



Hugo Ricardo Guarín Villamizar

**Identifying Concerns When Specifying Machine
Learning-Enabled Systems: A
Perspective-Based Approach**

Tese de Doutorado

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

Advisor: Prof. Marcos Kalinowski

Rio de Janeiro
December 2023



Hugo Ricardo Guarín Villamizar

Identifying Concerns When Specifying Machine Learning-Enabled Systems: A Perspective-Based Approach

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

Prof. Marcos Kalinowski

Advisor

Departamento de Informática – PUC-Rio

Prof. Daniel M. Berry

University of Waterloo

Prof. Daniel Mendez Fernandez

BTH

Prof. Hélio Côrtes Vieira Lopes

PUC-Rio

Prof. Sérgio Lifschitz

PUC-Rio

Rio de Janeiro, December 13th, 2023

All rights reserved.

Hugo Ricardo Guarín Villamizar

Received a Bachelor degree in Information Systems from the National University of Colombia (2014), and a M.Sc in Computer Science from the Pontifical Catholic University of Rio de Janeiro (2019). His main research interests include requirements engineering, software engineering for machine learning-enabled systems, agile methods, and information security.

Bibliographic data

Guarín Villamizar, Hugo Ricardo

Identifying Concerns When Specifying Machine Learning-Enabled Systems: A Perspective-Based Approach / Hugo Ricardo Guarín Villamizar; advisor: Marcos Kalinowski. – 2023.

141 f: il. color. ; 30 cm

Tese (doutorado) - 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. Estudo de caso. I. Kalinowski, Marcos. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

To Olga, Hugo, Danilo, Paris, and Ana Paula
for their support and encouragement.

Acknowledgments

I would like to express my sincere gratitude to those who supported me during the completion of this thesis. First and foremost, I am grateful to my supervisor, Professor Marcos Kalinowski, who has been an invaluable source of guidance and support throughout my academic and professional journey. I truly appreciate the freedom he gave me to explore my ideas and his guidance to help me develop them in the right direction. Marcos has taught me the importance of being a team player, a leader, and a researcher, and has been an inspiring role model both personally and professionally. His dedication to his students, his integrity, and his willingness to invest time and effort in helping us learn and explore different paths has been truly remarkable. I will always be grateful for his mentorship and the impact he has had on my life.

I am also grateful to Professor Dan Berry, Professor Daniel Mendez, Professor Hélio Lopes, and Professor Sérgio Lifschitz for the constructive feedback as members of my thesis proposal committee and for accepting to be part of my thesis defense assessment board. I also thank Professor Tatiana Escovedo and all colleagues of the ExACTa lab for their collaboration and the friendly working atmosphere over the years. This mix of experiences helped me a lot to lay the foundations for this thesis.

Bureaucracy can make the lives of doctoral students very difficult due to the numerous setbacks and administrative issues to be resolved. I would like to thank all the administrative staff of the Informatics Department at PUC-Rio for their help, especially Alex Alves, Cosme Leal, and Vagner Pires.

Achieving a balance between academic commitments and personal life was essential to reach this point. That's why I thank Luiza Toledo, Sergio Ramirez, Anderson Uchôa, Carlos Gamboa, Murillo Nascimento, Claudia Braillard, Laure Drouvot, Ophelie De Almeida, Estiven Zuluaga, Lina Álvarez, Sergio Álvarez, Rocio Escalante, Mauricio Cespedes, Alejandra Muñoz, Angélica Ortiz, Laura Corredor, Marcelo Introini, Rafael Affonso, Marcus Soliva, Solon Tarso, and Wesley Vasques for their friendship, sense of humor, encouragement, and of course, for the beach days and the 'botecos'. You all have been my rock in the 'cidade maravilhosa'. Thank you for being there for me.

All my gratitude is also towards my beloved girlfriend, Ana Paula. I consider myself fortunate to have a partner like her. Thank you for your incredible and endless love, support, patience, sacrifice, and positive energy, and for being my

confidante in every sense of the word.

Finally, I am deeply grateful to the most important people in my life, to whom I owe what I have achieved so far: my beloved parents Hugo and Olga, and my brother Danilo. The constant support of my family, even many kilometers away, was essential for me to achieve my goals. Words cannot express the gratitude I have for them.

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

Abstract

Guarín Villamizar, Hugo Ricardo; Kalinowski, Marcos (Advisor). **Identifying Concerns When Specifying Machine Learning-Enabled Systems: A Perspective-Based Approach**. Rio de Janeiro, 2023. 141p. Tese de Doutorado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Engineering successful machine learning (ML)-enabled systems poses various challenges from both a theoretical and a practical side. Among those challenges are how to effectively address unrealistic expectations of ML capabilities from customers, managers and even other team members, and how to connect business value to engineering and data science activities composed by interdisciplinary teams. In this thesis, we studied the state of the practice and literature of requirements engineering (RE) for ML to propose *PerSpecML*, a perspective-based approach for specifying ML-enabled systems that helps practitioners identify which attributes, including ML and non-ML components, are important to contribute to the overall system's quality. The approach involves analyzing 60 concerns related to 28 tasks that practitioners typically face in ML projects, grouping them into five perspectives: system objectives, user experience, infrastructure, model, and data. Together, these perspectives serve to mediate the communication between business owners, domain experts, designers, software and ML engineers, and data scientists. The conception of *PerSpecML* involved a series of validations conducted in different contexts: (i) in academia, (ii) with industry representatives, and (iii) in two real industrial case studies. As a result of the diverse validations and continuous improvements, *PerSpecML* stands as a promising approach, poised to positively impact the specification of ML-enabled systems, particularly helping to reveal key components that would have been otherwise missed without using *PerSpecML*.

Keywords

Requirements Engineering; Machine Learning; Case study.

Resumo

Guarín Villamizar, Hugo Ricardo; Kalinowski, Marcos. **Identificando Preocupações ao especificar sistemas com componentes de aprendizado de máquina: uma abordagem baseada em perspectiva.** Rio de Janeiro, 2023. 141p. Tese de Doutorado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A engenharia de sistemas habilitados em Machine Learning (ML) bem-sucedidos apresenta vários desafios, tanto do lado teórico quanto prático. Entre esses desafios estão como abordar eficazmente às expectativas irrealistas das capacidades de ML por parte de clientes, gestores e até mesmo outros membros da equipe de desenvolvimento, e como ligar o valor do negócio às atividades de engenharia e ciência de dados compostas por equipes interdisciplinares. Nesta tese, estudamos o estado da prática e da literatura da engenharia de requisitos para ML para propor *PerSpecML*, uma abordagem baseada em perspectiva para especificar sistemas habilitados para ML que ajuda os profissionais a identificar quais atributos, incluindo componentes de ML e não-ML, são importantes para contribuir para a qualidade geral do sistema. A abordagem envolve a análise de 60 preocupações relacionadas a 28 tarefas que os profissionais normalmente enfrentam em projetos de ML, agrupando-as em cinco perspectivas: objetivos do sistema, experiência do usuário, infraestrutura, modelo e dados. Juntas, essas perspectivas servem para mediar a comunicação entre gestores de projeto, especialistas de domínio, designers, engenheiros de software/ML e cientistas de dados. A criação da *PerSpecML* envolveu uma série de validações realizadas em diferentes contextos: (i) na academia, (ii) com representantes da indústria e (iii) em dois estudos de casos industriais reais. Como resultado das diversas validações e melhorias contínuas, *PerSpecML* se destaca como uma abordagem promissora, preparada para impactar positivamente a especificação de sistemas habilitados para ML, ajudando particularmente a revelar componentes-chave que, de outra forma, teriam sido perdidos sem o uso da *PerSpecML*.

Palavras-chave

Engenharia de Requisitos; Aprendizado de Máquina; Estudo de caso.

Table of contents

1	Introduction	15
1.1	Motivation	15
1.2	Research Goal and Questions	18
1.3	Research Method	20
1.4	Contributions	21
1.5	Thesis Organization	23
2	Background and Related Work	24
2.1	Introduction	24
2.2	Requirements Engineering (RE)	24
2.3	Machine Learning (ML)	25
2.4	ML projects	26
2.5	ML-Enabled Systems	27
2.6	RE for ML-Enabled Systems	28
2.7	Related Work	30
2.8	Concluding Remarks	33
3	A Systematic Mapping Study on RE for ML	34
3.1	Introduction	34
3.2	Related Work	35
3.3	Protocol	35
3.4	Results	40
3.5	Discussion	46
3.6	Threats to Validity	47
3.7	Concluding Remarks	47
4	A Catalog of Concerns for ML-Enabled Systems	49
4.1	Introduction	49
4.2	Background	50
4.3	Methodology	50
4.4	The Catalog	51
4.5	Focus Group	52
4.6	Discussion	55
4.7	Concluding Remarks	56
5	An Approach for Identifying Concerns When Specifying ML-Enabled Systems	58
5.1	Introduction	58
5.2	Methodology	59
5.3	<i>PerSpecML</i>	62
5.4	Validation in Academia	83
5.5	Static Validation	91
5.6	Dynamic Validation	101
5.7	Threats to Validity	113
5.8	Discussion	115

5.9	Concluding Remarks	116
6	Contributions, Limitations, and Future Work	118
6.1	Contributions	118
6.2	Limitations	119
6.3	Future Work	120
7	Bibliography	122
A	RE for ML Papers Identified in the Literature Review	130
B	Details of the Relationship of Concerns by Perspective	135
C	List of Publications	139

List of figures

Figure 1.1	A recent popular tweet about the importance of RE for ML by Pochetti, who was named a “Machine Learning Hero” by Amazon Web Services in 2019.	16
Figure 1.2	Technology transfer model proposed by (GORSCHEK et al., 2006).	20
Figure 1.3	Overview of the research method. The activities with blue backgrounds represent work directly related to this thesis, and the red backgrounds represent work not directly attributed to this thesis. For references, please see the list of publications in Appendix C.	21
Figure 1.4	A high-level view of <i>PerSpecML</i> . The components with green, yellow, red, and blue colors represent the stakeholders, the perspectives, the catalog of concerns, and the artifacts that compose <i>PerSpecML</i> , respectively.	21
Figure 2.1	Characterization of ML projects adapted from (AHO et al., 2020).	27
Figure 2.2	Components of a transcription service with an ML component: from models to products (KÄSTNER, 2022).	28
Figure 3.1	Papers selection process.	38
Figure 3.2	Distribution of the papers per RE activity.	41
Figure 3.3	Distribution of research type per year.	45
Figure 3.4	Distribution of empirical evaluation type per year.	46
Figure 4.1	Overview of the study steps for defining the catalog.	51
Figure 4.2	An overview of the catalog of concerns to support the specification and design of ML-enabled systems (VILLAMIZAR; KALINOWSKI; LOPES, 2022).	52
Figure 5.1	Candidate solution for identifying concerns when specifying ML-enabled systems. (VÍLLAMIZAR; KALINOWSKI; LOPES, 2022).	61
Figure 5.2	An illustration of the perspective-based ML task and concern diagram.	75
Figure 5.3	Elements of the Perspective-Based ML Specification Template.	76
Figure 5.4	Excerpt of the Perspective-Based ML Specification Template for the model and data perspectives.	77
Figure 5.5	Logical flow for executing <i>PerSpecML</i> .	78
Figure 5.6	Process diagram for the academic validation.	84
Figure 5.7	Frequencies of the relevance of each perspective of the candidate solution.	88
Figure 5.8	Frequencies of the TAM constructs for academic validation.	90
Figure 5.9	Process diagram for the static validation in industry.	93
Figure 5.10	Frequencies of the TAM constructs for static validation industry.	100
Figure 5.11	Process diagram for the dynamic validation in industry.	103
Figure 5.12	Frequencies of the TAM constructs for dynamic validation in industry.	112

List of tables

Table 3.1	Exclusion criteria.	38
Table 3.2	Data extraction form.	40
Table 3.3	Identified contributions from the literature review.	42
Table 3.4	Frequency of quality characteristics from the literature review.	43
Table 4.1	Overview of the participants of the focus group who evaluated the catalog of concerns.	53
Table 5.1	Description of the tasks to define the system objectives.	65
Table 5.2	Description of the tasks to ensure user experience.	66
Table 5.3	Description of the tasks to support the infra of ML-enabled systems.	66
Table 5.4	Description of the tasks to support the creation of ML models.	67
Table 5.5	Description of the tasks to support data quality in ML projects.	68
Table 5.6	Description of each concern of the system objectives perspective.	69
Table 5.7	Description of each concern of the user experience perspective.	70
Table 5.8	Description of each concern of the infrastructure perspective.	71
Table 5.9	Description of each concern of the model perspective.	72
Table 5.10	Description of each concern of the data perspective.	73
Table 5.11	Legend of the perspective-based ML task and concern diagram.	74
Table 5.12	Specification of the concerns of the system objectives perspective.	81
Table 5.13	Specification of the concerns of the infrastructure perspective.	82
Table 5.14	Study goal definition of academic validation.	85
Table 5.15	Subjects involved in the validation in academia.	86
Table 5.16	Projects involved in the static validation.	92
Table 5.17	Study goal definition of the static validation.	93
Table 5.18	Subjects involved in the static validation in industry.	94
Table 5.19	ML-enabled systems involved in the dynamic validation.	102
Table 5.20	Study goal definition of the dynamic validation.	103
Table 5.21	Subjects involved in the dynamic validation in industry.	105
Table B.1	Relationships between system objectives perspective concerns.	135
Table B.2	Relationships between user experience perspective concerns.	136
Table B.3	Relationships between infrastructure perspective concerns.	136
Table B.4	Relationships between model perspective concerns.	137
Table B.5	Relationships between data perspective concerns.	138

List of Abbreviations

AI – Artificial Intelligence

BS – Backward Snowballing

FS – Forward Snowballing

FRs – Functional Requirements

GQM – Goal-Question-Metric

ML – Machine Learning

NLP – Natural Language Processing

NFRs – Non-Functional Requirements

PoC – Proof-of-Concept

RE – Requirements Engineering

R&D – Research & Development

SE – Software Engineering

SMS – Systematic Mapping Study

*To successfully complete the doctoral thesis,
the results must overcome the excuses.*

Ignacio Mantilla, *Former President of the National University of Colombia.*

1

Introduction

1.1

Motivation

Contemporary advances in Machine Learning (ML) and the availability of vast amounts of data have both given rise to the feasibility and practical relevance of incorporating ML components into software-intensive systems. These types of systems have their behavior dictated by explicitly defined rules and data used by the ML component to make decisions. This shift from engineering purely conventional software systems to ones which have ML-components woven-in poses new challenges from the viewpoint of software engineering (SE). Moreover, there are other particularities that demand extra effort to successfully develop such systems:

- **Additional concerns that extend the concept of quality.** A good learned system is one in which, *e.g.*, the learning evolves over time, the ML model deals with fairness and transparency, the users are aware of how often the ML model outputs are right and wrong, and the customer knows how much ML techniques help the system to achieve their goals.
- **When practitioners evaluate ML-enabled systems they often look at measures such as accuracy, precision and recall.** However, it is important to understand the big picture of the constraints these systems put on the overall development. Where will the model be executed? What data will it have access to? How fast does it need to be? What is the business impact of a false negative? How should the model be tuned to maximize business results? The model is just a component of a larger system. There are other components that require attention, such as the infrastructure to deploy and serve the model, the integration of the model with the rest of the system functionality, and a user interaction design to build better experiences of using the model (HULTEN, 2019).
- **The introduction of ML components in software projects has created the need for software engineers to collaborate with data scientists and other specialists.** These roles do not usually have the same background and there can be cultural differences. When misaligned due to incorrect assumptions, may cause ML mismatches which can result in failed systems (LEWIS; BELLOMO; OZKAYA, 2021; NAHAR et al., 2022).

In recent years, multiple ML projects have failed, leading to severe repercussions for the organizations involved and to the society at large (FRY, 2018; BEEDE et al., 2020). Gartner reports that only 53% of ML projects make it from prototype to production (GARTNER, 2020). The reason is often the same: systems that incorporate ML components tend to put stakeholder needs in the background, and to oversimplify important scenarios and trade-offs. This leads to a problem that can be tackled by the requirements engineering (RE) discipline. Figure 1.1 summarizes this special connection between RE and ML.

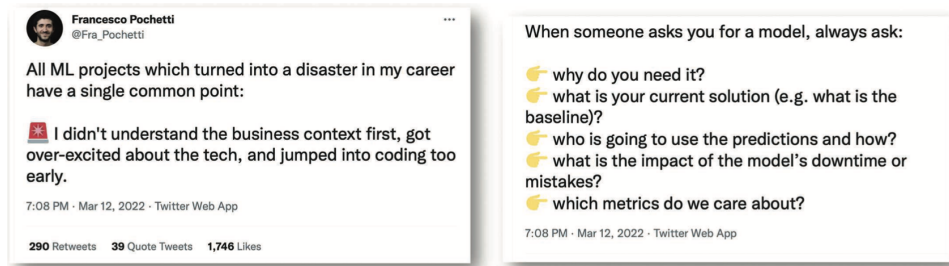


Figure 1.1: A recent popular tweet about the importance of RE for ML by Pochetti, who was named a “Machine Learning Hero” by Amazon Web Services in 2019.

Due to the communication and collaboration-intensive nature, as well as inherent interaction with most other development processes, RE can provide the very foundation to address several of the challenges of building ML-enabled systems (KÄSTNER, 2022). For example, 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 software systems where we need to identify representative stakeholders, validate specifications with customers, and address conflicting requirements. However, establishing RE in ML projects may be difficult due to two principal factors:

- **The position ‘requirements engineer’ hardly exists (HERRMANN, 2013; WANG et al., 2018; ALVES et al., 2023).** Business analysts, software engineers, and project managers typically perform RE activities informally. In the context of ML projects, assuming dedicated RE professionals seems unrealistic. In such cases, it is important that stakeholders (*e.g.*, ML engineers, product owners, scrum masters, and designers) understand basic RE knowledge to engage in RE activities by themselves and collaborate together from the early phases with specific emphasis on RE.
- **Recent studies emphasize that practitioners find RE as the most difficult activity of ML projects (ISHIKAWA; YOSH-**

IOKA, 2019; KUWAJIMA; YASUOKA; NAKAE, 2020; NAKAHAR et al., 2023; ALVES et al., 2023). This can be explained by several reasons. Requirements for ML are more uncertain than requirements for traditional software systems, new quality properties come into focus, setting more ambitious goals, dealing with a high degree of iterative experimentation, and facing unrealistic customer assumptions increases the level of complexity. In addition, the limited research on RE for ML difficult its application (VILLAMIZAR; ESCOVEDO; KALINOWSKI, 2021; AHMAD et al., 2023a).

This landscape has caught a new level of interest by the research community trying to better understand how RE techniques can be extended and what challenges need to be addressed to reliably build ML-enabled systems (DALPIAZ; NIU, 2020). In this way, studies have emerged related to issues with data requirements (CHALLA; NIU; JOHNSON, 2020), the understanding of the RE process in ML projects (VOGELSANG; BORG, 2019), non-functional requirements and particularities of certain quality attributes such as explainability, transparency and fairness (HABIBULLAH; GAY; HORKOFF, 2023; CYSNEIROS; RAFFI; LEITE, 2018; MARTÍNEZ-FERNÁNDEZ et al., 2022).

Despite these valuable contributions in the field so far, the importance of specifying ML components in a way that customers can understand and analyze it to make adequate decisions is too often overlooked (NASCIMENTO et al., 2019), and only a limited number of studies have looked into how to specify and document requirements for ML-enabled systems (VILLAMIZAR; ESCOVEDO; KALINOWSKI, 2021; PEI et al., 2022; AHMAD et al., 2023a). As a consequence, many ML-enabled systems lack requirements specifications (LWAKATARE et al., 2019; KUWAJIMA; YASUOKA; NAKAE, 2020), which is mainly due to the difference in the building process between these systems and traditional ones. Indeed, a recent roadmap for the future of SE emphasizes that existing RE methods will need to be expanded to decouple ML problem and model specification from the system specification (CARLETON et al., 2021).

Motivated by these studies and to tap the potential of RE in supporting the successful development of any software system that meets user, system, and business requirements, we propose and evaluate a perspective-based approach that helps identifying concerns when specifying ML-enabled systems. The approach emerged from analyzing the particularities of the ML domain, the literature, and our own experience with ML projects.

The focus of this thesis is on those types of software systems incorporating components that leverage ML algorithms and techniques to perform prediction tasks based on learning from data that was not explicitly programmed, where these tasks can refer to regressions and classifications. Hereafter, we refer to these systems as ML-enabled systems. However, it is noteworthy that other terms have been used for equivalent purposes, such as ML system, intelligent system and data-driven system. In this thesis, the proposed approach focuses on the entire software system, including the synergy between the ML and non-ML components.

We acknowledge that the usage of terms such as data science, data analytics, advanced analytics, artificial intelligence (AI), and ML have been evolving in recent years and may be used with different connotations by researchers and practitioners from different domains and industries.

1.2

Research Goal and Questions

Aiming to support the development of ML-enabled systems, requirements specification as a field of study is concerned with a complete description of what software is being developed is supposed to do (LAMSWEERDE, 2009). Therefore, well-defined and decoupled specifications, as a cornerstone of RE, can offer considerable value in effective design and implementation of ML-enabled systems (KÄSTNER, 2022; CARLETON et al., 2021; BERRY, 2022; AHMAD et al., 2023a).

The **main research question** in this thesis is as follows:

How can RE support the identification and specification of key components and attributes of ML-enabled systems?

To tackle this question, this thesis aims to **develop and evaluate an approach that helps to identify and specify key components and attributes that need to be defined for ML-enabled systems.**

To address the main research question of this thesis and contributing to RE for ML in a broader way, the following more specific research questions were formulated:

RQ1 *What is the state of the art of RE for ML?*

In traditional software systems, RE activities are well-established and researched. However, building ML-enabled systems with limited or no insight into the system's inner workings poses significant new challenges to RE. Existing literature mainly focuses on using

ML techniques to support RE activities rather than on exploring how RE can improve the development of such systems. By answering this research question, the aim was to understand the current practices and challenges pointed out by the literature. For this, we conducted a systematic mapping study (SMS) to find papers on current RE for ML approaches and analyzed the existing contributions in terms of RE activities, quality characteristics, challenges, limitations, and empirical evaluations. The description of the study is elaborated in Chapter 3. In addition, we participated in real ML projects and collaborated with an international survey to gather practitioner insights into the status quo and problems of RE for ML (ALVES et al., 2023).

