, M.; MENDES, E.; TRAVASSOS, G. H. Automating and evaluating probabilistic cause-effect diagrams to improve defect causal analysis. In: SPRINGER. **Product-Focused Software Process Improvement: 12th International Conference, PROFES 2011, Torre Canne, Italy, June 20-22, 2011. Proceedings 12**. [S.l.], 2011. p. 232–246.

KEPHART, J.; CHESS, D. The vision of autonomic computing. **Computer**, v. 36, n. 1, p. 41–50, 2003.

KHALAJZADEH, H. et al. A survey of current end-user data analytics tool support. In: **2018 IEEE International Congress on Big Data (BigData Congress)**. [S.l.: s.n.], 2018. p. 41–48.

KIM, M. et al. Data scientists in software teams: State of the art and challenges. **IEEE Transactions on Software Engineering**, IEEE, v. 44, n. 11, p. 1024–1038, 2017.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. **Commun. ACM**, Association for Computing Machinery, New York, NY, USA, v. 60, n. 6, p. 84–90, may 2017. ISSN 0001-0782. Disponível em: <https://doi.org/10.1145/3065386>.

KUMENO, F. Sofware engneering challenges for machine learning applications: A literature review. **Intelligent Decision Technologies**, IOS Press, v. 13, p. 463–476, 2019. ISSN 1875-8843. 4. Disponível em: <https://doi.org/10.3233/IDT-190160>.

KUWAJIMA, H.; YASUOKA, H.; NAKAE, T. Engineering problems in machine learning systems. **Machine Learning**, v. 109, n. 5, p. 1103–1126, May 2020. ISSN 1573-0565. Disponível em: <https://doi.org/10.1007/s10994-020-05872-w>.

LEI, S.; SMITH, M. Evaluation of several nonparametric bootstrap methods to estimate confidence intervals for software metrics. **IEEE Transactions on Software Engineering**, v. 29, n. 11, p. 996–1004, 2003.

LINAKER, J. et al. Guidelines for conducting surveys in software engineering v. 1.1. **Lund University**, v. 50, 2015.

LORENZONI, G. et al. **Machine Learning Model Development from a Software Engineering Perspective: A Systematic Literature Review**. 2021.

LUNDERVOLD, A. S.; LUNDERVOLD, A. An overview of deep learning in medical imaging focusing on mri. **Zeitschrift für Medizinische Physik**, v. 29, n. 2, p. 102–127, 2019. ISSN 0939-3889. Special Issue: Deep Learning in Medical Physics. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0939388918301181>.

LUNNEBORG, C. E. Bootstrap inference for local populations. **Therapeutic Innovation & Regulatory Science**, v. 35, n. 4, p. 1327–1342, 2001.

LWAKATARE, L. E. et al. Large-scale machine learning systems in real-world industrial settings: A review of challenges and solutions. **Information and Software Technology**, v. 127, 2020. ISSN 0950-5849.

MARTÍNEZ-FERNÁNDEZ, S. et al. Software engineering for ai-based systems: a survey. **ACM Transactions on Software Engineering and Methodology (TOSEM)**, ACM New York, NY, v. 31, n. 2, p. 1–59, 2022.

MITCHELL, T. M. **Machine Learning**. 1997.

MORANDINI, M. et al. Engineering requirements for adaptive systems. **Requir. Eng.**, Springer-Verlag, Berlin, Heidelberg, v. 22, n. 1, p. 77–103, mar 2017. ISSN 0947-3602. Disponível em: <https://doi.org/10.1007/s00766-015-0236-0>.

NAHAR, N. et al. A meta-summary of challenges in building products with ml components–collecting experiences from 4758+ practitioners. **arXiv preprint arXiv:2304.00078**, 2023.

NASCIMENTO, E. et al. **Software engineering for artificial intelligence and machine learning software: A systematic literature review**. 2020.

PETERS, M. E. et al. Deep contextualized word representations. In: **Proceedings of the 2018 Conference of the North American Chapter of he Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)**. Association for Computational Linguistics, 2018. p. 2227–2237. Disponível em: <https://doi.org/10.18653/v1/n18-1202>.

SCHARINGER, B. et al. Can re help better prepare industrial ai for commercial scale? **IEEE Software**, v. 39, n. 6, p. 8–12, 2022.

SCHRÖER, C.; KRUSE, F.; GÓMEZ, J. M. A systematic literature review on applying crisp-dm process model. **Procedia Computer Science**, Elsevier, v. 181, p. 526–534, 2021.

SCULLEY, D. et al. Hidden technical debt in machine learning systems. **Advances in neural information processing systems**, v. 28, 2015.

STOL, K.-J.; RALPH, P.; FITZGERALD, B. Grounded theory in software engineering research: a critical review and guidelines. In: **Proceedings of the 38th International Conference on Software Engineering**. [S.l.: s.n.], 2016. p. 120–131.

STRAUSS A.; CORBIN, J. Book review: Corbin, j., & strauss, a.(2008). basics of qualitative research: Techniques and procedures for developing grounded theory . thousand oaks, ca: Sage. **Organizational Research Methods**, Sage publications Sage CA: Los Angeles, CA, v. 12, n. 3, p. 614–617, 2009.

VILLAMIZAR, H.; ESCOVEDO, T.; KALINOWSKI, M. Requirements engineering for machine learning: A systematic mapping study. In: **47th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2021, Palermo, Italy, Sep 1-3**. [S.l.: s.n.], 2021. p. 29–36.

VOGELSANG, A.; BORG, M. Requirements engineering for machine learning: Perspectives from data scientists. In: **2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)**. [S.l.: s.n.], 2019. p. 245–251.

WAGNER, S. et al. Status quo in requirements engineering: A theory and a global family of surveys. **ACM Trans. Softw. Eng. Methodol.**, Association for Computing Machinery, New York, NY, USA, v. 28, n. 2, feb 2019. ISSN 1049-331X.

WAGNER, S. et al. Challenges in survey research. **Contemporary Empirical Methods in Software Engineering**, Springer, p. 93–125, 2020.

WANG, C. et al. Understanding what industry wants from requirements engineers: An exploration of re jobs in canada. In: **Proceedings of the 12th ACM/IEEE**

International Symposium on Empirical Software Engineering and Measurement. New York, NY, USA: Association for Computing Machinery, 2018. (ESEM '18). ISBN 9781450358231. Disponível em: <https://doi.org/10.1145/3239235.3268916>.

XIONG, W. et al. The microsoft 2017 conversational speech recognition system. In: **2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. IEEE Press, 2018. p. 5934–5938. Disponível em: <https://doi.org/10.1109/ICASSP.2018.8461870>.