RQ2 *What are the key components and properties that should be covered by practitioners when specifying ML-enabled systems, and how can a structured catalog of concerns support them effectively?*

Achieving success with ML-enabled systems goes beyond just developing models and handling data. It encompasses a holistic view that includes alignment with the business context, ensuring a seamless user experience, and maintaining a robust infrastructure. However, when it comes to specifying the requirements for such systems, there exists a challenge in recognizing and addressing a wide range of concerns, many of which are not easily identified. These ‘hidden requirements’ pose a substantial obstacle in the development of effective ML-enabled systems. Hence, it is crucial to identify them. To answer this research question, a catalog was proposed and evaluated, and the study is described in Chapter 4.

RQ3 *How can the integration of the proposed catalog of concerns into the specification process of ML-enabled systems lead to more effective, transparent, and reliable development outcomes?*

The solutions to support the specification of ML-enabled systems are not well established, and their consideration is in the initial stage. Therefore, it is important to develop these types of solutions in a structured way. In Chapter 5, we present the first effort to address this question with an early modeling of the proposed catalog of concerns for specifying ML-enabled systems (output of RQ2). Subsequently, we refined and improved it based on a set of evaluations conducted both in academia and industry. The resulting approach was called *PerSpecML*.

1.3

Research Method

Given the problem-solution finding nature of our research goal and questions, the work in this thesis falls into the paradigm of design science. Towards this, the research activities of this thesis followed the technology transfer model proposed by (GORSCHEK et al., 2006), which is recommended to foster successful transfer of technology from research to practice (WOHLIN et al., 2012). Figure 1.2 outlines the seven steps of the model.

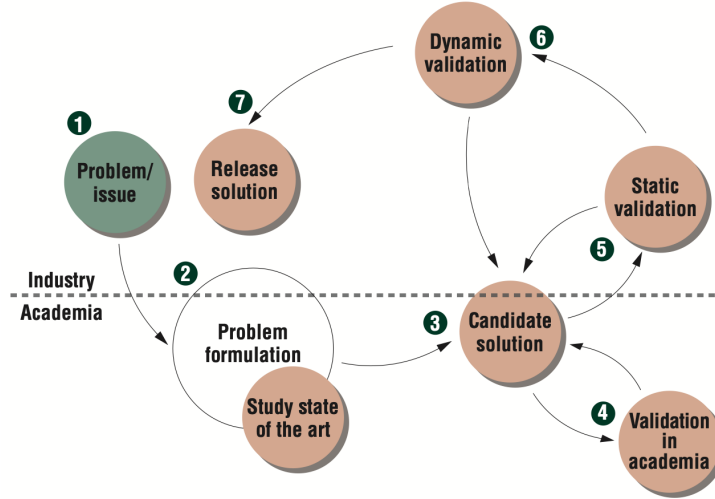


Figure 1.2: Technology transfer model proposed by (GORSCHEK et al., 2006).

An overview of the specific research method is presented in Figure 1.3, which maps the activities, research questions, and methods used in this thesis. First, we participated in real ML projects of a research and development (R&D) initiative (KALINOWSKI et al., 2020) (activity 1). To complement our understanding in the field, we conducted a literature review on RE for ML (VILLAMIZAR; ESCOVEDO; KALINOWSKI, 2021) (activity 2) and an international survey to gather practitioner insights into the status quo and problems of RE in ML-enabled systems (activity 3).

After exploring the problem space, we created a catalog of concerns (VILLAMIZAR; KALINOWSKI; LOPES, 2022) (activity 4) and proposed a candidate solution for specifying ML-enabled systems (VILLAMIZAR; KALINOWSKI; LOPES, 2022) (activity 5). We iteratively evaluated and improved the candidate solution towards a reaching the *PerSpecML* approach (Activity 6) by conducting three studies in different contexts: (i) in an academic validation involving two courses on SE for data science (activity 7), (ii) with practitioners working with ML-enabled systems in an R&D initiative (activity 8), and (iii) in two real industrial case studies conducted with a Brazilian large e-commerce company (activity 9).

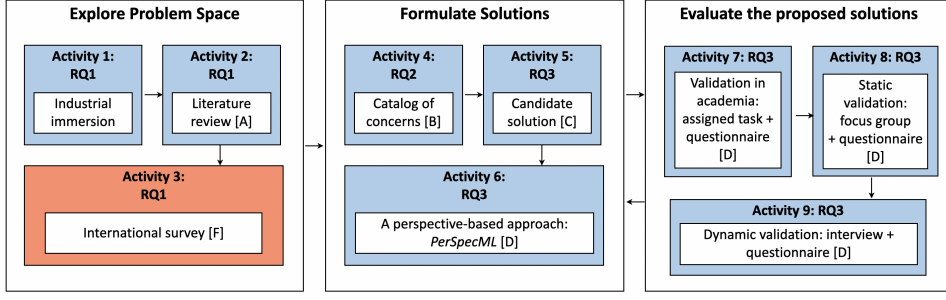


Figure 1.3: Overview of the research method. The activities with blue backgrounds represent work directly related to this thesis, and the red backgrounds represent work not directly attributed to this thesis. For references, please see the list of publications in Appendix C.

1.4 Contributions

Despite remarkable contributions of the RE and SE community, there has been little attention paid to requirements specification and documentation for ML. While different quality properties and modeling techniques have been proposed to assist the design of ML-enabled systems, we are not aware of any approach that provides a holistic view of their properties to support the specification of such systems. Figure 1.4 shows a high-level view of *PerSpecML* (VILLAMIZAR et al., 2023), the approach conceived in this thesis.

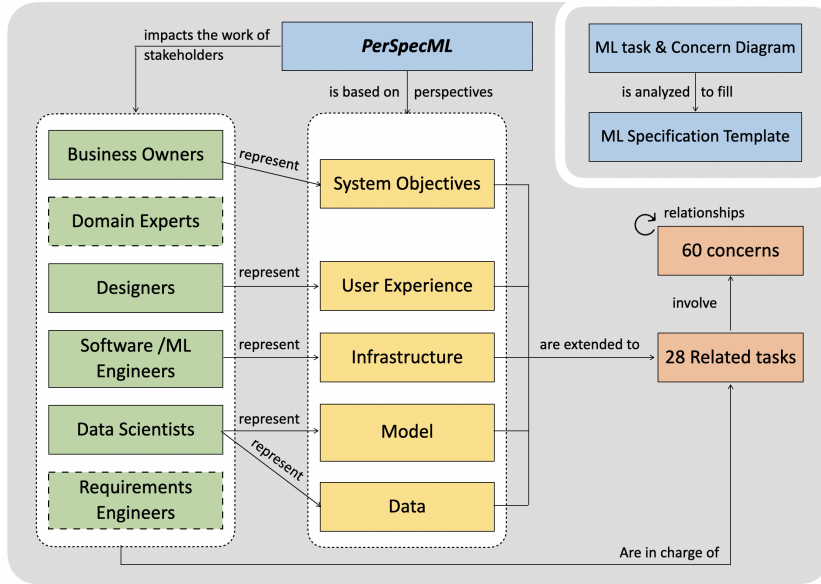


Figure 1.4: A high-level view of *PerSpecML*. The components with green, yellow, red, and blue colors represent the stakeholders, the perspectives, the catalog of concerns, and the artifacts that compose *PerSpecML*, respectively.

PerSpecML involves analyzing a set of concerns related to typical tasks that stakeholders face in ML projects, grouping them into five perspectives:

system objectives, user experience, infrastructure, model, and data. This thesis advances existing approaches by:

- Enabling the identification of requirements to specify ML-enabled systems.
- Supporting practitioners of ML projects in the design of ML-enabled systems, reflecting on how to address business, user, system, model, and data requirements.
- Providing a definition of a requirements specification for ML-enabled systems considering concerns at different levels from customers, users and model constraints to data preparations, and system operations.
- Modeling the expected value and operation of ML-enabled systems in the form of tasks and related concerns that are expressed in a conceptual diagram.
- Providing empirical evaluation of the proposed approach to strengthen the contributions and facilitate their adoption in practice.

From a practical point of view, based on the findings of the empirical investigations, *PerSpecML* can help practitioners:

- Enhance clarity of the ML workflow.
- Cover hidden and overlooked requirements for ML projects.
- Identify trade-offs between conflicting objectives and requirements, improving decision-making.
- Foster collaboration between team members.

In addition to the above, another contribution of this thesis is the literature review on RE for ML (VILLAMIZAR; ESCOVEDO; KALINOWSKI, 2021) that was driven by the need to synthesize the existing body of knowledge in a relatively nascent field. As a result, the literature review not only contributes to filling a critical gap in the research landscape but also laid a reference point for other researchers and practitioners.

1.5

Thesis Organization

This thesis is organized as follows:

- Chapter 2 provides an overview of the background and related work.
- Chapter 3 summarizes the existing literature on RE for ML, including gaps, challenges, and opportunities in the field that contribute to define the scope of this thesis.
- Chapter 4 describes the development, key concepts, and structure of a catalog of concerns for specifying ML-enabled systems.
- Chapter 5 presents the core contribution of the thesis, *PerSpecML*, a perspective-based approach for identifying key components when specifying ML-enabled systems that uses the catalog presented in Chapter 4. Here, we describe the conception, development, and key features of this approach, and share the results and insights from three evaluations conducted to validate the approach.
- Chapter 6 presents a summary of the contributions and limitations along with directions for future work.

2

Background and Related Work

2.1

Introduction

This chapter provides a comprehensive introduction to the foundational concepts for understanding this thesis. It delves into key terminologies and background information, defining fundamental elements such as the characteristics of ML projects, RE activities, and the distinctive features of ML itself. Additionally, a concise summary of the challenges inherent in RE for ML-enabled systems is presented. Furthermore, the chapter offers an insightful overview of the existing body of related work, setting the stage for a deeper exploration of the subject matter.

2.2

Requirements Engineering (RE)

RE constitutes approaches to understand the problem space and specify requirements that all stakeholders agree upon (DAMIAN, 2007). As such, it concentrates on understanding what the actual problem is, what needs towards a system result and how to resolve potential conflicts, and it is thus characterized by the involvement of interdisciplinary stakeholders and often resulting in uncertainty (WAGNER et al., 2019).

There are two main types of requirements in SE:

- **Functional Requirements (FRs)** are specifications that define the functions or operations that the software system must carry out. For instance, “The system shall allow users to log in using their email and password to access their personalized dashboard” might be a FR for an online social media (LOUCOPOULOS; KARAKOSTAS, 1995).
- **Non-Functional Requirements (NFRs)** are specifications that define a software system’s qualities or constraints (CHUNG et al., 2012). Performance, usability, and security are a few NFRs examples.

On the other hand, the traditional requirements activities usually involve (NUSEIBEH; EASTERBROOK, 2000):

- **Elicitation** aims to gather information from stakeholders about the system. This can be done through interviews, surveys, seminars, workshops, or other techniques.

- **Analysis** aims to examine the elicited material to look for contradictions, ambiguities, conflicts, or any type of defect in the requirements.
- **Specification** aims to create a formal document including FRs and NFRs. This document acts as a contract between the development team and the stakeholders. While the proposed approach of this thesis is mainly focused on supporting this RE activity, other activities such as analysis and validation can leverage the capabilities offered by the approach.
- **Validation** aims to evaluate the requirements specification, involving the development team and stakeholders, to make sure it appropriately reflects the demands of the stakeholders.
- **Management** aims to monitor changes in requirements to ensure they do not affect the overall project schedule or budget. The requirements specification is managed throughout the development process.

Understanding, defining, and comprehending different types of requirements is very crucial and important as a part of SE because it guides and enables software developers to develop software systems that satisfy the stakeholders' needs. It is well known, that poorly defined requirements can have serious consequences (FERNÁNDEZ et al., 2017). These may include project delays, cost overruns, scope creep, and, ultimately, the delivery of a system that does not meet the user's expectations. Inaccurate or incomplete requirements can lead to misunderstandings between stakeholders and development teams, resulting in rework and increased project complexity.

In theory, requirements activities are defined before designing and implementing any part of the system, and performed by requirements engineers. However, this is rarely applied in practice (HERRMANN, 2013; WANG et al., 2018). Dedicated roles of requirements engineers are often missing, and dedicated RE activities are usually carried out informally, unless they are required by contract or compliance rules, e.g., in safety-critical software projects.

2.3

Machine Learning (ML)

ML is a sub-field of AI that involves the study of algorithms and statistical models that allow software systems to learn and make predictions based on data (JORDAN; MITCHELL, 2015). By recognizing patterns in the data they are trained on, ML algorithms are developed to automatically improve over time. The availability of vast amounts of data have both given

rise to the feasibility and practical relevance of incorporating ML components into software-intensive systems.

ML can be categorized into two main types of learning: supervised and unsupervised (JORDAN; MITCHELL, 2015). In this thesis, we are mainly interested in the supervised category where an ML algorithm learns from labeled data, making it a guided learning process. In other words, the ML algorithm tries to learn the mapping from input data to the corresponding target or output labels. It is widely used in various fields, including computer vision, natural language processing (NLP), and recommendation systems.

In supervised learning two main tasks can be performed:

- **Classification.** ML models classify data into predefined categories. For example, they can categorize emails as spam or not spam, classify images into specific objects or animals, or classify text for sentiment analysis.
- **Regression.** ML models predict continuous values. They can be used to predict housing prices based on features, stock prices, or even the age of a person based on certain health indicators.

2.4

ML projects

ML has rapidly gained prominence and are increasingly integrated into various software projects due to its ability to extract valuable insights from vast datasets, automate complex tasks, and enhance decision-making processes. The process flow for an ML project differs significantly from that of traditional software systems. Figure 2.1 shows the three main characteristics of ML projects according to (AHO et al., 2020): experimentation, development approach, and multidisciplinary teams.

ML projects involve a high degree of **experimentation** and dealing with the uncertainty outcomes. Data scientists need to experiment with data, models, and algorithms to find the most satisfying way of meeting their goals. Knowledge gained during the experimentation phase may lead to changes in goals or requirements, to more accurate models. The **development approach** in ML projects seems to incorporate data scientists into larger development teams. The work is also clearly iterative in its nature. On the other hand, ML projects often are executed as small Proof-of-Concept (PoC) efforts that eventually make it into production. In this setting, **a multidisciplinary team** is required, *e.g.*, domain experts and software development, to complement the data science and engineering skills.

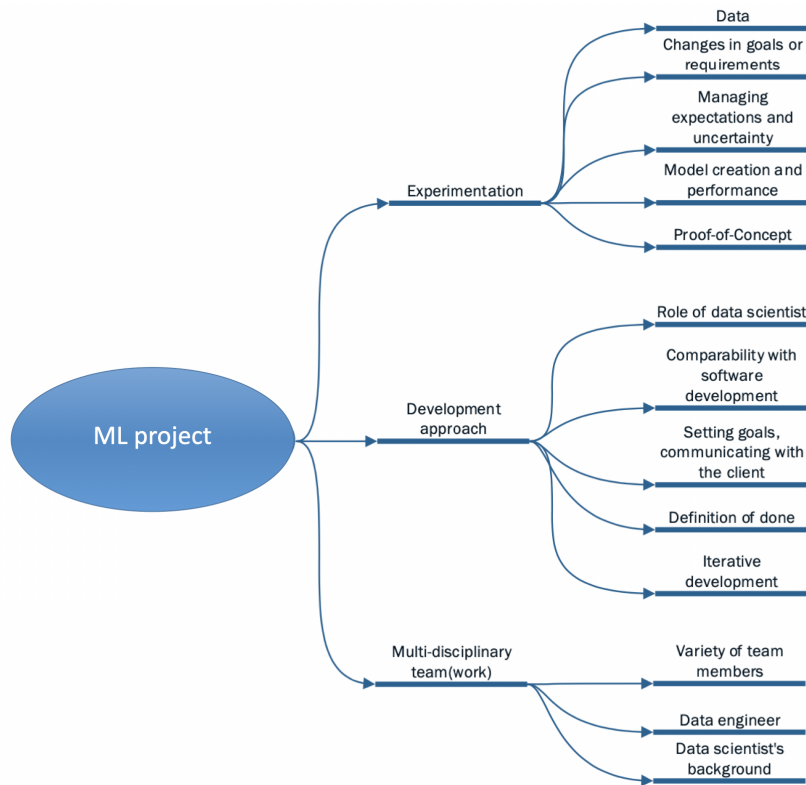


Figure 2.1: Characterization of ML projects adapted from (AHO et al., 2020).

2.5

ML-Enabled Systems

Much of the attention in ML education has been on learning how specific ML algorithms work (*e.g.*, understanding overfitting and underfitting concepts) or how to apply them to train accurate models from provided data (KÄSTNER; KANG, 2020). Similarly, ML research focuses primarily on the learning steps, trying to improve prediction accuracy of models trained on common datasets (*e.g.*, exploring new deep neural network architectures). Comparatively little attention is paid at how the learned models might actually be used for a real task, and how systems might use the model’s predictions.

Throughout this thesis, we mention ML-enabled systems, which are production systems where ML is used as a component within a larger system. Figure 2.2 illustrates how ML provides the core functionality of a system that converts uploaded audio files into text. In this example, other non-ML parts are needed, such as a user interface, a data storage and processing infrastructure, a payment service, and monitoring infrastructure to ensure the system is operating as expected.

The quality of ML-enabled systems goes beyond ML model performance metrics such as accuracy, precision or recall. This implies taking care of not only data and models, but also business context, user interactions, and integration

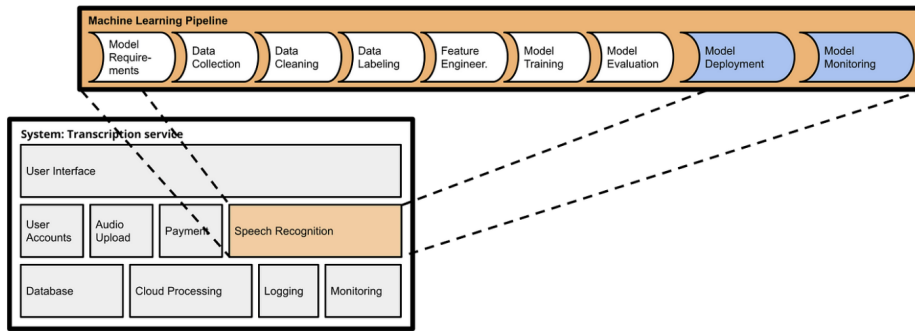


Figure 2.2: Components of a transcription service with an ML component: from models to products (KÄSTNER, 2022).

of several services. One key challenge arises at the interface between these ML components and non-ML components, and how they, together, achieve system goals. There is an incredible amount of work to be done between the development of an ML model, the incorporation of it into a system and the eventual sustainable customer impact (KUWAJIMA; YASUOKA; NAKAE, 2020; ISHIKAWA; YOSHIOKA, 2019; BELANI; VUKOVIC; CAR, 2019). Thinking about possible strategies to address these concerns increases the chance of designing and development an ML-enabled system that meets customer’s needs, and can avoid often costly problems later.

2.6

RE for ML-Enabled Systems

Requirements and ML models have a special connection. There is a perception that ML corresponds to the RE phase of a project rather than the implementation phase and, as such, terminology that relates to validation (*i.e.*, do we build the right system, given stakeholder needs) is more suitable than terminology that relates to verification (*i.e.*, do we build the system right, given a specification). In that sense, an ML model can be seen as a specification based on training data since data is a learned description of how the ML model shall behave. This means that the learned behavior of an ML model might be incorrect, even if the learning algorithm is implemented correctly. This landscape gives an essential place to RE in the development of ML-enabled systems.

Given the data-driven nature of such systems, data requirements have become a new category of requirements. Understanding the data, its features, and its distributions is key to ensuring the quality assurance of ML models (WAN et al., 2019). To understand the data, one also has to understand the context or the domain from where the data is gathered, thus, this should be specified (HEYN et al., 2021). The main argument for this is that the per-

formance of the ML model can change drastically if it were to be deployed in another context than it was developed for. This arises another important and often overlooked concern should be accounted for in the requirements processes: ML model degradation over time. To deal with this, (VOGELSANG; BORG, 2019) highlight the fact that ML models regularly need to be retrained, and (ARPTTEG et al., 2018) add that this maintenance of the ML model requires plenty of resources.

The large degree of uncertainty in the development of ML components heavily affects RE. For instance, several studies have surveyed practitioners around the world and found that unpredictability makes it difficult to define any criteria or requirements regarding the output of ML-enabled systems, and also introduces a challenge in the collaboration with stakeholders (ISHIKAWA; YOSHIOKA, 2019; ALVES et al., 2023). This may lead stakeholders to have a wrong perception of what ML is capable of (GIRAY, 2021).

Recent attention has been paid to NFRs for ML-enabled systems, and there is a consensus about the lack of knowledge regarding quality characteristics and potential trade-offs between them (VILLAMIZAR; ESCOVEDO; KALINOWSKI, 2021; AHMAD et al., 2021). In order to help contribute to this gap, (HABIBULLAH; GAY; HORKOFF, 2023) conducted interviews and a survey to understand how NFRs are perceived among practitioners of ML projects from both industry and academia. They found several challenges:

- Accuracy, reliability, integrity, and security are particularly important.
- Most practitioners focused on defining NFRs over the whole system. Several also define NFRs on models, and few considered NFRs for data.
- NFRs challenges relate to uncertainty, domain dependence, awareness, regulations, dependency among requirements, and specific NFRs (e.g., safety, transparency, and completeness).

From a broader perspective on how RE works in ML projects, other studies have reported that ML-enabled systems are rarely built based on comprehensive specifications (LWAKATARE et al., 2019; LEWIS; BELLOMO; OZKAYA, 2021), and such specifications are often developed by data scientists and management roles of the projects (VOGELSANG; BORG, 2019). According to (BERRY, 2022), the literature and practice of AI development, including ML, does not clarify what a requirements specification of an AI actually should be in order to allow determining whether an implementation of the AI is correct. Berry states that the measures used to evaluate a learned machine, the criteria for acceptable values of these measures, and the information about the ML context that inform the criteria and trade-offs in these measures

collectively constitute the requirements specification of these types of systems. As important as the definition, (WAN et al., 2019) argue that requirement engineers need to adapt to this and build a good technical understanding of ML to be able to identify and specify requirements correctly.

Overall, as there are significant differences between the development of ML-enabled systems and the development of traditional software systems (WAN et al., 2019), adaptation and rework are needed in RE for ML (MAALEJ; PHAM; CHAZETTE, 2023).

2.7 Related Work

In this section, we highlight research that has investigated what quality attributes should be analyzed and how practitioners can identify, specify and document requirements for ML-enabled systems. We further take a more holistic RE perspective where an ML model is part of a larger software system.

(DORARD, 2015) proposed a management template for ML, also known as *ML Canvas*, that can be used to describe how ML systems will turn predictions into value for end-users, considering elements such as problem definition, data collection and preparation, feature engineering, model selection, evaluation metrics, deployment, and monitoring. This is probably the most spread approach for documenting ML-enabled systems given its simplified representation. However, this can be seen as a limitation since *ML Canvas* may not capture all the intricate details and complexities of real-world projects, leading to potential oversights or gaps in the analysis. We seek to bridge these gaps with *PerSpecML* by focusing on five different perspectives covering technical aspects and broader contextual concerns such as ethical considerations, legal constraints, and business implications, which can be crucial in real-world implementations.

(RAHIMI et al., 2019) discussed on ideas for extracting and visualizing safety-critical requirements specifications and how a self-driving car would recognize pedestrians. The authors describe how RE can be useful to better understand the domain and context of a problem and how this helps to better select a high-quality dataset for model training and evaluation. We are aware that identifying gaps in the associated dataset and the constructed ML model is essential to improve the overall quality, fairness, and long-term effectiveness of the ML-enabled system, but at the same time other external components such as those related to the operation (*e.g.*, data streaming and ML model performance degradation) play an important role and can make the difference between an ML-enabled system that fits customer’s needs and one that doesn’t.

In an effort to model the representation of data-driven systems, several works have been proposed. For instance, (CHUPRINA; MENDEZ; WNUK, 2021) presented an artefact-based RE approach that encompasses four layers: context, requirements, system, and data. While the context specification captures the operational environment of a system, the requirements specification covers the user-visible black-box behaviour and characteristics such as explainability, transparency and ethics. On the other hand, the system specification defines the solution space and considers the system in a glass box view. The data-centric layer captures artifacts such as training and test datasets, and verifying algorithms.

Similarly, (NAKAMICHI et al., 2020) proposed a requirements-driven model to determine the quality attributes of ML-enabled systems that covers perspectives such as environment/user, system/infrastructure, model, data and quality characteristics. Despite the important contributions of these works, we found some limitations when compared to *PerSpecML*. Firstly, our intention is to be more specific, including more fine-grained attributes for each layer/perspective and modeling their relationships so that practitioners can have a complete view of the ML context and the software system as a whole. Secondly, we detail ML-related concerns that we faced in practice that were not considered as part of their proposals, such as concerns related to business requirements and user experience, which in our context showed being important for the success of ML-enabled systems.

Another study we consider relevant is one conducted by (NALCHIGAR; YU; KESHAVJEE, 2021). They reported on an empirical study that evaluates a conceptual modeling framework for ML solution development for the healthcare sector. It consists of three views consumed by business people, data scientists, and data engineers. The business view shows how business goals are refined into decision goals and question goals, and how such questions can be answered by ML. The analytic design view models a solution in terms of algorithms, non-functional requirements and performance indicators. Lastly, the data preparation view conceptualizes the design of data preparation tasks in terms of data tables, operations, and flows. We also find this work as relevant as the previous ones, but we believe that other views related to the operation of ML-enabled systems such as infrastructure and user experience must be considered to support the activities of practitioners such as software and ML engineers, and designers.

(SIEBERT et al., 2022) presented a formal modelling definition for quality requirements in ML-enabled systems that allows to identify attributes and quality measures related to components such as model, data, system,

infrastructure and environment. We consider this work strongly related to ours. For instance, the authors discuss quality attributes of an ML-enabled system beyond the ML components, just as *PerSpecML* proposes. It is also explicit about considering multiple perspectives: of the entire system, and of the environment the system is embedded in. As a key difference between the works, we provide a diagram that summarizes the perspectives, the quality attributes/concerns, and shows their relationships. This seeks to support the communication and collaboration among stakeholders, provide a visual representation that can be easily understood by technical and non-technical team members, capture and document various aspects of the ML-enabled system’s design, and support RE analysis and validation activities.

Similarly, (MAFFEY et al., 2023) proposed *MLTE*, an initial framework to evaluate ML models and systems that provides domain-specific language that teams, including model developers, software engineers, system owners, can use to express model requirements, an infrastructure to define, generate, and collect ML evaluation metrics, and the means to communicate results. While *MLTE* defines a general measurable process to evaluate ML systems, our proposal differs by going a step back and pointing out typical concerns involved when setting objectives and defining key components of ML-enabled systems. We see *MLTE* and *PerSpecML* as tools that can complement each other by supporting practitioners from different angles, since they share the same purpose of early addressing practical problems faced by multidisciplinary teams throughout the ML development process.

More recently, (AHMAD et al., 2023b) presented the *RE4HCAI* framework for specifying and modeling requirements for human-centered AI-based software that includes a catalog to elicit these requirements and a conceptual model to present them visually. The conception of *RE4HCAI* and *PerSpecML* follows the same principles, since the approaches provide a catalog and diagrams to support users, and both were based on literature findings and user feedback coming from empirical studies. While they share common goals such as modeling user, model, and data areas, they exhibit differences in their scopes. For instance, *RE4HCAI* lacks of an infrastructure area, vital for operating ML-enabled systems over time. In addition, *RE4HCAI* models few relationships between the attributes of different areas of the catalog when compared with *PerSpecML*, and don’t match such attributes with the stakeholders who should be in charge. We consider these features can support the collaboration and communication between stakeholders. In counterpart, *RE4HCAI* presents a more structured sequence for approach usage, which could be implemented by *PerSpecML* in the future.

From the industry perspective, several frameworks on human-centered AI development have been proposed by big tech companies, specifically Google’s PAIR guidebook (GOOGLE, 2021), Apple’s human interface guidelines for building ML applications (APPLE, 2020), and Microsoft’s eighteen guidelines for human-centered AI interaction (MICROSOFT, 2022). These frameworks delve deeply into critical elements such as user needs and defining success, data evaluation, explainability and trust, feedback and control, and handling errors. It’s worth noting that many of these elements align with the proposals outlined in *PerSpecML*. However, while these industry resources provide extensive documentation in the form of templates and worksheets, they often lack a comprehensive overview, potentially posing challenges in the application of their recommendations. A more holistic understanding of these resources, as offered by *PerSpecML*, may enhance their practical implementation in the development of ML-enabled systems.

2.8

Concluding Remarks

In conclusion, despite the important contributions in the field, there remains a relevant gap in the attention given to requirements specification and documentation in the context of ML-enabled systems. Existing efforts have primarily focused on a partial viewpoint of this type of system, proposing diverse quality properties and modeling techniques to enhance the design of ML-enabled systems. Nevertheless, we argue for a more comprehensive RE approach that goes beyond the current landscape. Our vision involves the development of a simple but comprehensive approach, accessible not only to requirements engineers but also to the broader spectrum of stakeholders engaged in ML projects. Such an approach could provide a cohesive overview of the multifaceted activities, requirements, and their relationships, providing to practitioners with the means to specify ML-enabled systems that align more closely with the expectations of diverse stakeholders typically involved in this domain.

3

A Systematic Mapping Study on RE for ML

3.1

Introduction

RE is a cornerstone discipline in the development of any software system. This should also apply for those that have an ML component (FRANCH; JEDLITSCHKA; MARTÍNEZ-FERNÁNDEZ, 2023). In traditional software systems, RE activities are well-established and researched. However, applying this RE knowledge to ML-enabled systems is more challenging due to the paradigm shift these systems represent (LWAKATARE et al., 2020). This added to the fact that existing literature has focused on using ML to support RE activities rather than on exploring how RE can improve the use of these technologies in the entire software development process (DALPIAZ; NIU, 2020), makes it necessary to investigate current RE approaches for ML-enabled systems, such as available frameworks, methodologies, tools, and techniques used to model requirements, and existing challenges and limitations.

In response to the importance and benefits that RE can offer to the development of ML-enabled systems, the first contribution of this thesis (in 2021) was synthesizing existing work on RE for ML. In particular, we conducted a systematic mapping study (SMS) in order to characterize RE contributions in terms of RE topics, quality characteristics, challenges, research directions, research type facets, and empirical evaluations (VILLAMIZAR; ESCOVEDO; KALINOWSKI, 2021).

We found that most of the RE contributions are in the form of approaches addressing different RE activities, quality models, analysis of specific quality characteristics of ML-enabled systems, taxonomy of challenges, checklists, and guidelines to support requirements engineers. We have observed that requirements elicitation is the most extensively researched activity within RE for ML, with numerous contributions in this area. On the other hand, requirements specification, modeling, and validation appear to be in need of further research and attention. We also identified specific quality characteristics, such as explainability, fairness, transparency and ethics, and RE challenges for ML, such as how to deal with stakeholder expectations, aligning data with the business goals, and how to properly cover and validate requirements.

3.2

Related Work

Back in 2021, there were no explicit literature reviews on RE for ML. We found two mapping studies that somehow relate to RE for ML. In (SCHUH et al., 2020), the authors conducted a literature review aiming at identifying design patterns, data model requirements, and technology potentials for ML systems in manufacturing companies. Their results in terms of RE focused on data attributes, such as quantity, quality, and dimensionality. Another study is the review of verification and validation for ML in the automotive industry conducted by (BORG et al., 2019). Here, the authors identified challenges such as transparency and requirements specification. We consider that the scope of these studies is significantly different from the literature review we conducted since it only partially addresses RE and limits its scope to one specific industrial sector.

As RE for ML gained attention from the research community in the past few years, previous studies on SE for ML also reported RE issues, challenges, and research directions. (KUMENO, 2019) and (LORENZONI et al., 2021) surveyed the literature in order to outline SE challenges that emerge during the development of ML-enabled systems. Regarding RE, they found that requirements activities mainly involve data and feasibility analysis, elicitation, specification, validation, and performance evaluation of ML models. On the other hand, (NASCIMENTO et al., 2020) conducted a literature review in order to investigate how SE has been applied in the development of ML-enabled systems, including challenges and practices. Their findings in terms of RE focused on practices performed by data scientists to improve the quality of data requirements, such as cross-validation and data distribution.

We acknowledge that pointing out these insights is important, however, we considered the coverage of these studies insufficient from the RE perspective since there are other interesting aspects to review in the literature, such as proposed scientific contributions and their evaluation. In summary, related work in 2021 reported a partial RE practices and challenges without covering a broader vision of RE for ML.

3.3

Protocol

A SMS is a study designed to provide a wide overview of a research area, to establish if research evidence exists on a topic and provide an indication of the quantity of the evidence. This SMS was performed following the guidelines

proposed by (KITCHENHAM; CHARTERS, 2007) and the specific guidelines by (PETERSEN; VAKKALANKA; KUZNIARZ, 2015).

3.3.1

Research Goal and Questions

The main research goal of this SMS is **to outline the state of the art of RE for ML-enabled systems**. The following research questions were derived from the objective in order to further characterize the RE contributions.

RQ1 *What RE contributions have emerged to support the software development of ML-enabled systems?*

This question aims at providing a general overview of RE contributions (*e.g.*, approaches, quality models, checklists) that have been proposed for ML.

RQ2 *What RE activities do the contributions address?*

The aim of this question is to identify the specific RE activities that were the focus of the contributions, helping to further understand their purpose.

RQ3 *What quality characteristics do the RE contributions consider for ML-enabled systems?*

There is a consensus in the SE community that the quality of ML-enabled systems must go beyond metrics such as accuracy and recall (KÄSTNER; KANG, 2020). This question aims at pointing out concerns about quality attributes for ML-enabled systems.

RQ4 *What are the reported challenges and research directions on the interplay between RE and ML-enabled systems?*

This question aims at identifying open challenges. One of the main reasons to conduct a SMS is supporting the planning of new research. Thus, this question seeks to indicate the aspects that may be studied by other researchers.

RQ5 *What are the research type facets of the contributions?*

The purpose of this question is to classify the papers according to their research type facets. We adopt the classification scheme proposed by (WIERINGA et al., 2006).

RQ6 *Which kind of empirical evaluations have been performed to assess the contributions?*

The purpose of this question is to identify what types of empirical studies have been conducted, focusing on the research type facets of evaluation and validation research from the previous question. Obtaining this information allows to get a first idea on the scientific rigour of the evidence reported in the field.

While research questions RQ1-RQ4 aim at structuring the publication landscape in a conceptual manner, the last two shall provide insights into the nature of the current reported evidence.

3.3.2 Search Strategy

The SMS employed a hybrid search strategy created by (MOURAO et al., 2020) that involves conducting a search string-based database search on a specific digital library (*Scopus*) and then complementing the set of identified papers with iterative backward and forward snowballing (using *Google Scholar*) following the guidelines proposed by (WOHLIN, 2014). We intentionally refrained from using various specific libraries, given that the chosen hybrid strategies typically achieve an appropriate balance of precision and recall. Between the different hybrid strategies (sequential, parallel, and iterative), we chose the more complete iterative snowballing for maximizing the recall, even though it would imply in analyzing more papers (WOHLIN et al., 2020). Iterative backward and forward snowballing concerns applying backward and forward snowballing on each new included paper.

We chose *Scopus* because it claims to be the largest database of titles and abstracts (KITCHENHAM; CHARTERS, 2007), which would allow us to identify a representative and unbiased seed set. It is, however, backward and forward snowballing via *Google Scholar* which we used as an effective way to complement the identification of the broader population of studies (WOHLIN, 2014; MOURAO et al., 2020).

We formulated the search string to conduct the initial database search on *Scopus* using the PICO (*Population, Intervention, Comparison, Outcome*) criteria (LEONARDO, 2018) strategy. Our study focuses on ML-enabled systems (*population*) and aims at identifying RE contributions for such systems (*intervention*). As our study concerns a SMS, there was no specific *comparison* nor the need of limiting the search space regarding *outcomes*. Therefore, we needed keywords for ML and RE. The defined search string, to be applied on titles, abstracts and keywords was:

“(Software OR Applications OR Systems) AND (Machine Learning) AND (Requirements Engineering)”

3.3.3 Study Selection

Following suggestions by (MENDES et al., 2020), we initially defined a well limited intended time-frame for our SMS, comprising papers published by the end of 2020. The primary inclusion criteria was on papers that describe RE contributions in the context of ML. When several papers reported the same study, only the most recent one was included. When multiple studies were reported in the same paper, each study was considered separately. The exclusion criteria applied for filtering the papers are shown in Table 3.1.

Table 3.1: Exclusion criteria.

Criteria	Description
EC1	Papers that do not meet the inclusion criteria
EC2	Papers about the use of ML techniques for improving RE activities
EC3	Papers not written in English
EC4	Grey literature, including blogs, white papers, theses, and papers that were not peer reviewed
EC5	Papers that are only available in the form of abstracts/posters and presentations

Figure 3.1 shows all the steps performed in the paper selection process. The database search results on Scopus, filters, and backward and forward snowballing are detailed below.

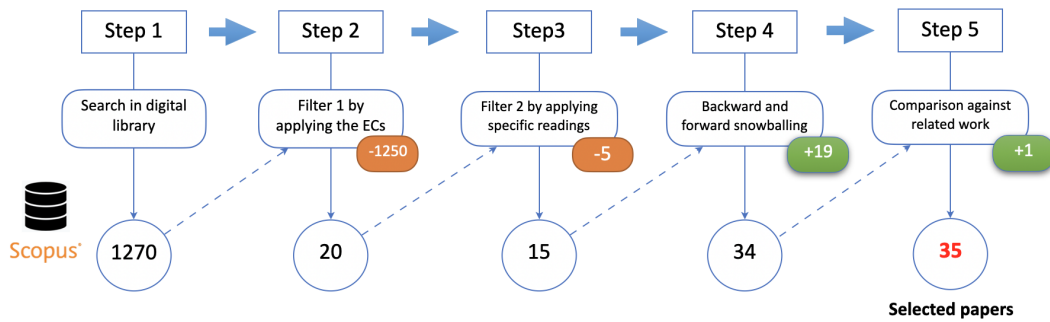


Figure 3.1: Papers selection process.

The first step consisted of searching for papers using the search string in the digital library selected for this study. The search string was applied on titles, abstracts and keywords in *Scopus* in January 2021, and returned 1270 papers. In the second step (Filter 1), the first filtering took place. In this step, we applied the exclusion criteria. Regarding *EC1*, at this step we excluded the papers that clearly didn't have information on RE for ML-enabled systems. We identified that a substantial number of papers (175) concern *EC2*. This

confirms that the intersection between RE and ML is predominant for papers that use ML to support RE activities. As a result, we reduced our set of candidate papers to 20.

In the third step, we applied a second filter (Filter 2), filtering papers by reading the titles, abstracts and selected paper parts (when necessary) while applying the inclusion and exclusion criteria. This step left us with 15 papers representing the result of the search on Scopus. All exclusions and the final set of included papers were peer reviewed by an independent researcher. In case of divergence, a third researcher was involved and a discussion was held to reach consensus.

In the next step, during the month of February, we applied backward and forward snowballing iteratively following the snowballing guidelines in (WOHLIN, 2014). In total, four backward snowballing (BS) and two forward snowballing (FS) iterations were applied (order: BS1, BS2, FS1, BS3, BS4, FS2) until reaching our final set of papers. The four backward snowballing iterations involved analyzing 624 ($259 + 200 + 131 + 34$) papers (including duplicates) and allowed identifying twelve additional papers to be included. The two forward snowballing iterations involved analyzing 304 ($290 + 14$) papers (including duplicates) and allowed identifying seven additional papers to be included. The whole snowballing process was peer reviewed. It is noteworthy that the first forward snowballing iteration retrieved a paper accepted for publication in 2020, but published January 1st 2021 (NALCHIGAR; YU; KESHAVJEE, 2021). As this paper was on the borderline of our scoped time frame, but represents a valuable contribution, we decided to also include it in our mapping. This explains the single paper from 2021.

Finally, in the fifth step, we compared our results against the results provided by the related work (*cf.* Section 3.2). We found only one paper (BARASH et al., 2019) that was not identified by our search strategy, because this paper didn't cite any of the remaining studies on the topic. Hence, in total, 35 papers were included in the SMS, where 15 papers came from *Scopus*, 19 papers came from snowballing and one paper came from analyzing related work as shown in Fig. 3.1. The selected papers are shown in Section 3.4. A spreadsheet with all details on the filtering and snowballing process, documenting each iteration, can be found in our online open science repository ¹.

¹<https://doi.org/10.5281/zenodo.4682374>

3.3.4

Data Extraction and Classification Scheme

The information extracted from each of the selected papers and the classification schemes describing the different categories are presented in Table 3.2. The complete extracted data is available in our online repository.

Table 3.2: Data extraction form.

Information	Description
Study metadata	Includes the paper title and information such as venue, type of venue and year of publication.
RE contribution	Description of the RE contribution for ML.
RE topics	RE topics that the contribution addresses. These topics were coded based on typical RE activities (<i>e.g.</i> , elicitation, analysis, modeling, specification, validation, verification, and management) or other RE aspects (<i>e.g.</i> , data quality requirements, requirements assurance) that were the focus of the contributions.
Quality characteristic	Characteristic that influences the quality of ML-enabled systems, also known as NFRs (<i>e.g.</i> , safety, explainability, performance).
RE problems	Challenge/issue that arises during the RE process.
Research directions	RE topic suggested by contributions.
Research type facet	Classification of research type facets according to (WIERINGA et al., 2006), including the following categories: evaluation research, solution proposal, philosophical paper, opinion paper, or experience paper.
Empirical evaluation	Classification of the empirical strategy, according to (WOHLIN et al., 2012), including the following categories: experiment, case study, survey.

3.4

Results

This section presents the results of the SMS. First, we provide an overview of the included papers. Overall, we identified 35 papers. Regarding the years of publication, the papers range from 2018 to 2021. Most of the publications (31) are conference and workshop papers and only 4 papers have been published in journals. The venues in which the topic has been addressed comprise premier international SE conferences and journals such as *FSE*, *ICSE*, *RE*, *ESEM*, *REJ*, and *TSE*. This gives an idea of the relevance and interest on this topic on behalf of the SE community in the last years.

3.4.1

RQ1 What RE contributions have emerged to support the development of ML-enabled systems?

Similar to our previous SMS in the field of RE (VILLAMIZAR et al., 2018), we followed open coding guidelines (SALDAÑA, 2021) with the aim of characterizing the papers by the type of contribution. We coded the following different main contribution types for the papers: analyses (*e.g.*, analyzing some RE aspects for ML), approaches (*e.g.*, methods, methodologies, processes, and conceptual frameworks), checklists and guidelines (C & G), quality models (QM), and taxonomies (T). Table 3.3 shows an overview of these contributions by contribution type, and Appendix A outlines them.

3.4.2

RQ2 Which RE activities do the contributions address?

The majority of the selected papers concern requirements elicitation practices (14 out of 35), where authors consider problems such as defining business goals and problems of understanding. Furthermore, we found five papers about requirements analysis, more specifically addressing customer expectations and requirements prioritization. We also identified contributions regarding data related requirements in five papers. Other contributions are focused on requirements specification, assurance, modeling, verification, and validation. Figure 3.2 shows the distribution of the papers by the covered RE activities.

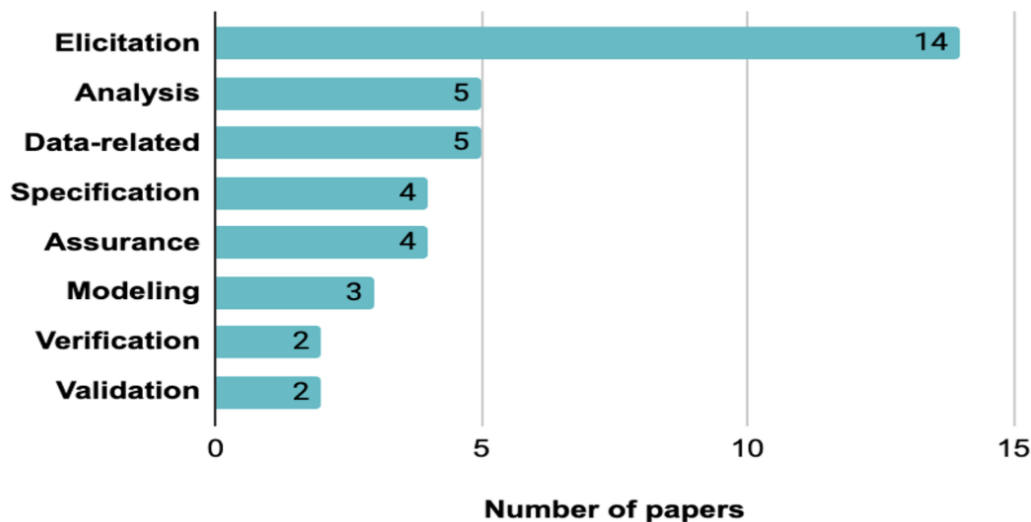


Figure 3.2: Distribution of the papers per RE activity.

Table 3.3: Identified contributions from the literature review.

Type	Id	Description
Analyses	P3	Teaching Software Engineering for AI-enabled systems
	P4	Emerging and changing tasks for developing ML systems
	P11	Perspectives from data scientists
	P14	Challenges and new directions of NFRs for ML
	P16	How to adapt SQuaRE for AI systems
	P17	How engineers perceive difficulties in engineering ML systems
	P20	How does ML change software development practices?
	P21	Studying SE patterns for designing ML systems
	P24	Approaches for safety requirements assurance
	P25	The importance of quality requirements for deep learning
	P28	ML challenges of safety-critical cyber-physical systems
	P33	Using conceptual modeling to support ML
	P34	NFRs for human-centered software
Approaches	P2	Approach for evidence-driven RE to handle uncertainty in ML
	P6	Conceptual framework for ML model lifecycle management
	P8	Approach to software metrics for ML systems
	P9	Method for identifying stakeholders needs
	P12	Approach for specifying ML systems
	P15	Methodology to guide the development of ML systems
	P18	Approach for specifying ML systems based on human aspects
	P22	Method for understanding XAI requirements in ML systems
	P23	Method for explainability in AI systems
	P26	Method for dataset augmentation to improve neural networks
	P27	Conceptual framework for eliciting and designing ML systems
	P29	Method for eliciting security requirements in ML systems
	P31	Methodology for the evaluation of NFRs in ML systems
	P32	Process for developing data-driven applications
	P35	Method for bridging the gap between ML and business goals
C & G	P7	Guidelines for quality assurance of ML systems
	P13	Checklist to support business modeling
	P19	Best ML and SE practices as requirements
	P30	Ethical guidelines for developing AI systems
QM	P1	Determining quality characteristics and measurements for ML
	P7	Guidelines for quality assurance of ML systems
T	P5	Engineering problems in ML systems
	P10	RE Challenges in Building AI-Based Complex Systems

3.4.3

RQ3 What quality characteristics do the RE contributions consider for ML-enabled systems?

During the analysis of the contributions, it was possible to identify several quality requirements that authors consider in their research. Table 3.4 shows these quality characteristics that were considered in the papers with their frequencies. Note that one paper can address one or more quality characteristics.

Table 3.4: Frequency of quality characteristics from the literature review.

Characteristic	Frequency	Characteristic	Frequency
Security	6	Testability	2
Explainability	6	Accountability	2
Privacy	6	Ethics	2
Data quality	5	Accuracy	2
Fairness	5	Suitability	1
Transparency	5	Uncertainty	1
Reliability	4	Autonomy	1
Safety	4	Robustness	1
Performance	3	Modularity	1
Maintainability	3	Scalability	1
Legal requirements	2	Usability	1

3.4.4

RQ4 What are the reported challenges and research directions on the interplay between RE and ML-enabled systems?

Some papers explicitly report challenges from the point of view of RE when developing ML-enabled systems. We grouped them in order to provide a better understanding and then outline the challenges. An overview is summarized below.

Lack of validated techniques. Developing ML-components mainly relies on applying techniques to achieve an objective. However, there seems to be a lack of validated techniques for some important aspects of RE for ML. For instance, several studies (*e.g.*, [P5] [P8] [P12] [P13] [P14]) state that ML researchers and users currently lack an ML-specific way to express and specify requirements for ML, including targets and trade-offs, and the influence of domain context. Other studies, such as [P1] [P3] [P5], mention that measuring quality beyond traditional metrics, such as accuracy and precision, may be complicated since identifying quality attributes is often difficult. The authors of [P13] also outline that identifying business metrics is not trivial since customers want to have policies to improve their business, but they don't understand

what metrics and data are required to do so. This represents a challenge for requirements engineers of ML-enabled systems. In addition, in [P28] [P35] the authors raise issues on how to properly cover and validate requirements for ML systems and how to deal with testing and verification activities.

Knowledge regarding NFRs. In [P14] the authors state that the understanding of NFRs for ML is fragmented and incomplete, including how to define and refine NFRs in ML-specific contexts. Quality attributes such as explainability ([P22] [P23] [P33]), safety ([P3]), security ([P18]), fairness ([P3]), robustness ([P5]), and transparency ([P19]) are pointed out as challenging by researchers.

Handling customer expectations. Organizations did not realize that ML models are mainly probabilistic models that commonly have to learn patterns from messy data. This reflects difficulties customers have to understand potential limitations of ML-enabled systems. Papers such as [P7] [P8] [P13] [P17] reveal that customers commonly expect to see magic coming out of data.

Furthermore, we wanted to know what research directions are encouraged by the authors. After analyzing the papers, we identified that the authors are mainly asking the community to conduct more empirical studies to uncover more insights on best practices and to propose and investigate approaches that are suitable to be used in practice. The authors also mention other research directions, such as:

- Address transparency, explainability and safety for ML.
- Develop tools to support requirements specification.
- Create requirements guidelines for ML-enabled system.
- How to address ethics, security, and privacy in ML context.
- How to verify ML requirements and validate ML models.
- Extend the ML quality characteristics.
- Develop standard quality models for ML-enabled systems.
- Survey ML literature and/or ML experts on NFRs.
- Understand how ML models can be integrated into a larger system
- Provide guidance to non-technical stakeholders about what is possible and what is not.

3.4.5

RQ5 What are the research type facets of the contributions?

Figure 3.3 shows the distribution of the research type facets of the papers per year. It is possible to observe that most of the papers (16 out of 35) concern evaluation research papers. In the next question we address the types of empirical evaluations that were conducted. Opinion papers, with ten studies, significantly contribute to the account. We also identified that solution proposals are still scarce in this field. This contrasts the identified lack of techniques and the absence of tools supporting RE activities for ML-enabled systems, further motivating research directions pointed out by the authors.

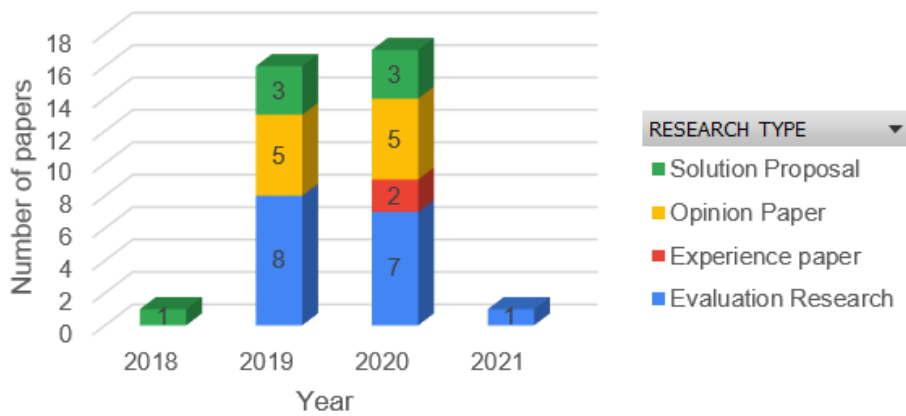


Figure 3.3: Distribution of research type per year.

3.4.6

RQ6 Which kind of empirical evaluations have been performed?

When analyzing the empirical evaluations conducted within the studies (see Figure 3.4), it was possible to identify 16 papers out of 35 that have performed empirical evaluations (twelve case studies, three surveys and one experiment). Note that five papers provided a proof of concept, *i.e.*, a realization of a certain method or idea in order to demonstrate its feasibility. This is not considered as an empirical evaluation by (WOHLIN et al., 2012), therefore they were not classified as evaluation research. In fact, 14 studies did not contain any type of empirical evaluation or even a proof of concept. Most of these concern opinion and experience papers. The most applied empirical evaluation strategy in the analyzed studies was case study (12 papers) in contrast with survey (three papers) and experiment (one paper).

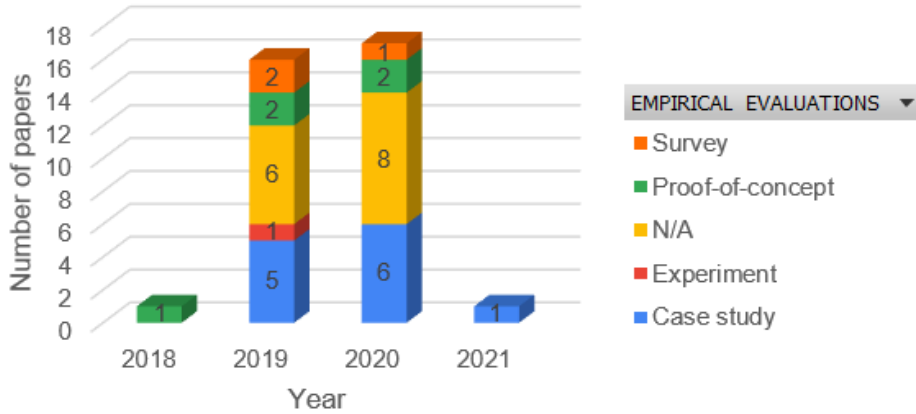


Figure 3.4: Distribution of empirical evaluation type per year.

3.5 Discussion

The landscape of RE for ML has undergone a significant transformation since the initiation of this research in 2020. At that time, the SMS presented in this chapter was carried out, which, to the best of our knowledge, was the first of its kind to offer a comprehensive overview of this emerging field. In this time period, we have witnessed a surge in interest and exploration in this area, resulting in the publication of several other literature reviews by both the RE and SE communities (AHMAD et al., 2021; GIRAY, 2021; PEI et al., 2022; MARTÍNEZ-FERNÁNDEZ et al., 2022; AHMAD et al., 2023a; NAHAR et al., 2023). These contributions show the growing importance of specifying, managing, and understanding requirements for ML-enabled systems.

The existence of subsequent literature reviews further underscores the relevance of this topic, reflecting the recognized need for a deeper understanding of the challenges and solutions within the scope of requirements in ML projects. As the discourse around RE for ML continues to expand, these additional reviews provide valuable perspectives and insights, contributing to a more holistic and nuanced understanding of the field. This context of evolving research and the emergence of new perspectives highlights the relevance of the current work within the evolving landscape of RE for ML, building upon the foundations set by earlier reviews and offering new insights and approaches that can enrich the discourse further.

Most of these new findings are cited at any point in this thesis where an observation or contribution made by the work is mentioned.

3.6

Threats to Validity

Internal validity. We used a hybrid search strategy combining a database search on a single database (*Scopus*) with iterative backward and forward snowballing (using *Google Scholar*), and precisely documented each step. One could argue that the search string was confined to a small set of keywords. These keywords were objectively selected using the PICO strategy and are directly related to our research goal. It is important to remember that the database search was used to reveal an unbiased and representative seed set, as starting point for iterative forward and backward snowballing, and that this search strategy has been effective for secondary studies (MOURAO et al., 2020).

External validity. We systematically applied a search strategy that has shown good results regarding recall (MOURAO et al., 2020) and validated it by comparing it against related work. Still, there is a possibility of having missed studies. Nevertheless, we were unable to manually find any additional study to be included and are confident that we have an unbiased and representative sample. The claims made in our paper are related to the findings reported in the primary studies. While all of them were peer reviewed, and many of them were published in venues that have a rigorous selection process, we did not assess their quality. Such quality assessment is typically not part of mapping studies, and could be part of a systematic review extension. The complete information concerning the process, the extracted data and coding is available in our online repository and is publicly auditable.

Reliability. In order to reduce the bias when selecting relevant studies, it was decided to examine the selected papers in pairs. Hence, two researchers evaluated the selected studies, extracted data and coding in a peer-reviewed manner.

3.7

Concluding Remarks

The literature review conducted as part of this research makes for one of the initial contributions to this thesis. It served as the foundation upon which the proposed catalog and *PerSpecML* was developed. The SMS presented in this chapter was paramount in not only identifying the gaps and shortcomings in the current landscape of RE for ML but also in understanding the challenges faced by practitioners in this context. By synthesizing the literature, we were able to pinpoint critical areas where improvements were needed. The understanding acquired from the review was essential in designing and evaluating

a solution that addresses the identified issues, bringing empirical rigor to the field. In sum, the literature review laid the groundwork for the subsequent work in this thesis, highlighting its crucial role in the inception of *PerSpecML* and, consequently, its significance to the overall contributions of this research.

To perform the SMS on RE for ML, we applied a hybrid search strategy, complementing a database search on *Scopus* with iterative backward and forward snowballing. Our search strategy allowed identifying a total of 35 studies. We identified several proposed research contributions, some published in premier SE conferences and journals. These contributions comprise analyses, approaches, checklists and guidelines, quality models, and taxonomies. We identified research gaps by relating these contributions to RE investigation topics. We also highlighted quality characteristics considered within the papers and reported on challenges and potentially promising research directions.

The main contributions of this SMS are twofold: (i) mapping relevant knowledge about the current state of RE for ML, a subject that is still not widely explored by researchers and confused by practitioners; and (ii) helping to identify points that still require further investigation. This SMS was the first literature review that organized evidence to provide a comprehensive overview of contributions related to RE for developing ML-enabled systems.

4

A Catalog of Concerns for ML-Enabled Systems

4.1

Introduction

The impact of incomplete requirements on overall system development is considered one of the most critical problems of RE in practice (FERNÁNDEZ et al., 2017). Identifying and specifying requirements for ML-enabled systems is even more challenging, where practitioners, including requirements engineers, are typically not aware of the wide range of requirements that might apply to such systems. On the other hand, many of these systems focus on technical aspects such ML performance metrics, model development and deployment, and ignore important business, user, and human aspects (AHMAD et al., 2023b).

Recent research papers have studied the understanding, challenges, and use of quality attributes (also known as NFRs) among practitioners (HORKOFF, 2019; HABIBULLAH; GAY; HORKOFF, 2023) and others have drawn the attention of researchers and practitioners on the fact that RE topics such as identifying quality attributes, specifying them, and understanding how they can be analyzed are not well-established and investigated in the context of ML (CYSNEIROS; LEITE, 2020; HEYN et al., 2021; AHMAD et al., 2023a).

In order to help address the issues presented in current RE for ML research, in this chapter, we present a catalog of concerns to be used by practitioners of ML projects to support the identification and specification of requirements (VILLAMIZAR; KALINOWSKI; LOPES, 2022). The catalog is organized into five different perspectives: objectives, user experience, infrastructure, model, and data, and was based on the SMS presented in Chapter 3, on our own experience and those of other authors with the development of ML-enabled systems (HULTEN, 2019; KALINOWSKI et al., 2020). We conducted a focus group session with eight software professionals experienced in developing such systems to validate the catalog. The results revealed that the professionals were not explicitly aware of many of the concerns but that they recognized their relevance and potential impact on the overall system being developed. In general, they agreed with the concerns and the way of grouping them into perspectives. In addition, we received relevant feedback that we used to improve our catalog.

4.2 Background

4.2.1 Concerns in SE

In SE, a concern typically refers to a specific aspect, interest, or issue that needs to be addressed or considered during the development and maintenance of a software system, consequently influencing its design, implementation and behavior. When designing ML-enabled systems and breaking them down into components, it is crucial to identify which attributes or characteristics are important to contribute to the overall system's quality. Determining this requires a deep understanding of the system's goals, stakeholders' requirements, and the overall context in which the software will be used. In the case of ML-enabled systems, the challenge is further amplified since it incorporates ML models that make predictions based on patterns and trends learned from data, which introduce unique considerations. In this thesis, we refer to *concerns* as the aspects and issues related to ML and non-ML components, which can be addressed during the specification and design of ML-enabled systems.

4.2.2 Perspectives in SE

In SE, a perspective refers to a representation of a system or its components. It provides a focused way of analyzing a particular aspect of the system, allowing to capture different concerns and stakeholders' viewpoints. Perspectives have been effectively used in SE to model scenarios where team members work on a particular phenomena (BASILI; ROMBACH, 1988).

4.3 Methodology

To conceive the catalog, we used the constructionism theory that advocates a person needs to understand how something works before exploring the different ways to construct solutions (FOSNOT, 2013). Figure 4.1 illustrates what we did and how we created, validated and improved our catalog to support the specification of ML-enabled systems.

Throughout this process, we first understood how ML works in practice and how RE could be used to support the overall development of ML-enabled systems (step 1). For this, we had prior practical experience with real ML projects in a R&D initiative called *ExACTa*¹. These projects involved several

¹<http://www.exacta.inf.puc-rio.br>

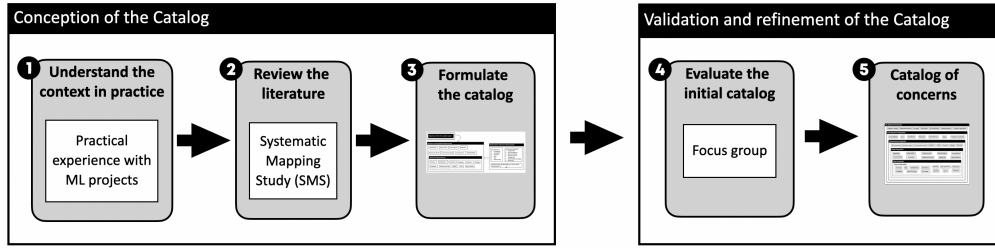


Figure 4.1: Overview of the study steps for defining the catalog.

deliveries of solutions involving different types of ML problems and algorithms (*e.g.*, decision trees, logistic regression, neural networks). In experiential learning, this is defined as learning through reflection on doing. Additionally, we took advice from an industry-oriented publication based on more than a decade of experience building systems with ML components (HULTEN, 2019). With this, we seek to align the knowledge and insights acquired up to this point. The next activity (step 2) involved conducting a literature review on how ML could benefit from the RE perspective and what research opportunities could be addressed (See Chapter 3. We took advantage of these planned synergies between literature and practice to understand the current use and challenges among practitioners in the context of RE for ML.

After analyzing the literature and the ML-enabled system development context in practice, we created an initial catalog of concerns (step 3). It is noteworthy that the catalog was critically reviewed by two active researchers in the areas of SE and data science and that have been exploring the intersection between these two areas. The literature review led us to focus on requirements definition, since this was one of the main identified research challenges, and revealed several quality properties of ML-enabled systems. Our industrial experiences allowed us to validate our findings and revealed complementary perspectives and concerns to be considered for ML-enabled systems. Finally, we conducted a focus group session (step 4) with eight software professionals with large experience developing ML-enabled systems. The results of the focus group allowed us to improve the initial catalog of concerns (step 5).

4.4 The Catalog

The specification of ML-enabled systems involves concerns that are often not easily identified, resulting in "hidden" requirements. For instance, it is clear that a model needs good data to be trained and then evaluated, but it is not clear what are the criteria that define the data as good, nor defining the frequency and forcefulness of the ML model to get better user

experiences, among others. Given this, we proposed a catalog of 45 concerns for supporting the specification and design of ML-enabled systems that models five perspectives: objectives, user experience, infrastructure, model, and data. This catalog accommodates the findings of our literature review and that showed being relevant in practice, bringing a big picture of the ML workflow. Figure 4.2 shows the concerns grouped by perspective.

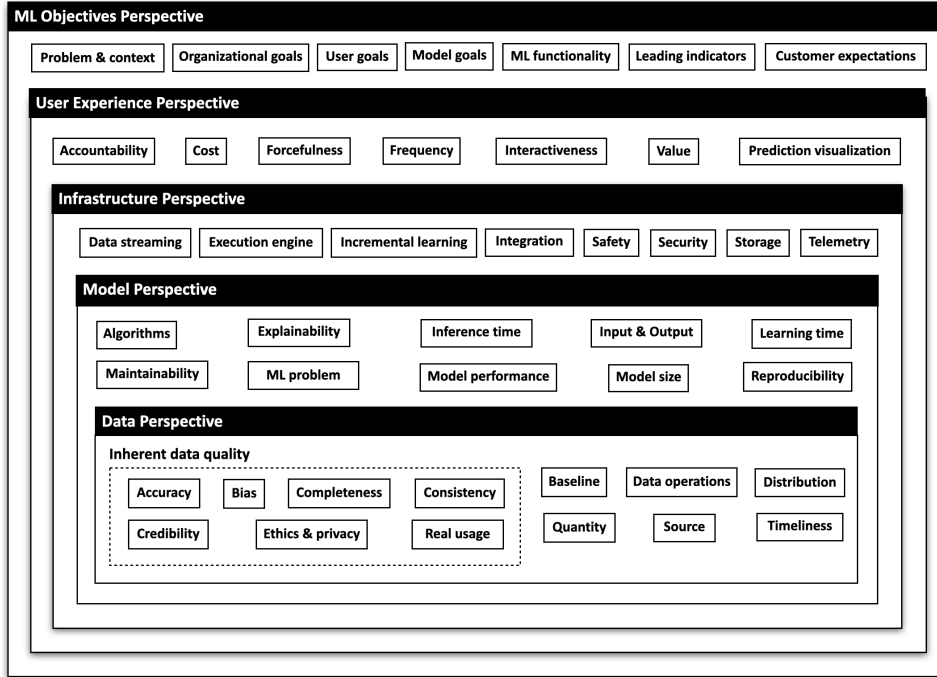


Figure 4.2: An overview of the catalog of concerns to support the specification and design of ML-enabled systems (VILLAMIZAR; KALINOWSKI; LOPES, 2022).

The depicted version already considers the adjustments made based on the focus group feedback. We seek this catalog can be used by requirements engineers to support the specification and design of ML-enabled systems, making them aware of the big picture and helping to avoid incomplete requirements. We suggest the concerns to be analyzed by requirements engineers and discussed with stakeholders to understand the degree to which related requirements should be met. In Chapter 5, we describe the perspectives and detail their concerns as part of *PerSpecML*, our proposed solution that incorporates this catalog.

4.5 Focus Group

In this section, we present the focus group we conducted to evaluate the catalog of concerns and gain feedback which contributed to improve it.

4.5.1

Research Questions

In order to evaluate the catalog of concerns, we defined the following research question:

Is the catalog of concerns promising and could it support the requirements specification and design of ML-enabled systems?

To answer this question, we evaluate this work from three angles.

First, the **perception of importance** to know if the catalog of concerns is addressing a relevant problem. Second, the **perception of quality** to establish to what extent the catalog of concerns is complete, consistent and correct. By last, the **perception of feasibility** to have an idea to what extent the catalog of concerns can be applied in practice.

For this purpose, we designed a focus group session for promoting in-depth discussion about the catalog and its suitability. Focus group is a qualitative research method based on gathering data through the conduction of group interviews and it has been conducted in SE for revealing arguments and feedback from practitioners (KONTIO; LEHTOLA; BRAGGE, 2004).

4.5.2

Participants

We invited eight practitioners who have been actively working with the development of ML-enabled systems at the *ExACTa* initiative. Before conducting the focus group session, we applied a characterization form. We asked them about the position they have within the initiative, and their experience in years and number of ML projects they participated in. Table 4.1 shows an overview of the participants.

Table 4.1: Overview of the participants of the focus group who evaluated the catalog of concerns.

Id	Position	# years	# ML projects
P1	Data scientist	13	12
P2	Data scientist	9	7
P3	Data scientist	1	3
P4	Developer	1	1
P5	Developer	3	2
P6	Project lead	1	2
P7	Project lead	2	5
P8	Project lead	2	2

4.5.3

Execution

Before starting the focus group, we introduced to the participants the main challenges when developing ML-enabled systems and presented how RE may address some of them. The focus group was conducted in a Zoom meeting recorded for the study. The study was planned to be executed in two phases. In the first phase that took 20 minutes, we explained the catalog of concerns by decomposing each requirement perspective in its related concerns. The second phase, that took 45 minutes, was a question & answer session about the importance, quality and feasibility of the catalog of concerns. The focus group was moderated and recorded by the main author of this thesis.

4.5.4

Results

Perception of importance. We asked the participants how they define, document and organize requirements for ML-enabled systems and if they think it is important. P7 stressed the lack of formal methods to support their definition, modeling and documentation: “I have constant difficulties to find tools and methods to help my team and customers understand requirements in ML projects”. On the other hand, P2 stated: “In my opinion, the requirements process for ML is ad-hoc, which makes it highly dependent on people’s knowledge” and P3 manifested: “I noticed that requirements have constant rework in ML projects”. Considering the overall discussion with the participants, we understood that creating new methods in this direction is important to address the problems practitioners are facing.

Perception of quality. The participants evaluated the catalog by (i) analyzing the concerns and perspectives in terms of completeness, consistency and correctness, and by (ii) measuring the capacity of the catalog to support the requirements specification in ML projects. Overall, there was a clear consensus that practitioners are unaware of the big picture. They did not know about many of the concerns, while judging them as relevant and helpful to support more precise specifications. P1 mentioned that “I wish I had such concerns specified upfront in my ML projects. Decisions regarding these concerns should not be taken without appropriately involving stakeholders or when coding”. When analyzing the perspectives, P2 emphasized the importance of the ML objective perspective: “I understand the need to consider data, model, user and infrastructure, but in my opinion, the functional behaviour of ML models, which is reflected in the objectives, is crucial”. Regarding the concerns, P5 stated that from the technical view, the concerns and their grouping make

sense: “When seeing the concerns I was able to relate them to problems and tasks I faced in the past”. P1 manifested the importance to evaluate in depth the completeness of the concerns. For instance: “In the data perspective, a common concern is the definition of a baseline that helps in the acquisition of new data. I would definitely consider it”. We improved the catalog based on the session feedback.

Perception of feasibility. The participants found the catalog of concerns useful to support the requirements specification and design of ML-enabled systems. P6 stressed the concerns organization into perspectives: “I think the catalog can help us to analyze requirements in our ML projects since it covers several perspectives for different situations”. In addition, P8 stated: “We need to use this type of proposals in practice due to the overview that it provides and the concerns that may apply in our context”. Given this early feedback, we believe that it is feasible to further evaluate the catalog.

4.6

Discussion

Developing ML-enabled systems involves, at least, a set of skills of three areas: operations, data science and SE (LEWIS; BELLOMO; OZKAYA, 2021). Our perception, based on practical experiences, is that many companies have data engineers writing REST APIs, data scientists building pipelines, and software engineers building ML models. This reflects one of the main issues in the development of such systems, leading to extra efforts and low software quality. We have also seen that practitioners of ML projects often scribble a few Jupyter Notebooks to build and evaluate ML models where code quality is bad. They run experiments and the generated artifacts are saved to folders named in mysterious ways, randomly spread across the filesystem. In addition, documentation is often missing. As a consequence, the implementation is difficult to understand. Efforts to address these challenges may include creating approaches to:

- Identify and specifying requirements for ML projects.
- Provide a more holistic view of the ML development process by modeling activities, stakeholders, and their relationships.
- Encourage stakeholders to collaborate closely.

We believe that the catalog of concerns we proposed is a resource that can help, as a first step, to address several of the RE for ML challenges identified both in literature and industry.

We are aware that not every ML-enabled system needs to address all the concerns we proposed and not every ML-enabled system needs to implement them to the same degree. Our intention is to provide a resource so that requirements engineers can analyze, together with stakeholders, the needs of their ML-enabled systems. It is noteworthy that the catalog focuses on concerns for both ML and non-ML components, that together make up a larger system. However, when considering the overall system, general quality characteristics of software products such the ones mentioned in the ISO/IEC 25010 standard (ISO/IEC, 2011), should also be analyzed.

From the point of view of practical benefits when using the catalog, we believe that the set of concerns grouped by perspectives may eventually be useful in various situations. First, to validate an already specified system. In this case, our concerns would be a reference since they come from a literature review and different industrial experiences on building ML-enabled systems. Second, the catalog may help to understand the communication between several services involved in ML projects (*e.g.*, data ingestion and ML models), since it highlights functional and non-functional aspects at different levels.

4.7

Concluding Remarks

The development of ML-enabled systems involves understanding how ML can add value to business objectives, translating them into ML tasks, designing and experimenting with ML algorithms, evaluating ML models, designing pipelines, among other tasks. This needs to be considered from early stages of ML software development. Based on the literature and on practical experiences, we proposed a catalog of 45 concerns to support the specification and design of ML-enabled systems covering five perspectives: objectives, user experience, infrastructure, model, and data. This is the first effort aiming at providing the big picture of the concerns involved in the development of such systems. With this catalog, we seek to empower requirements engineers with an ML overview so that they can analyze the concerns with business owners, data scientists, software engineers and designers.

We evaluated the catalog by conducting a focus group session with eight ML practitioners involved in the development of ML-enabled systems. The purpose was to gain insights about the relevance of the problem we are addressing, the benefits of using it and its feasibility. The results indicated that practitioners consider the identified concerns relevant and the catalog useful. They stated that grouping perspectives and organizing concerns can help them to identify constraints upfront with other practitioners. Therefore,

we believe that the conceptual perspectives and concerns we herein proposed can be helpful to support the specification and design of ML-enabled systems.

5

An Approach for Identifying Concerns When Specifying ML-Enabled Systems

5.1

Introduction

Requirements can be hard to specify for ML-enabled systems due to the issues related to measuring and defining requirements for non-deterministic systems (MARTÍNEZ-FERNÁNDEZ et al., 2022). Also, the emergence of new requirements such as data, ethics and explainability has posed issues to requirements specifications. Furthermore, we also found limited studies that focused on identifying and specifying requirements for ML-enabled systems. Given this, several works recommend that researchers should construct a reference map to document requirements in this context, allowing to capture key components and attributes needed when specifying the requirements for ML-enabled systems (VILLAMIZAR; ESCOVEDO; KALINOWSKI, 2021; AHMAD et al., 2021).

In order to help addressing these issues and recommendations, in this chapter, we present *PerSpecML*, an approach for identifying concerns when specifying ML-enabled systems that involves analyzing 60 concerns related to 28 tasks that practitioners typically face in ML projects, grouping them into five perspectives: system objectives, user experience, infrastructure, model, and data. Together, these perspectives serve to mediate the communication between business owners, domain experts, designers, software and ML engineers, and data scientists.

We created *PerSpecML* by following a technology transfer model proposed by (GORSCHEK et al., 2006), which is recommended to foster successful transfer of technology from research to practice (WOHLIN et al., 2012). Throughout this process, we participated in real ML projects of the *ExACTa* initiative, conducted a literature review on RE for ML presented in Chapter 3, formulated a catalog with an initial set of concerns presented in Chapter 4, and proposed a candidate solution for specifying ML-enabled systems. We iteratively evaluate and improve the catalog and the candidate solution by conducting three studies in different contexts: (i) in an academic validation involving two courses on SE for data science, (ii) with practitioners working with ML-enabled systems in a R&D initiative, and (iii) in two real industrial case studies conducted with a Brazilian large e-commerce company.

The iterative validations and continuous improvements result in *PerSpecML*, our approach for identifying concerns when specifying ML-enabled systems, and collectively corroborated its potential as a comprehensive tool for guiding practitioners in collaboratively designing ML-enabled systems, enhancing their clarity, exploring trade-offs between conflicting requirements, uncovering overlooked requirements, and improving decision-making. Furthermore, we found that the participants involved in the validations gradually improved their perception of *PerSpecML*'s ease of use, usefulness, and intended to use.

5.2 Methodology

In this section, we describe the process we followed to design and evaluate *PerSpecML* based on the technology transfer model introduced by (GORSCHEK et al., 2006). We used this model since our research method involved evaluations in both academia and industry with the aim of scaling the proposal up to practice, for which this model is recommended (WOHLIN et al., 2012). This mix of evaluations provides an opportunity to gather user feedback and incorporate it into the solution design. By involving stakeholders and practitioners in the evaluation process, we gathered valuable insights about their experience, needs, and preferences. This feedback informed iterations and refinements of the solution, making it more user-centric and aligned with actual user requirements. In the following, we detail the seven steps of the transfer model.

Step 1: Identify improvement areas based on industry needs.

During the last four years, the author of this thesis participated in R&D projects designing and developing ML-enabled systems. These projects involve different types of ML tasks (*e.g.*, supervised and unsupervised learning, computer vision) and algorithms (*e.g.*, decision trees, logistic regression, neural networks). This experience allowed us to assess current practices, observing domain and business settings, understand typical industry needs for ML-enabled systems, and issues related to their development. More specifically, we identified

- a) How important the domain and business settings are to align the stakeholder needs, requirements, and constraints with the engineering and data science activities.
- b) Interdisciplinary teams typically involved in ML projects.

- c) The lack of tools and documents that can capture key components when specifying ML-enabled systems.

Step 2: Formulate a research agenda. In order to better define the problem and gain more insights into existing solutions and what needs to be created, we conducted a SMS on RE for ML detailed in Chapter 3, analyzed later literature reviews (AHMAD et al., 2021; PEI et al., 2022; AHMAD et al., 2023a) and took advice from an industry-oriented publication based on more than a decade of experience in engineering ML-enabled systems (HULTEN, 2019). Here, we identified, for instance:

- a) Additional quality attributes of ML-enabled systems that practitioners should analyze.
- b) The lack of studies focused on identifying key components of ML-enabled systems that may later be specified.
- c) the lack of studies evaluated in practice to validate its effectiveness, feasibility and gather user feedback.

Step 3: Formulate a candidate solution. After observing and gathering experience from real-world ML projects and reviewing the literature, we decided to focus on the creation of a candidate solution that can support the specification and design of ML-enabled systems. As a first step, we proposed a catalog of 45 concerns presented in Chapter 4. This initial set of concerns were evaluated in a focus group with practitioners with different levels of experience of a R&D initiative, more specifically, three data scientists, two developers and three project leads. Their feedback was positive as they perceived the catalog of concern as prominent, and allowed us to identify initial improvements.

Therefrom, we used this catalog to create a candidate solution for identifying concerns when specifying ML-enabled systems (VÍLLAMIZAR; KALINOWSKI; LOPES, 2022). This candidate solution modeled the concerns in a structured manner by proposing a diagram that categorizes the concerns into perspectives, pointing out relationships and stakeholders involved in the analysis of the concerns. The purpose was to capture essential information about the desired functionality, components, and constraints of ML-enabled systems. Figure 5.1 shows the diagram we proposed in a first effort to support the specification of ML-enabled systems.

We iteratively improve this candidate solution by conducting three different evaluations that are briefly described hereafter. The resulting approach, which we baptized *PerSpecML*, is detailed in Section 5.3.

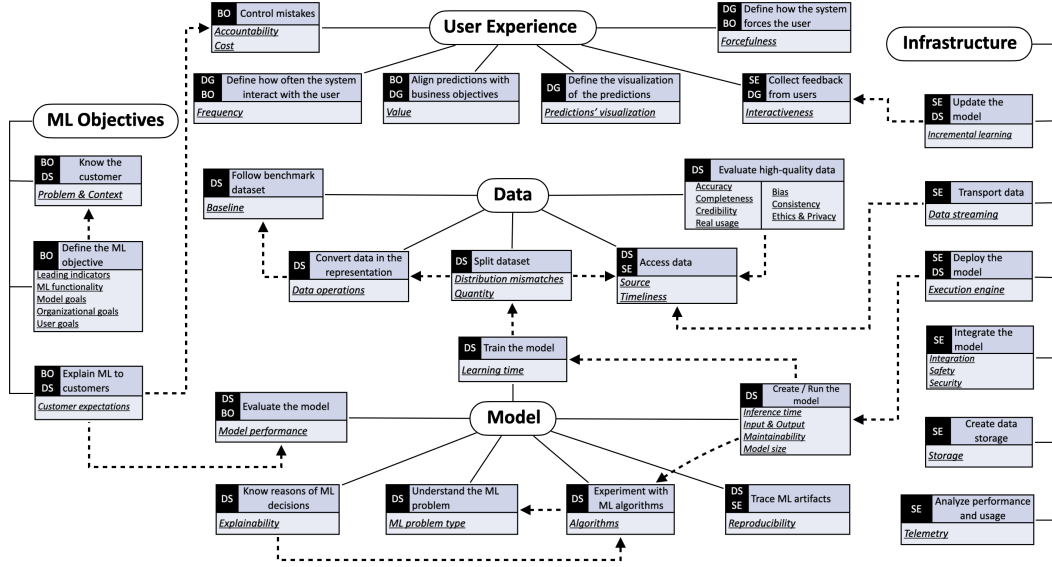


Figure 5.1: Candidate solution for identifying concerns when specifying ML-enabled systems. (VÍLLAMIZAR; KALINOWSKI; LOPES, 2022).

Steps 4, 5, and 6: Evolution and transfer preparation through validation. The goal of these steps was to refine the candidate solution towards its industry-readiness. In order to accomplish this goal, we conducted three evaluations in different contexts, as suggested by (GORSCHKE et al., 2006):

- Validation in academia** with students from two courses on SE for data science specifying an ML-enabled system for a toy scenario (validation in academia).
- Static validation** with practitioners working in a R&D initiative discussing specifications of ML-enabled systems built retroactively with stakeholders of real projects.
- Dynamic validation** in two industrial case studies conducted with an e-commerce company, specifying real ML-enabled systems from scratch using the approach.

Note that, according to (GORSCHKE et al., 2006), the terminology ‘static’ refers to evaluating the candidate solution off-line, involving industry participants and real artifacts, but not as part of a real project life-cycle activity, which is the ‘dynamic’ one. With these iterative validations we seek to ensure early issue detection, user satisfaction, continuous improvement, adaptability and overall confidence in the final solution. Details on the validations are provided in Section 5.4, 5.5, and 5.6.

Step 7: Release the solution. *PerSpecML*, which is presented in the next section, is now being adopted within the R&D initiative involved in the

static validation to specify their ML-enabled system projects. In addition, the approach has been successfully transferred to the data science team responsible for the two case study projects involved in the dynamic validation. At first, the team decided to limit *PerSpecML* to ML projects involving supervised learning tasks. The full adoption is pending results from other evaluations.

5.3 *PerSpecML*

In this section, we present *PerSpecML*, a perspective-based approach for identifying concerns when specifying ML-enabled systems that involves analyzing 60 concerns related to typical tasks that practitioners face in ML projects when defining and structuring such systems. The concerns are grouped into five perspectives: system objectives, user experience, infrastructure, model, and data, providing a structured way to analyze and address different aspects of a large system, including ML components. Together, these perspectives align the activities between business owners, domain experts, designers, software and ML engineers, and data scientists. By using *PerSpecML*, practitioners are expected to be able to:

- **Enhance clarity.** Different stakeholders such as software engineers and data scientists may have varying goals, requirements, and concerns. Modeling perspectives and tasks helps to identify and explicitly represent these diverse viewpoints, ensuring a clear understanding of the ML-enabled system from multiple angles.
- **Foster collaboration.** Providing a perspective-based approach encourages collaboration and communication among stakeholders. It facilitates discussions and negotiations by providing a common structure to express and compare different viewpoints.
- **Identify trade-offs.** Perspectives and concerns enable the exploration of trade-offs between conflicting objectives and requirements. By explicitly modeling a high-level ML-enabled system workflow, practitioners can analyze the impact of design decisions on each perspective and make informed choices that balance different concerns.
- **Improve decision-making.** Understanding the tasks and concerns of both ML and no-ML components helps practitioners to evaluate and compare alternative solutions, enabling informed decision-making as the project progresses. ML projects are full of decisions that stakeholders must make.

- **Ensure completeness.** By considering multiple perspectives and concerns, practitioners can uncover hidden or overlooked requirements or risks. This helps in ensuring that the final ML-enabled system addresses the needs of all stakeholders and avoids potential pitfalls or shortcomings.

In the following, we detail each element of *PerSpecML* that we evolved throughout the iterative validations we conducted. We describe the stakeholders, the perspectives and their concerns, the relationship between them, and the two final artifacts that structure the above elements: the perspective-based ML task and concern diagram and the corresponding specification template. We also describe the logical flow for executing *PerSpecML*.

5.3.1 Stakeholders

Building successful ML-enabled systems requires a wide range of skills, typically by bringing together team members with different specialties (KIM et al., 2017; HULTEN, 2019). Taking a holistic system view is essential because ML expertise alone is not sufficient and even engineering skills to, for example, build pipelines and deploy ML models cover only small parts of a larger system. We also need to be concerned about how to improve the experience of end-users in order to deal with unrealistic assumptions, and align business value to ML technical activities in order to cover business requirements. Given this, we seek *PerSpecML* to impact the work of business owners, domain experts, designers, software/ML engineers, data scientists and requirements engineers. Note that these stakeholders can also represent specific roles within ML projects.

Business owners (BO) should understand what properties and components are essential to achieve the business objectives and be aware of the ML capabilities in order to set realistic goals and expectations. For instance, how to connect business objectives with ML outcomes? What is the real cost involved in maintaining an ML-enabled system? What team and skills are needed to successfully building ML-enabled systems?

Domain experts (DE) play an important role in accurately defining the problem in a way that aligns with real-world scenarios and requirements, ensuring that the ML-enabled system addresses the specific challenges and objectives of the domain. By collaborating closely with domain experts, other stakeholders can benefit from their in-depth knowledge and insights to define relevant features and data sources, and interpreting the results of the ML model in a meaningful context.

Designers (DG) collaborate to translate complex ML concepts and model outputs into intuitive and easy-to-understand interfaces that provide value to end users. For instance, where and how the ML outcomes will appear? how often it will appear? and how forcefully it will appear? A good user experience must be on the user’s side and make them happy, engaged, and productive. Creating interactions with users to get feedback and grow learning is essential to ensure the quality of the ML model over time.

Software/ML engineers (SE) should understand how the entire system will interact with the ML model. They work on transforming the data scientists’ research prototypes into ML-enabled systems that can handle large-scale data, ensure scalability, and meet performance concerns. For instance, what are the pros and cons of deploying an ML model as a back-end application or as a web service? online or batch predictions are enough to meet user demand?

Data scientist (DS) leverages their expertise in data analysis, statistical modeling, and ML algorithms to extract insights, develop ML models, and drive data-driven decision-making, but they should also understand the constraints these systems put on the ML models they produce. For instance, what quality properties the ML model should consider? What domain restrictions may apply? what should be the complexity of the ML model? and how should the ML model be tuned to maximize business results?

Requirements engineers collaborate closely with stakeholders to support the discussions between business owners, domain experts, and data scientists, and the development team, facilitating effective communication and understanding of project requirements. We seek to empower requirements engineers by using *PerSpecML* to identify and resolve conflicts often associated with ML projects. For instance, how much loss of accuracy is acceptable to cut the inference latency in half? can data scientists sacrifice some accuracy but offer better interpretability and explainability? One of the main benefits of applying RE for ML projects is to help balance these concerns.

5.3.2 Concerns

One of the main elements of *PerSpecML* are its concerns. In total, we identified 60 concerns, of which 45 came from the catalog presented in Chapter 4, and the remaining 15 came from the evaluations conducted to iteratively improve our solution. *PerSpecML* highlights concerns such as data streaming, model serving and telemetry when thinking on the operation of the ML-enabled system, and inference time, explainability and reproducibility

when thinking on the development of the ML model. In *PerSpecML*, the concerns are part of tasks that stakeholders typically face throughout the development of ML-enabled systems.

5.3.3

Related Tasks Modeling

In *PerSpecML* we also focus on capturing and representing the tasks that should be performed by stakeholders to develop successful ML projects. In total, our approach outlines 28 tasks that are covered by the five perspectives. These tasks group associated concerns that should be analyzed by stakeholders. With this feature, stakeholders can more easily understand and describe how tasks are performed, what concerns are involved, the relationships between concerns, and the interactions with other stakeholders. For instance, typically in ML projects, data scientists are tasked with training, validating, and deploying ML models. These tasks involve implicit concerns that are not easily identified at first sight, such as inference time, learning time, model complexity and hyperparameters tuning. In addition, some specific tasks can benefit from involving more than one stakeholder in the analysis. For instance, to validate ML models it is necessary to generate model performance metrics, typically performed by data scientists, and analyze such metrics in collaboration with domain experts who deep understand the problem and data.

In the early phases of developing ML-enabled systems, several key tasks should be performed to lay a strong foundation for the project's success. These tasks typically involve all the stakeholders, and concern understanding the problem, setting goals, among other. Table 5.1 details the tasks from a system objectives perspective.

Table 5.1: Description of the tasks to define the system objectives.

Task	Description
Understand the problem	understand the problem and the context in which the ML model will be deployed, and define the ML problem and the specific task to be solved
Set goals at different levels	define the ML project goals at different levels to ensure that it meets the stakeholders' expectations
Establish success indicators	define measures that provide early insights on the achievement of the objectives
Manage expectations	define what the ML model can and cannot do. Stakeholders may have unrealistic expectations about the ML capabilities

A positive user experience is crucial for the successful adoption, acceptance, and utilization of ML-enabled systems. It enhances user engagement,

Table 5.2: Description of the tasks to ensure user experience.

Task	Description
Establish the value of predictions	determine that the ML model's outputs are relevant, accurate, and impactful and how they contribute to achieving the project's objectives
Define the interaction of predictions with users	define how users will interact with predictions (<i>e.g.</i> , frequency and forcefulness) in order to design user-friendly interfaces and workflows
Visualize predictions	present ML model outputs in a visually understandable format
Collect learning feedback from users	offer feedback mechanisms to users in order to provide updates on ML models
Ensure the credibility of predictions	ensure that users have a clear understanding of potential inaccuracies of the ML model

improves user satisfaction, and ultimately contributes to the overall success of the ML project. Table 5.2 details the tasks should be done to ensure that ML-enabled systems become a valuable and integral part of users' workflows.

A robust and well-designed infrastructure is fundamental for the success of ML projects. It enables efficient development, deployment, and scaling of ML models. Table 5.3 details the tasks of the infrastructure perspective.

Table 5.3: Description of the tasks to support the infra of ML-enabled systems.

Task	Description
Transport data to the model	involves moving the data from its source to the ML model for analysis, training, or prediction
Make the ML model available	refers to the process of deploying the trained ML model so that it can be accessed by users
Update the ML model	refers to the process of making improvements to an existing ML model to enhance its performance
Store ML artifacts	involves the storage and management of the artifacts generated in the ML development process
Observe the ML model	involves analyzing the performance, behavior, and outcomes of both the ML model and the system
Automate the ML workflow	involves the implementation of a streamlined process that automates the ML workflow
Integrate the ML model	involves incorporating the trained ML model into the larger system where it will be used
Evaluate the cost of infrastructure	analyze the expenses related to the computational resources required to support the ML project

Table 5.4: Description of the tasks to support the creation of ML models.

Task	Description
Select and configure the ML model	shortlist a set of ML algorithms that are well-suited for the task at hand, and experiment with different hyperparameters
Train the ML model	create an ML model that captures the underlying patterns in the data
Validate the ML model	ensure that the trained ML model meets the desired criteria
Deploy the ML model	make the trained ML model operational in a production environment, allowing it to serve predictions to end-users or other systems
Evaluate other quality characteristics	assess various aspects of the ML model beyond its predictive

A structured ML model development process fosters transparency, reproducibility, and accountability. It supports the creation of robust, reliable, and trustworthy ML solutions. Table 5.4 details the tasks of the model perspective.

The management of data in ML projects is essential for building accurate and reliable ML models. Table 5.5 details the tasks to be done, mainly by data scientists and domain experts, to maintain high-quality data throughout the lifecycle of ML projects.

Table 5.5: Description of the tasks to support data quality in ML projects.

Task	Description
Access data	involves timely obtaining and retrieving the necessary data from various sources to be used for model development and evaluation
Select and describe data	involves carefully choosing the relevant data that will be used to train and validate ML models, and describing the features of the data
Evaluate high-quality data	involves a comprehensive assessment of the data used for training and testing ML models to ensure that the data meets certain criteria to produce reliable results
Convert data in the representation of the ML model	involves transforming the raw input data into a format that can be processed by the ML algorithm
Split dataset	involves dividing the data into separate subsets for training and validation purposes
Define a golden dataset	involves creating a high-quality dataset that represents the problem and serves as reference for training and evaluating ML models

5.3.4 Perspectives

In *PerSpecML*, we modeled five perspective that are detailed as follows.

System Objectives Perspective. When evaluating ML solutions, there is a tendency to focus on improving ML metrics such as the F1-score and accuracy at the expense of ensuring business value and covering business requirements (BARASH et al., 2019). Success in ML-enabled systems is hard to define with a single metric, therefore it becomes necessary to define success at different levels. This perspective involves analyzing the context and problem that ML will address to ensure that ML is targeting at the right problem; defining measurable benefits ML is expected to bring to the organization and users; what system and model goals will be evaluated; the ML expected results in terms of functionality, and trade-off to deal with customer expectations. Table 5.6 details the concerns when thinking on objectives for such systems.

User Experience Perspective. A good ML-enabled system includes building better experiences of using ML. The goal of this perspective is to present the predictions of the ML model to users in a way that achieves the

Table 5.6: Description of each concern of the system objectives perspective.

Id	Concern	Addressing this concern involves
O1	Context	the specific circumstances or conditions in which the ML-enabled system will operate
O2	Need	the desire that must be addressed to achieve a particular condition within a given context
O3	ML functionality	the desired outcome of the ML model (<i>e.g.</i> , classify customers)
O4	Profit hypothesis	how the ML system's outcomes will translate into tangible gains for the organization
O5	Organizational goals	measurable benefits ML is expected to bring to the organization
O6	System goals	what the larger system tries to achieve with the support of an ML model
O7	User goals	what the users want to achieve by using ML
O8	Model goals	metrics and acceptable measures the ML model should achieve
O9	Leading indicators	measures correlating with future success, from the business' perspective (<i>e.g.</i> , customer sentiment and engagement)
O10	ML trade-off	the balance of customer expectations (<i>e.g.</i> , inference time vs accuracy)

system objectives and gets user feedback to improve the ML model. Therefore, we consider analyzing concerns such as defining what is the added value as perceived by users from the predictions to their work; how strongly the system forces the user to do what the ML model indicates; how often the ML model interacts with users; how the predictions will be presented so that users get value from them; how the users will provide new data for learning; and what is the user impact of a wrong ML model prediction. Table 5.7 details the concerns when thinking on user experience for ML-enabled systems.

Infrastructure Perspective. ML models produced by data scientists typically are turned into functional and connected software systems that demand special characteristics when in operation. The goal of this perspective is to cover the execution of the ML model, the monitoring of both data and model outputs, and its learning from new data. We consider analyzing concerns such as defining what streaming strategy will be used to connect data with the ML model; how the ML model will be served; the need for the ML model to continuously learn from new data to extend its knowledge; where the ML artifacts (*e.g.*, experiments, ML models, datasets) will be stored; the need for monitoring the ML model and data; the strategy to automate ML operations that allow to reproduce and maintain ML artifacts, and the integration the

Table 5.7: Description of each concern of the user experience perspective.

Id	Concern	Addressing this concern involves
U1	Value	the added value as perceived by users from the predictions
U2	Forcefulness	how strongly the system forces the user to do what the ML model indicates they should
U3	Frequency	how often the system interacts with users
U4	Visualization	user-friendly interfaces to showcase the ML model's outputs
U5	Learning feedback	what interactions the users will have with the system to provide new data for learning
U6	Acceptance	how well the ML model arrives at its decisions
U7	Accountability	who is responsible for unexpected model results
U8	Cost	the user impact of a wrong ML model prediction
U9	User education & Training	the need to provide user education and training on the limitations of the ML model and how to interpret its outputs

ML model will have with the rest of the system functionality. Table 5.8 details the concerns when thinking on the infrastructure for ML-enabled systems.

Model Perspective. Building a ML model implies not only cleaning and preparing data for analysis, and training an algorithm to predict some phenomenon. Several other aspects determine its quality. This perspective involves analyzing concerns such as defining the initial candidate of expected inputs and outcomes (of course, the set of meaningful inputs can be refined during pre-processing activities); the set of algorithms that could be used according to the problem to be addressed; the need to tune the hyperparameters of the algorithms; the metrics used to evaluate the ML model and measurable performance expectations that tend to degrade over time; the need for explaining and understanding reasons of the model outputs; the ability of the ML model to perform well as the size of the data and the complexity of the problem increase (scalability), to deal with discrimination and negative consequences for certain groups (bias & fairness), to protect sensitive data and prevents unauthorized access (security & privacy); the acceptable time to train and execute the ML model, and the complexity of the ML model in terms of size and generalization. In Table 5.9, we provide the description of the concerns that may be relevant to select, train, tune and validate a ML model.

Data Perspective. Data is critical to ML. Poor data will result in inaccurate predictions. Hence, ML requires high-quality input data. Based on the Data Quality model defined in the standard ISO/IEC 25012 (ISO/IEC, 2012) and our own experience, we elaborate on the data perspective. In this

Table 5.8: Description of each concern of the infrastructure perspective.

Id	Concern	Addressing this concern involves
I1	Data streaming	what data streaming strategy will be used (<i>e.g.</i> , real time data transportation or in batches)
I2	Model serving	how the ML model will be executed and consumed (<i>e.g.</i> , client-side, web service end-point)
I3	Incremental learning	the need for the ML model continuously learns from new data
I4	Storage	where the ML artifacts (<i>e.g.</i> , models, data, scripts) will be stored
I5	Monitorability	the need to monitor the data and the outputs of the ML model to alert when data drifts
I6	Telemetry	what ML-enabled system data needs to be collected. Telemetry involves collecting data such as clicks on particular buttons
I7	Reproducibility	the need to repeatedly run an ML process on certain experiments and obtain similar results
I8	Maintainability	the need to modify ML-enabled systems to adapt to a changed environment
I9	Integration	the integration of ML model with a larger system
I10	Hybrid decision intelligence	the essence of combining ML model outputs with rule-based to create comprehensive results
I11	Cost	the financial cost involved in executing the inferences of the ML model

perspective, we considered concerns such as defining from where the data will be obtained; the strategy to select data; the description of data; evaluating the inherent quality data attributes (*e.g.*, accuracy, completeness, consistency, real usage); what data operations and modeling must be applied; the expected data distributions and how data will be split into training, validating and test data; the time between when data is expected and when it is readily available for use, and the need for a golden dataset approved by a domain expert. Table 5.10 details the concerns when thinking on data for ML-enabled systems.

5.3.5 Relationship between Concerns

Identifying relationships that show influence and implications between the concerns of an ML-enabled system is important for successful project outcomes. These relationships extend across various dimensions, such as system design, risk management, and resource allocation. Understanding these factors allows for optimal decision-making, alignment with ML project goals, and efficient workflow planning.

In *PerSpecML*, we highlight these relationships to (i) help stakeholders

Table 5.9: Description of each concern of the model perspective.

Id	Concern	Addressing this concern involves
M1	Algorithm & model selection	the algorithms that could be used based on constraints such as explainability, performance
M2	Algorithm tuning	the need to choose optimal hyperparameters for a learning algorithm
M3	Input & Output	the expected inputs (features) and outcomes of the ML model
M4	Learning time	the acceptable time to train the model
M5	Performance metrics	the metrics used to evaluate the ML model (<i>e.g.</i> , precision, recall, F1-score)
M6	Baseline model	the current model that acts as a reference to contextualize the results of trained models
M7	Inference time	the acceptable time to execute the model and return the predictions
M8	Model size	the size of the model in terms of storage and its complexity
M9	Degradation	the awareness of performance degradation
M10	Versioning	the versions of libraries, ensuring compatibility, and handling any conflicts that may arise due to dependencies
M11	Interpretability & Explainability	the need to understand reasons for the model inferences
M12	Scalability	the need for the model to perform well as the size of the data and the complexity of the problem increases
M13	Bias & Fairness	the need for the model to treat different groups of people or entities
M14	Security & Privacy	the need for the model to protect sensitive data and prevents unauthorized access

identify conflicting objectives and requirements, and (ii) promote transparent communication between team members, ensuring the long-term viability and impact of ML projects. For instance, if users require to know the reasons of the ML model's decision-making then the explainability & interpretability concern arises. But this may depend on the chosen algorithm since some ML algorithms tend to be less explainable than others (*e.g.*, simpler ML algorithms such as decision trees, linear regression, and logistic regression are often considered more explainable than complex ML algorithms such as deep neural networks, random forests, and gradient boosting models). In addition, complex ML models may provide high accuracy, making it necessary to strike a balance between these concerns based on the specific needs and constraints of the ML project.

Identifying these relationships is also important within the infrastructure

Table 5.10: Description of each concern of the data perspective.


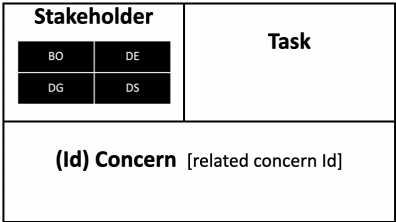
Id	Concern	Addressing this concern involves
D1	Source	from where the data will be obtained
D2	Timeliness	the time between when data is expected and when it is readily available for use
D3	Data selection	the process of determining the appropriate data type and suitable samples to collect data
D4	Data dictionary	the collection of the names, definitions, and attributes for data elements and models
D5	Quantity	the expected amount of data according to the problem type and the algorithm complexity
D6	Accuracy	the need to get correct data
D7	Completeness	the need to get data containing sufficient observations of all situations
D8	Credibility	the need to get true data that is believable and understandable by users
D9	Real usage	the need to get data representing the problem
D10	Bias	the need to get data fair samples and representative distributions
D11	Consistency	the need to get consistent data in context
D12	Ethics	the need to get data to prevent adversely impacting society
D13	Anonymization	the need to anonymize data while still maintaining the utility of the data for ML purposes
D14	Data operations & Modeling	what operations must be applied on the data and what is necessary to convert data in the representation of the model
D15	Data distribution	the expected data distributions and how data will be split into training and validating data
D16	Golden dataset	the need for a baseline dataset approved by a domain expert that reflects the problem

perspective. For instance, defining the source to access data influences the implementation or setup of a data streaming solution, which is required to transport the data to the ML model. Understanding these kind of relationships helps optimize the ML workflow and streamline the project execution. On the other hand, in the system objectives perspective, the ML functionality guides the selection of appropriate ML algorithms (*i.e.*, different tasks, such as classification or regression, require specific algorithms that are suitable for the task at hand). Furthermore, it affects how the ML model's performance is evaluated and measured (*i.e.*, different performance metrics, such as accuracy or recall are used based on the specific task). All *PerSpecML* relationships are outlined and detailed in Appendix B.

5.3.6
Perspective-Based ML Task and Concern Diagram

In order to provide a holistic view of the ML-enabled system that facilitates producing a description of what will be built and delivers it for approval and requirements management, we present a perspective-based ML task and concern diagram that integrates the key elements discussed earlier (concerns and their relationships, tasks, perspectives, and stakeholders). Table 5.11 shows the notation we used to represent these components in the diagram.

Table 5.11: Legend of the perspective-based ML task and concern diagram.

Notation	Description
	The diagram contains five rounded rectangles that represent the perspectives. Each perspective is associated with a color to facilitate its identification, and is connected to their tasks
	The diagram contains rectangles attached to a perspective that connect a task (at the top right) to one or more concerns (at the bottom). Each task has at least one actor suggested (at the top left) related to the execution of the task and the analysis of the concerns

The perspective-based ML task and concern diagram shown in Figure 5.2 serves as a visual representation of the interplay between these elements and their relationships within the context of ML projects. It offers a comprehensive overview of how different perspectives shape the tasks at hand, while considering the specific concerns associated with each task. Additionally, it highlights the involvement of various stakeholders who contribute their expertise and insights throughout the development process. By presenting this integrated diagram, we aim to provide a clear and structured approach for understanding the complex dynamics involved in building successful ML-enabled systems.

Based on the perspective-based ML task and concern diagram, we established a definition of what a requirements specification is for an ML-enabled system.

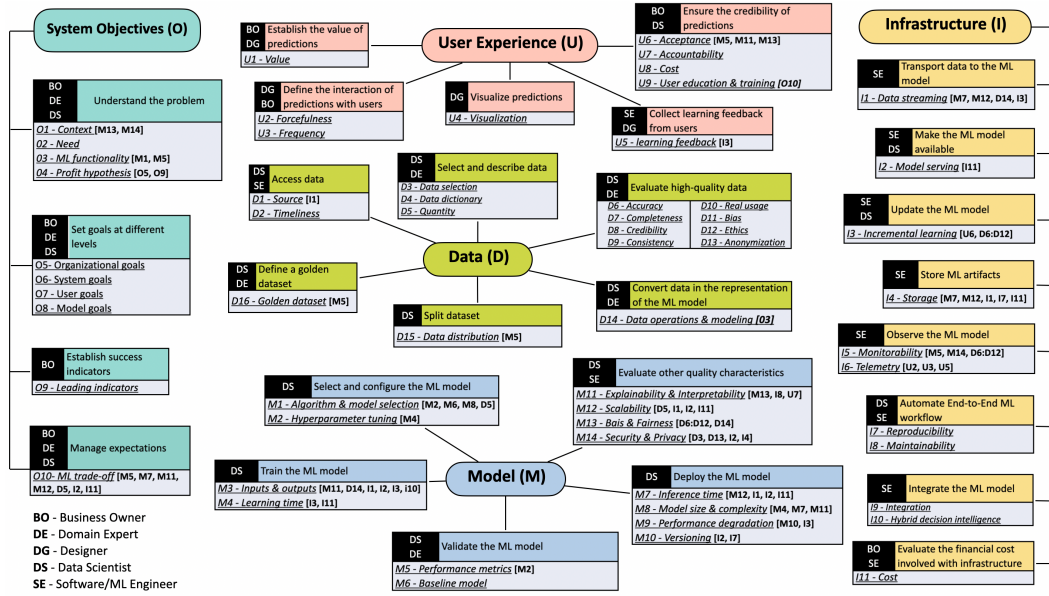


Figure 5.2: An illustration of the perspective-based ML task and concern diagram.

The requirements specification for ML-enabled systems can be seen as a detailed document that outlines the properties, in this thesis called concerns, used to evaluate not only an ML model and its data, but also extend to the operations of the system employing it (infrastructure), interactions with its end-users (user experience), and ML contexts (objectives) that provides the necessary background for making informed decisions throughout the ML-enabled system development lifecycle.

5.3.7 Perspective-Based ML Specification Template

Documenting and organizing requirements is crucial for ensuring a clear understanding of the desired software system functionality, facilitating communication and collaboration, verifying and validating requirements, managing changes, and enabling knowledge transfer. It plays a vital role in successful software development and project outcomes. To fulfill these commitments, we proposed a specification template based on the Perspective-Based ML Task and Concern Diagram. This template offers a standardized format for systematically documenting and organizing the applicable concerns associated with ML-enabled systems. We refer to this document as the Perspective-Based ML Specification Template, and its constituent elements are illustrated in Figure 5.3.

The template designed for documenting and organizing requirements of ML-enabled systems incorporates six distinct elements for each perspective,

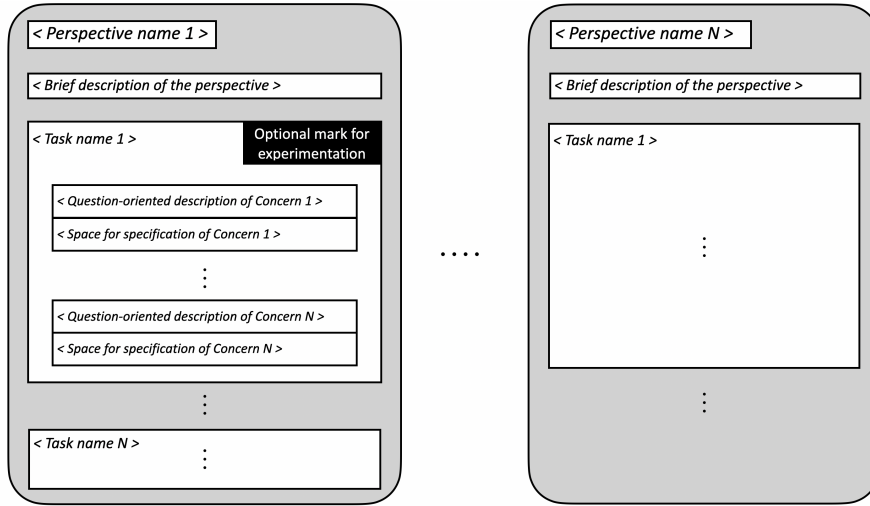


Figure 5.3: Elements of the Perspective-Based ML Specification Template.

each outlined as follows:

1. **Perspective Name:** Positioned at the top of the template, this element identifies the specific perspective under consideration.
2. **Perspective Description:** This second element offers an overview of the perspective, providing practitioners with a contextual understanding of its significance within the ML-enabled system.
3. **Task Names:** The third element entails the names of tasks within a given perspective. Multiple tasks may exist within a single perspective, each with its set of associated elements.
4. **Experimentation Mark ('E'):** The fourth element is a designated mark ('E') indicating tasks that involve an experimentation component (*e.g.*, select and configure the ML model). This implies that concerns within these tasks may be subject to refinement as the ML project progresses.
5. **Question-Oriented Descriptions:** Element five comprises question-oriented descriptions for each concern within a task. This serves as a guide for practitioners, allowing them to explore and assess each concern systematically.
6. **Space for Concern Specification:** Finally, the sixth element provides dedicated space for practitioners to specify details related to each applicable concern, allowing for a comprehensive documentation of the ML-enabled system.

Instead of starting from scratch each time, stakeholders can utilize this predefined template that already includes relevant sections, headings, and prompts, saving time and effort during the specification process. This may reduce redundancy and allow stakeholders to focus on the specific details and concerns of the ML-enabled system.

For example, consider the scenario where operational staff in charge of ML engineers examine the Perspective-Based ML Task and Concern Diagram. Upon identifying that the concern regarding the strategy for storing ML artifacts is pertinent to the system under development, the Perspective-Based ML Specification Template provides targeted prompts. In this case, these prompts, in the form of questions, guide ML engineers to consider ML artifacts such as models, data, experiments, and environments that should be stored. In another instance, if there is a concern regarding enhancing the performance of ML algorithms, the template highlights potential solutions, such as hyperparameter tuning.

Through a detailed analysis of the perspective's description, the experimentation markers associated with certain concerns, and the question-oriented descriptions of these concerns, we aim to empower stakeholders to undertake a comprehensive and systematic exploration of the ML-enabled system's requirements. We make the Perspective-Based ML Specification Template available in our online repository¹, accompanied by one illustrative example and two real case studies where the template was filled out to detail three distinct user stories incorporating ML components. Due to limitations in size on the Miro Board, Figure 5.4 showcases a segment of the Perspective-Based ML Specification Template focused on the model and data perspectives.

Model Perspective

Building an ML model implies not just training an algorithm with data to predict or classify some phenomenon. Several other aspects determine its success.

Define the inputs (features) and output (label) of the model E

- What will be the **input and output data** of the model? (the set of inputs can be refined during experimentation activities)

<Space for specification>

Define the set of algorithms to be considered during the model selection E

- What is the **set of algorithms** that can be used/investigated based on the ML problem type and other constraints?

<Space for specification>

Data Perspective

Bad data will result in inaccurate predictions, something like "garbage in, garbage out". Thus, ML requires high quality input data.

Define the strategy to access the data

- What is the **source/origin** where the data is stored (e.g., worksheets, databases, on cloud solutions)?

<Space for specification>

Define the strategy to select and describe the data

- How much data will be **selected**?
 - What do the collected data mean? (consider to create a **data dictionary**)

<Space for specification>

Figure 5.4: Excerpt of the Perspective-Based ML Specification Template for the model and data perspectives.

¹<https://doi.org/10.5281/zenodo.7705002>

5.3.8

How to apply *PerSpecML*

In order to provide clarity, structure, reproducibility, and consistency, this section shows the steps to be followed for executing *PerSpecML*. The purpose is to break down the overall process into manageable and sequential tasks, making it easier for stakeholders to understand and follow. Figure 5.5 shows the workflow to ensure that *PerSpecML* is applied in a systematic and organized manner, leading to more successful outcomes.

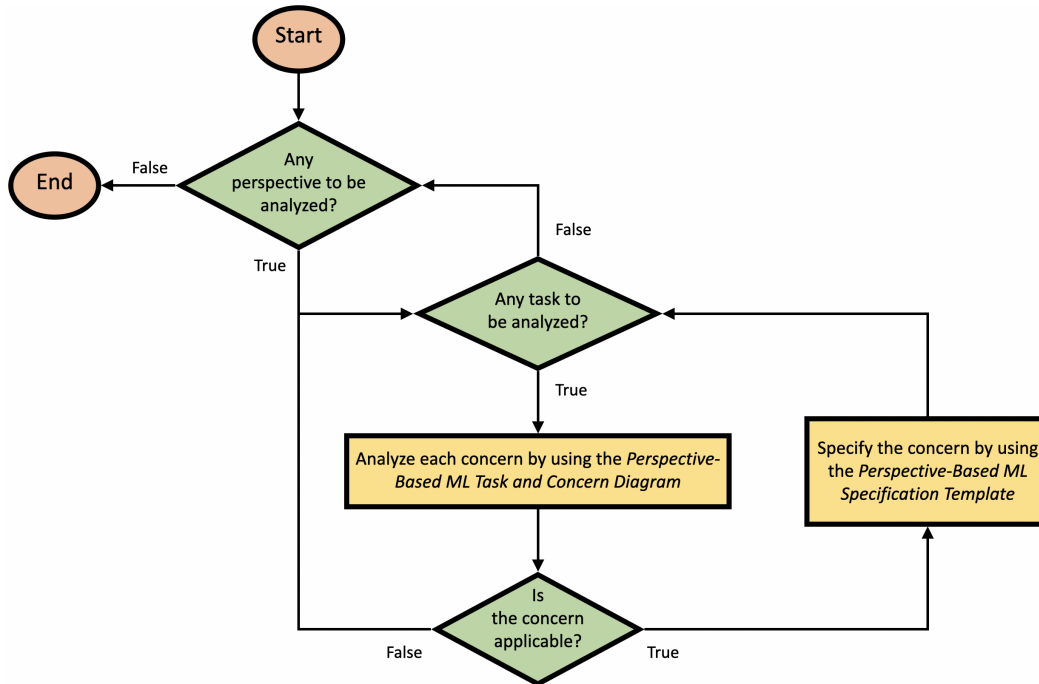


Figure 5.5: Logical flow for executing *PerSpecML*.

In the following, we break down the workflow to provide additional information that we consider relevant to apply *PerSpecML*.

1. **Analyze Each Perspective:** This step involves analyzing the perspectives in the following order:
 - System Objectives.
 - User Experience.
 - Infrastructure.
 - Model.
 - Data.
2. **Analyze Each Task within Each Perspective:** This involves breaking down each perspective into tasks and understanding the associated concerns.

- Tasks: A total of 28 tasks across the five perspectives must be analyzed.
- Concerns: A total of 60 concerns across all tasks and perspectives must be analyzed.

3. Analyze Concerns Using the Perspective-Based ML Task and Concern Diagram: This resource offers insights and connections before specifying details of the concerns.

- Applicability check: Before diving into detailed analysis, practitioners must determine if the concern is relevant to the ML-enabled system.

4. Specify Concerns Using the Perspective-Based ML Specification Template: This resource seeks to frame the concern as a question that prompts additional information for precise specification.

Each step is a structured approach to break down the complexity of specifying an ML-enabled system. It emphasizes thorough analysis, ensuring that all facets, from objectives to user experience, infrastructure, model, and data, are comprehensively considered through a series of tasks and concerns, leading to a well-defined and refined system specification. We expect *PerSpecML* to be used by requirements engineers or practitioners performing or representing that function in collaboration with the recommended stakeholders (business owners, domain experts, designers, software/ML engineers, and data scientists).

5.3.9

Application Example of *PerSpecML*

This section provides a demonstration of how *PerSpecML* can be applied to systems that incorporate an ML component. In this case, we provide a hypothetical scenario, including a user story and acceptance criteria, to address a sentiment analysis problem. Here, we seek to illustrate the application of *PerSpecML* simulating a real-world context. We present this case in a user story and acceptance criteria format since it brings clarity to the application of a methodology, making it more accessible and understandable for a wider audience.

User story:

As a business owner in the e-commerce industry, I want to analyze customer reviews for sentiment, So that I can gain insights into customer satisfaction and make data-driven business decisions

Acceptance criteria:

1. The sentiment analysis model should accurately classify customer reviews as positive, negative, or neutral based on the expressed sentiment
2. The model should provide a confidence score or probability for each sentiment prediction to indicate the level of certainty
3. The system should process a large volume of customer reviews in a timely manner to enable real-time or near-real-time analysis
4. The sentiment analysis results should be presented in an easily interpretable format, such as a sentiment distribution chart
5. The system should allow filtering and searching of customer reviews based on sentiment to facilitate in-depth analysis
6. The sentiment analysis model should be regularly evaluated and updated to ensure its performance remains reliable and accurate over time
7. The system should handle potential challenges such as language variations, slang, or sarcasm in customer reviews to ensure robust sentiment analysis
8. The sentiment analysis solution should be scalable, capable of processing an increasing number of customer reviews as the business grows
9. The model should be designed with fairness and bias mitigation techniques to ensure equitable sentiment analysis across different customer groups
10. The system should prioritize data privacy and security, ensuring that customer reviews are handled and stored securely in compliance with relevant regulations

In this example, the ML-enabled system has a sentiment analysis component that enables the business owner to gain valuable insights from customer reviews. By accurately analyzing sentiment, the system will empower the business to make data-driven decisions, identify areas for improvement, and enhance customer satisfaction. In the following, we apply the *PerSpecML* approach for two perspectives: system objectives and infrastructure.

First, we analyzed the system objectives perspective in order to identify and define the primary goals and purpose of the system within the context

it operates. When looking at the Perspective-Based ML Task and Concern Diagram, we observe 10 concerns within this perspective grouped into four tasks. We understand that all these concerns apply. Therefore, after analyzing each one, we specify them by using the Perspective-Based ML Specification Template. To facilitate its visualization in this static text document, we show the specifications of these concerns in Table 5.12.

Table 5.12: Specification of the concerns of the system objectives perspective.

Concern	Specification
Context	With a large volume of customer reviews generated daily, manually analyzing and extracting sentiment from these reviews becomes a time-consuming and error-prone process
Need	Analyzing the customer reviews automatically in order to respond quickly to the market demands
ML functionality	Classify customer reviews into positive, negative, or neutral based on their expressed sentiment
Profit hypothesis	Improve customer satisfaction by promptly addressing negative sentiments expressed in reviews
Organizational goals	Enhancing customer satisfaction, Improving brand reputation, increasing customer retention, and gaining competitive advantage in the market
System goals	Scalability to handle large volumes of customer reviews, real-time processing of reviews, easy integration with existing systems and workflows
User goals	Promptly identifying customer concerns, understanding customer sentiment towards specific products, and tracking overall customer satisfaction and its impact on business outcomes
Model goals	High accuracy in sentiment classification, robustness to handle variations in language and expression, interpretable outputs to understand the factors influencing sentiment, handling sentiment in different domains or industries
Leading indicators	Volume of changes and new customer reviews. For example, if the system observes a sudden increase in negative sentiment, it indicates potential issues
ML trade-off	The accuracy of the negative reviews is more important than the accuracy of the positive reviews

Note that the system objectives perspective encompasses four concerns (*Context*, *ML functionality*, *Profit hypothesis*, *ML-trade off*) that interrelate with other facets of the ML-enabled system. The specification of these concerns holds the potential to impact various system aspects. For instance, the ‘*Context*’ influences ‘*ethical*’ considerations within the data perspective, since critical domains such as medical diagnosis need to be carefully designed to avoid unfair outcomes. Similarly, the ‘*ML functionality*’ influence concerns within the model perspective, such as *model selection* and *Performance metrics*. In this case, the definition of ML functionality guides the selection of appropriate algorithms and affects how the ML model’s performance is evaluated.

After completing the analysis of concerns within the system objectives perspective, the subsequent perspective to be examined is the user experience perspective. However, for illustrative purposes, we opt to specify the infrastructure perspective in Table 5.13, given it encompasses a diverse set of concerns.

Table 5.13: Specification of the concerns of the infrastructure perspective.

Concern	Specification
Data streaming	The system shall ingest data in real-time, ensuring timely analysis and response
Model serving	The system shall has low-latency responses and process high-volume predictions
Incremental learning	The system shall provide the services to adapt and improve the ML model over time
Storage	Storage containers are needed for storing raw and processed data and ML models
Monitorability	The system shall capture system metrics, errors and latency for issue resolution
Telemetry	The system shall analyze user interactions of the sentiment analysis solution
Reproducibility	The system shall ensure the ML model is reproducible across different environments
Maintainability	The system shall follow SE best practices and provide clear documentation
Integration	The system shall integrate with other services with appropriate APIs
Hybrid decision intelligence	The system shall use the ML outputs to create heuristics reflecting the context of the problem
Cost	The infrastructure necessary to execute and maintain the ML model must not exceed the budget

Similar to the system objectives perspective, the infrastructure perspective encompasses seven concerns that exhibit relationships with other system aspects. For instance, when specifying the ‘*data streaming*’ concern, various interrelated aspects were identified. Data streaming plays a critical role in minimizing *latency* by processing and responding to data in near real-time. Its functionality often needs on-the-fly preprocessing and feature extraction, demanding the implementation of efficient techniques. Furthermore, handling high volumes of data is a common aspect of data streaming, requiring designs in ML-enabled systems that demonstrate *scalability*. Moreover, the dynamic nature of data streaming allows for real-time updates and retraining of ML models as new data becomes available.

5.4

Validation in Academia

As we mentioned before, *PerSpecML* is the result of a series of validations that were conducted in different contexts. The first validation was carried out within an academic environment where students were tasked to use the candidate solution introduced in Section 5.2 to specify a toy problem. The simplified nature of the toy problem allowed for a clear understanding of how the candidate solution performed and how it could be improved. This led to valuable lessons and discoveries that were applied in the next validation with a more complex problem. In the following, we detail the validation in academia.

5.4.1

Context

The academic validation took place in the context of two courses on SE for data science with professionals from a Brazilian logistic company called Loggi² (on-line course), and computer science graduate students from the Pontifical Catholic University of Rio de Janeiro (in-person course). This validation began by informing the students about the research study, its objectives, and the nature of their participation. Clear explanations were given regarding the voluntary and non-compulsory nature of their participation, with an emphasis on their right to withdraw from the study at any point without facing any consequences. We did not compensate them in any way. Instead, we underscored the educational value of their participation and how it contributed to the research goals. Participants who opted to engage in the study were tasked with specifying a feature for an ML-enabled system employing an illustrative context of a bank loan scenario. Their assignments

²<https://www.loggi.com>

entailed a detailed examination of the candidate solution’s perspectives and concerns to determine which aspects should be included in the specification. The feature consisted of automatically classifying customers into good or bad payers and was described in user story format.

As a Bank Manager I want to automatically classify customers so that I can decide upon granting a requested loan

From the user story, we can infer that the ML component needs to access, for learning purposes, data on customer characteristics, previously granted loans, and payment records. Regarding non-ML components and integration with other services, the participants could assume restrictions and requirements of the software system that the ML component would use. With this information, we asked the participants to analyze each concern of the candidate solution and provide a reasonable specification, if applicable, in a drafted template we provided. Thereafter, they were asked to individually answer a follow-up questionnaire critically assessing the relevance and completeness of the candidate solution’s perspectives and concerns. In-person participants were allocated a two-hour timeframe to complete the study, a duration that proved sufficient as they successfully concluded within the designated time. For online participants, the time spent on the study was not regulated. All the material provided to the participants is available in our online repository¹. Figure 5.6 illustrates the academic validation.

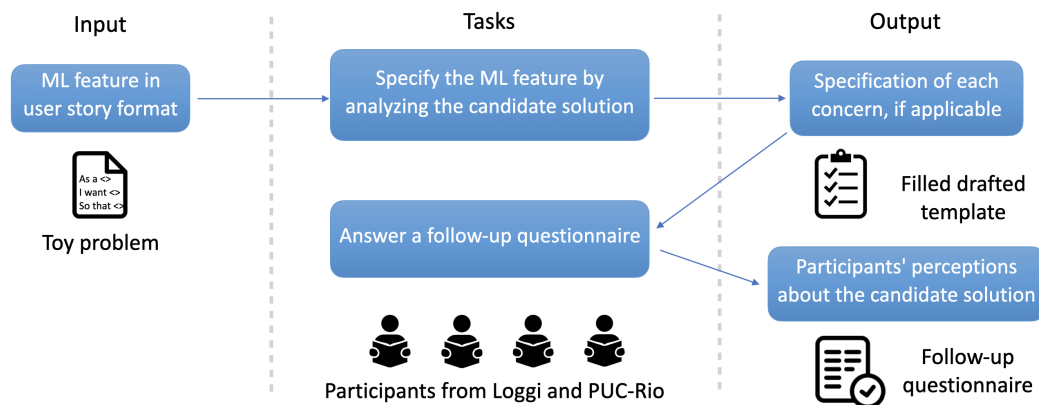


Figure 5.6: Process diagram for the academic validation.

5.4.2 Goal and Method

We detail the goal of the validation in academia in Table 5.14. We followed the Goal-Question-Metric (GQM) goal definition template (BASILI; ROMBACH, 1988), which is a structured approach commonly used in SE

and other disciplines, to help establish a clear connection between the overall goal, the specific questions that need to be answered, and the metrics used to measure progress.

Table 5.14: Study goal definition of academic validation.

Analyze	the candidate solution’s perspectives and concerns
for the purpose of	characterization
with respect to	perceived relevance and completeness, and ease of use, usefulness and intended use
from the viewpoint of	professionals and computer science graduate students
in the context of	two courses with 53 data science professionals from Loggi and 15 computer science students from PUC-Rio who were learning SE for data science

Based on the goal, we established the following research questions for the validation in academia:

RQ1 *What is the relevance of each perspective of the candidate solution?*

We wanted to identify whether the perspectives of the candidate solution were perceived as meaningful and pertinent by the participants. This feedback helped confirm that the perspectives align with the needs and expectations of the intended users, and allowed us to identify areas that may need refinement.

RQ2 *Are the perspectives of the candidate solution and their concerns complete?*

This research question relates to the coverage of both the perspectives and concerns. This feedback helped to determine if critical components were missing or if there are gaps that need to be addressed.

RQ3 *To what extent does participants perceive the candidate solution as useful and beneficial?*

With this, we seek to understand the factors that influence the acceptance and adoption of the candidate solution. The question followed the technology acceptance model (TAM) (DAVIS, 1989) and aimed to capture participants’ overall assessment and intention to use the candidate solution, incorporating elements of perceived usefulness, perceived ease of use, and intended use.

RQ4 *What are the limitations and opportunities for improvement of the candidate solution?*

This research question seeks feedback on the approach itself.

5.4.3 Selection of Subjects

The subjects were the attendants of two SE for data science courses. The in-company course at Loggi had 53 professionals with different background being trained in SE practices for building ML-enabled systems. The graduate course at PUC-Rio had 15 students (nine master and six Ph.D students). While students may have limited expertise compared to professionals in the field, they can provide fresh perspectives, helping us identify potential blind spots. In fact, using students as subjects remains a valid simplification of real-life settings needed in laboratory contexts (FALESSI et al., 2018). In Table 5.15, we characterized the subjects by their educational background and average year of experience in ML projects.

Table 5.15: Subjects involved in the validation in academia.

Course	Total	Background	Experience (Average in years)
In-company	33	computer science	1.2
	20	other discipline	1.9
University	15	computer science	1.3

We can see that in the in-company course, not controlled by us, the professionals interested in data-driven projects are divided into those with a computer science background and those with background in other areas such as economics and mathematics. However, it is not surprising since the literature has already noted these findings for this role (KIM et al., 2017). Overall, the participants were perceived as relatively inexperienced, as they possess only a few years of practical experience in developing ML-enabled systems. While the participants were selected by convenience (attendants of the courses), we believe that their profiles were suitable for our intended initial validation.

5.4.4 Data Collection and Analysis Procedures

To address the research questions related to the relevance, completeness, perceived usefulness, and potential improvements of the candidate solution in specifying ML-enabled systems, a questionnaire-based evaluation method was

employed. This section outlines the data collection and analysis procedures used in the validation in academia.

Questionnaire Design: A follow-up questionnaire was designed to gather responses from participants regarding the research questions. The questionnaire included a combination of closed-ended questions related to *RQ1*, *RQ2* and *RQ3*, and one open-ended question related to *RQ4* to get both quantitative and qualitative data.

Data Collection: The questionnaire was delivered to the participants in online format for the in-company course and in-person session for the university course. Clear instructions were provided to guide participants through the specification task, which involved analyzing the candidate solution and completing a drafted template. This template included descriptions of each concern and perspective, along with corresponding spaces to specify concerns if applicable. Participants were also given detailed instructions on how to complete the follow-up questionnaire and for those who performed the study in person were provided with specific considerations to keep in mind while responding.

Quantitative Data Analysis: For *RQ1*, *RQ2*, and *RQ3*, which involve assessing relevance, completeness, and perceived usefulness, quantitative data analysis techniques were employed. Closed-ended questions were used to capture participants' ratings on a two-point likert scale for *RQ1* and *RQ2*, and four-point likert scale for *RQ3*. Statistical analysis, such as mean and frequency distribution were computed by the author of this thesis to summarize the quantitative data. At the end, three research collaborators reviewed the consolidated analysis.

Qualitative Data Analysis: For *RQ4*, which seeks to identify potential changes or additions to the candidate solution, qualitative data analysis techniques were utilized. Open-ended questions allowed participants to provide detailed and descriptive responses. Qualitative analysis followed a systematic procedure to extract meaningful themes from the data. Initially, the author of this thesis explored the raw data coming from the follow-up questionnaire, gaining an understanding of the participants' responses. After this, the same author performed initial coding, identifying recurring patterns, concepts, and insights within the data. Then, higher-order themes were generated by grouping related codes and identifying overarching concepts. Finally, through collaborative discussions involving three research collaborators, the identified themes were reviewed, refined, and validated.

Interpretation and Findings: The analysis of the collected data was interpreted according to the research questions. The findings were presented in a clear and concise manner, addressing each research question separately. In

this case, charts were used to illustrate the results, providing a comprehensive overview of the validation in academia.

5.4.5
Results

5.4.5.1
RQ1. What is the relevance of each perspective of the candidate solution?

This question was designed as a single choice question. To assess the relevance of each perspective of the candidate solution, participants were asked to rate the importance high or low. The perspectives considered in this evaluation included ML objectives, user experience, infrastructure, model, and data. The results indicated that all perspectives were deemed relevant by the participants. Out of a total of 68 participants, 67 considered the data perspective highly relevant, indicating its significant importance in specifying ML-enabled systems. The ML objectives, model and infrastructure perspectives followed closely, at 66, 65 and 63 respectively. The user experience perspective received a slightly lower number of 58, indicating its relatively high but somewhat lesser relevance. Figure 5.7 presents the relevance of the candidate solution’ perspectives based on their respective ratings.

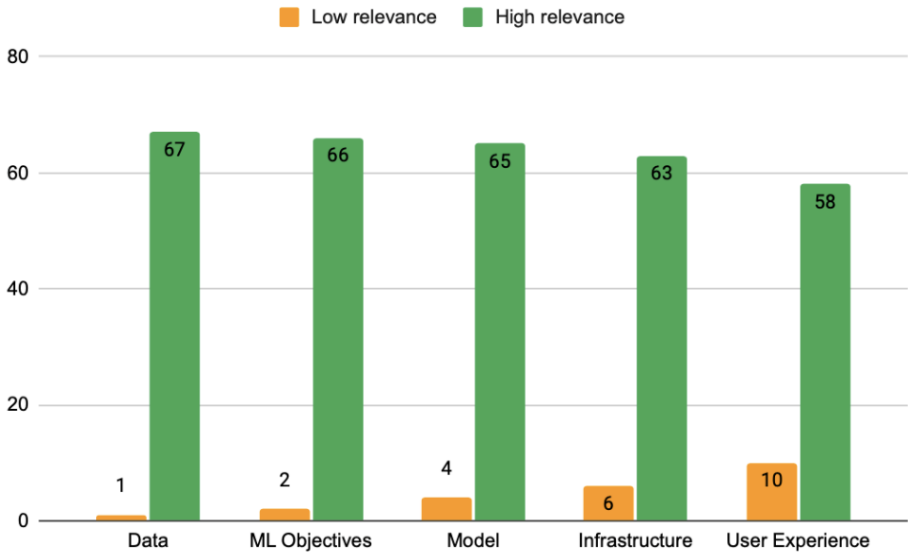


Figure 5.7: Frequencies of the relevance of each perspective of the candidate solution.

Somehow we expect these results, since typically the main focus of practitioners in ML projects is data and models. In contrast, user experience concerns take a back seat to the development of ML-enabled systems. That is

why with this work we seek to reinforce the importance of considering a user experience perspective.

5.4.5.2

RQ2. Are the perspectives of the candidate solution and their concerns complete?

This question was also designed as a single choice question with the option to explain the answer. To assess the completeness of perspectives and their associated concerns of the candidate solution, participants were provided with a list of predefined concerns corresponding to each perspective. They were then asked to indicate whether they believed the list was complete or if there were additional concerns that should be considered. The results revealed that participants generally considered the initial concerns and perspectives to be comprehensive but suggested some additional concerns. Only six out of 68 participants felt that something was missing. Across perspectives, the model perspective had the highest number of additional concerns identified by participants, highlighting the importance of monitoring ML models, optimizing parameters of ML algorithms, and breaking concepts about explainability. Below are the comments of the participants in that direction.

“There should be a monitoring concern related to the model view. In the same way we have to train the model, we have to monitor the model outputs”

“Parameter tuning in algorithms helps improve model performance. I would include this concern”

“Explainability could be divided into two: explainability and interpretability, given that there are explainable models that are not necessarily interpretable”

5.4.5.3

RQ3. To what extent does participants perceive the candidate solution as useful and beneficial?

To gauge participants' perception of the acceptance of the candidate solution for specifying ML-enabled systems, participants were asked to rate the solution on various aspects. These aspects included ease of use, usefulness and intended use. Ratings were provided on a scale of 1 to 4 (four-point likert scale), with 1 indicating strongly disagree, 2 indicating partially disagree, 3 indicating partially agree, and 4 indicating strongly agree. The TAM questionnaire results are shown in Figure 5.8.

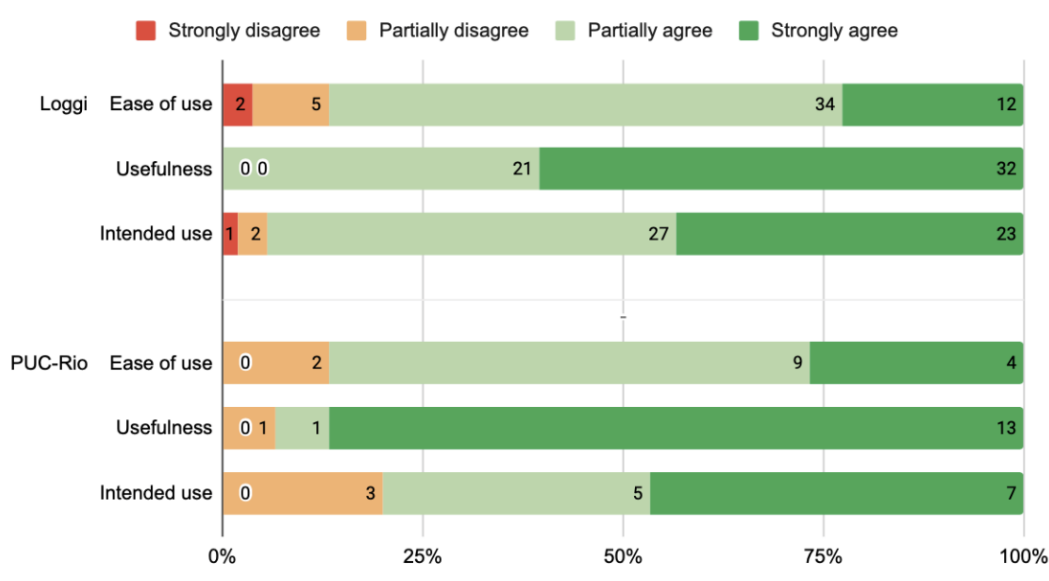


Figure 5.8: Frequencies of the TAM constructs for academic validation.

The responses indicated a positive perception of the candidate solution. Participants from both courses rated the solution highly in terms of usefulness, with an average rating of 3.7, suggesting that the candidate solution can support the specification of ML-enabled systems. The ease of use of the candidate solution received an average rating of 3.1, indicating that the candidate solution did not provide enough guidance to be considered clear. The intended use of the candidate solution was rated at an average of 3.3, reflecting its feasibility and applicability. Overall, the candidate solution was perceived as highly useful, but showed potential for improvement in terms of ease of use. We understood that improving the candidate solution's guidance will imply an improvement in the perception of intended use.

5.4.5.4

RQ4. What are the limitations and opportunities for improvement of the candidate solution?

Here, participants had the option to respond in open text format. To identify potential improvements in supporting practitioners in specifying ML-enabled systems, participants were asked to provide suggestions regarding components, perspectives, or concerns that could be changed or added to enhance the candidate solution. The analysis of participants' responses revealed several valuable suggestions. As identified in the results of *RQ3*, some participants emphasized the need to further integrate the relationship between concerns. Others highlighted the importance of incorporating a road-map to apply the candidate solution. Additionally, one participant recommended providing more practical examples and case studies to enhance the solution's applicability. In

the following, we present the comments of the participants in that direction.

“It would be interesting to connect more concerns because I clearly see some relationships. For example, in the model perspective the explainability concern depends, to some extent, on the selection of the algorithm”

“I would suggest explaining better how to use the approach because sometimes I did not know where to start and when to end”

“Definitely a practical example would help to better understand the proposal”

These results provided insights into the relevance of the perspectives, the completeness of the concerns, the perceived usefulness, and potential improvements, guiding the refinement of the candidate solution. The validation in academia resulted in the following improvement opportunities.

1. In the infrastructure perspective, we decided to include ‘**monitorability**’ as a new concern, since this may require implementing different services such as real-time logging, alerts, and data drift detection
2. In the model perspective, we broke the explainability concern into ‘**explainability and interpretability**’, since these terms can have different interpretations
3. We added ‘**algorithm parameter tuning**’ as a new concern of the model perspective, since data scientists typically need to analyze strategies to improve ML metrics
4. We defined a **set of steps** to be followed by stakeholders in order to apply the candidate solution

5.5 Static Validation

At this point, we made some improvements to the candidate solution, resulting in a version called *PerSpecML v1*. Building upon the foundation of the candidate solution, *PerSpecML v1* incorporates refinements and additions based on valuable feedback and insights from the students involved in the academic validation. In this section, we detail the second evaluation that was carried out in industry where practitioners had to use *PerSpecML v1* to retroactively specify two ready-made ML projects. We called this evaluation

as static since it was performed without executing *PerSpecML v1* in a real or simulated environment.

5.5.1 Context

The static validation in industry involved practitioners of the *ExACTa* initiative who developed two ML-enabled system projects from different domains for a large Brazilian oil company. The projects were developed following the Lean R&D approach (KALINOWSKI et al., 2020) and are already deployed in production in several oil refineries. We refer to these projects as project A and B, since for reasons of confidentiality and undergoing patent requests they cannot be explicitly mentioned. Table 5.16 details these projects.

Table 5.16: Projects involved in the static validation.

Project	ML domain	Description
A	Logistic regression	It alerts oil refineries about the likelihood of emitting strong odors that may result in claims from the community
B	Computer vision	It monitors images of the flame of oil refineries, helping refineries to decrease the disproportionate burning of gases that causes unnecessary energy consumption

We retroactively specified Project A and B using *PerSpecML v1* with the support of the product owner of each project, analyzing the perspectives and their concerns, and filling a drafted specification template. This means that the specifications were added after the project had already finished. Given the assistance provided by the author of this thesis during this task, the time spent was not strictly regulated and exceeded one hour. Subsequently, the practitioners who developed these projects were tasked with analyzing the resulting specifications and then they were interviewed in a focus group session, with each project allocated a two-hour time slot. The goal was to gain insights about the issues they face and the activities they perform in practice, and their perception of the resulting specifications. Lastly, we provided to practitioners with a follow-up questionnaire to gain more data about the evaluation of *PerSpecML v1*, including its limitations and opportunities for improvement. All mentioned artifacts are available in our online repository¹. Figure 5.9 shows the process diagram for the static validation in industry.

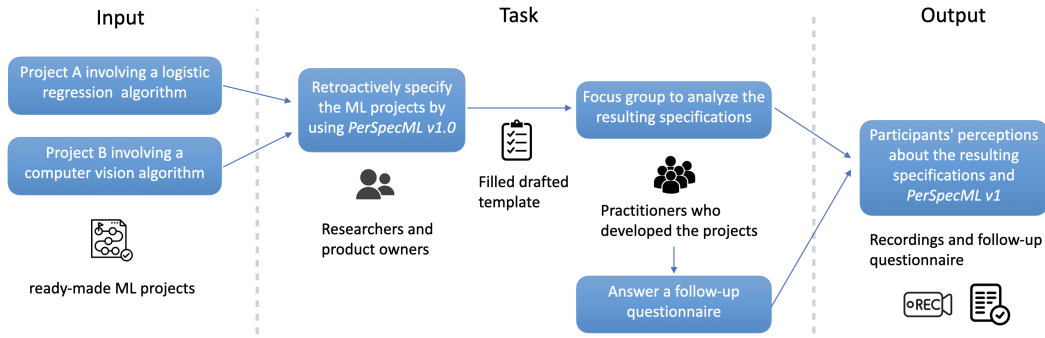


Figure 5.9: Process diagram for the static validation in industry.

5.5.2 Goal and Method

We detail the goal of the static validation in Table 5.17. We followed the GQM template to describe what we evaluated in this first industrial validation. Here, we also describe the research questions.

Table 5.17: Study goal definition of the static validation.

Analyze	<i>PerSpecML v1</i> (academically validated improved version) and its resulting specifications
for the purpose of	characterization
with respect to	perceived industrial relevance, ease of use, usefulness and intended use
from the viewpoint of	practitioners
in the context of	retroactively elaborated ML-enabled systems specifications using <i>PerSpecML v1</i> with six experienced software practitioners involved in the development of these systems

In contrast with the academic validation, involving practitioners with more experience ensures the evaluation reflects real-world scenarios and challenges. Their expertise can provide valuable insights into the practical applicability of *PerSpecML v1* and its alignment with industry standards and best practices. Based on the goal, we established the following research questions for the static validation in industry.

RQ1 *What problems do participants face in practice when specifying ML-enabled systems?*

We wanted to identify the challenges and difficulties encountered by participants when specifying ML-enabled systems. By understanding these problems, we analyzed the adherence to our solution, and identified the suitability of *PerSpecML v1* to cover the needs of practitioners.

RQ2 *What perception do the participants have of the retroactive specifications of projects A and B derived from PerSpecML v1?*

By answering this research question, we gathered insights about the benefits or detriments of using *PerSpecML v1*.

RQ3 *What are the limitations and opportunities for improvement of PerSpecML v1?*

With the feedback received, we refined *PerSpecML v1*.

RQ4 *To what extent do the participants perceive PerSpecML v1 as easy to use, useful and usable in the future?*

Through the TAM questionnaire, we explored the level of satisfaction and confidence participants had in *PerSpecML v1* as an approach for specifying ML-enabled systems.

5.5.3 Selection of Subjects

We invited six practitioners who have been actively working with the development of ML-enabled systems in the *ExACTa* initiative. This number of practitioners was determined based on the size of the project A and B. They were selected based on their position within the project, ensuring a comprehensive representation of the perspectives of our approach, and willingness to contribute, including only those who agreed to participate, ensuring a collaborative and engaged group. We asked them about their functions in the projects and their experience in years working with ML projects. Table 5.18 shows an overview of the participant characterization.

Table 5.18: Subjects involved in the static validation in industry.

Id	Role	Project	Experience (Years)
P1	Data scientist	A	6
P2		B	2
P3		B	2
P4	Developer	A	2
P5		B	3
P6	Project lead	A	2

It is possible to observe that in this study participants represent three different roles: data scientists who are interested in how the approach can help to build suitable and functional ML models, developers who are interested in

how the approach can help to design the integration between components, and project leaders who are interested in how the approach can help the team achieve its goals. This allowed to gather feedback from people who have different needs and priorities. On the other hand, participants showed have more than two years of experience, helping us determine whether *PerSpecML v1* would work well in practice and what could be improved. Note that we selected three practitioners of each project involved in the evaluation.

5.5.4 Data Collection and Analysis Procedures

To address the research questions, a combination of focus group discussions and questionnaires were employed for data collection. In the following, we outline the data collection and analysis procedures used in the static validation in industry.

5.5.4.1 Focus Group

We conducted a focus group for promoting in-depth discussion on *RQ1* and *RQ2* (KONTIO; LEHTOLA; BRAGGE, 2004). Focus group is a qualitative research method that involves gathering a group of people together to discuss a particular topic, allowing for interaction between the participants, which can help to surface different viewpoints. We based the discussion on the specification task, which involved retroactively specifying the projects with the support of the product owners by using *PerSpecML v1* and completing a drafted template that included descriptions of each concern and perspective, along with corresponding spaces to specify concerns if applicable.

Procedure. The focus group was conducted in a structured and moderated format. The discussions were guided by the first author using open-ended questions related to *RQ1* and *RQ2*, allowing participants to share their experiences, perspectives, and challenges faced when specifying ML-enabled systems.

Data Collection. We recorded the focus group with the consent of the participants to gather qualitative data. Transcripts of the focus group discussions were generated by the author of this thesis from the recordings, capturing participants' insights, ideas, and suggestions regarding *RQ1* and *RQ2*.

Data Analysis. Thematic analysis was employed to identify common themes, patterns, and recurring topics in the focus group data (SERVICE, 2009). The transcripts were coded, and emerging themes were categorized with the consensus of the three research collaborators. Lastly, the final set of categories was analyzed by three research collaborators to address the

research questions. The transcriptions and all codes are available in our online repository¹. Examples of codes are highlighted when presenting the results.

5.5.4.2

Questionnaire

Questionnaire design: The questionnaire included structured questions and rating scales designed to capture quantitative and qualitative data related to *RQ3* and *RQ4*, respectively. It addressed perceptions and feedback regarding the problems faced, usefulness of *PerSpecML v1*, ease of use, and identified limitations or opportunities for improvement.

Data Collection: The questionnaire responses were collected electronically through an online survey platform, taking care of anonymity and confidentiality. We provided the participants with clear definitions of the quality characteristics that we wanted to measure, ensuring that the participants understood what was asked of them.

Data Analysis: Quantitative data analysis techniques, such as descriptive statistics and inferential analysis, were used to analyze the questionnaire responses related to *RQ4*. These findings provided numerical insights and trends, allowing for a comprehensive understanding of participants' perceptions about the acceptance of *PerSpecML v1*. Qualitative data analysis techniques were also used to respond *RQ3*, involving coding and categorization. Here, we used the same procedures applied in the academic validation. For instance, we explored data, created initial codes and then reviewed them to report our findings to participants.

5.5.5

Results

5.5.5.1

RQ1. What problems do participants face in practice when specifying ML-enabled systems?

We asked the participants about the problems they face when specifying ML-enabled systems. We coded and categorized the transcriptions of such discussions and then analyzed them to answer this research question. We found that participants frequently mentioned *lack of approaches to support the specification* given that ML incorporates additional challenges, which can make it difficult to specify ML-enabled systems. For instance, P6 stressed:

“To the best of my knowledge there are no tools or approaches spread in industry helping practitioners to elicit, specify and validate requirements for ML systems”

In the same line, P4 and P5 complemented:

“I’m curious to see a formal specification of an ML component. Based on my experience, these definitions are informal and emerge as the project progresses”

“Sometimes I feel that the ML development team often transmits skepticism to customers, not because of the lack of knowledge of its members, but because of the lack of an established process to define what can be done in ML terms with what the customer makes available (*e.g.*, data, business information)”

On the other hand, we identified expressions about specification problems derived from the *need to involve domain experts*. For instance, P1 reported that understanding the specific domain plays a major role for accurate specifications:

“Typically domain experts are busy, so they tend to be less involved in the early phases of ML projects. In the end, they often find unexpected results. Their involvement is important in areas such as feature engineering, data pre-processing and model evaluation”

P4 highlighted that customers often overestimate what ML can do. This leads to *unrealistic expectations of ML capabilities*, posing challenges in the specification process. The participant expressed:

“Most of the time, customers expect that ML systems can solve all problems. They also don’t imagine the number of components that are required to operate and maintain an ML model over time. Requirements engineering could help to address these challenges”

These findings reflect some of the problems faced by participants in practice when specifying ML-enabled systems, as identified through the focus group discussions with experienced practitioners. The insights gained from these discussions shed light on the key areas that require attention to overcome challenges such as *the lack of approaches to support the specification, the need to involve domain experts, and the customer unrealistic expectations of ML capabilities*

5.5.5.2

RQ2. What perception do the participants have of the retroactive specifications of projects A and B derived from *PerSpecML v1*?

After the participants analyzed the resulting specifications for Project A and B derived from *PerSpecML v1*, we asked them what they thought about it. Their feedback indicated positive perceptions of the specifications and their future impact on the development process. For instance, the participants highlighted that the specifications acted as a *guide during the development process*, helping to improve the overall development workflow. P1 manifested:

“Looking at the diagram and its corresponding specifications allowed me to get an early overview of the requirements that can be refined as the project progresses. It is like a high-level guided development”

P1, P3 and P6 expressed that the retroactive specifications *enhanced clarity and understanding* of the ML-enabled systems for both projects:

“I found that the specifications facilitated a better understanding of the systems’ functionality, components, and data requirements, specially for Project A, in which I was involved”

“I really liked the focus on diverse aspects such as data, model, and infrastructure. This landscape facilitates the understanding of the projects”

“Identifying the tasks and concerns and their relationships allows identifying dependencies and influences as intended”

In addition, P3 mentioned that using *PerSpecML v1* allowed to *identify hidden concerns* that are not easily identified at first sight:

“Typically, user experience concerns are put in the background. With *PerSpecML* was possible to early specify forcefulness, a concern analyzed late in the validation phase of Project B”

Finally, P5 noted that the retroactive specifications derived from *PerSpecML v1* helped in *documenting and communicating* the ML-enabled systems for both projects:

“In my opinion, it is easy to convey the specifications to stakeholders, enabling better collaboration and alignment throughout the development process. For example, as a developer I can identify tasks where I need to collaborate with data scientists”

Overall, there was a clear consensus on the benefits of the retroactive specifications of Project A and B, derived from *PerSpecML v1*. According to the

participants, the specifications *enhanced clarity and understanding, improved documentation and communication, acted as guide during the development process, and identified hidden concerns.*

5.5.5.3

RQ3. What are the limitations and opportunities for improvement of *PerSpecML v1*?

Participants' feedback revealed several limitations and opportunities for improvement. These insights, derived from the open-ended question of the questionnaire, can be related to the findings of *RQ4*, where we had participants who expressed partial agreement and disagreement about ease of use, usefulness, and intended use. For instance, P1 and P2 suggested that *providing additional guidance* could help users grasp *PerSpecML v1* more easily.

"It is not clear to me how to get the specifications from analyzing the diagram. Even with the provided steps to apply the solution, it is not clear to me"

"Providing tutorials or additional documentation could improve its application"

Participants also provided feedback on *improving the user interface* of *PerSpecML v1*, suggesting a more user-friendly design.

"In my opinion, the specification template, which summarizes what the system should do, should be cleaner. I mean, the relationships between concerns are not needed as they exist in the diagram"

"Better visualizations and intuitive navigation could further enhance the user experience and ease of use"

On the other hand, P6 commented on *improving the relationship between tasks and concerns*. More specifically, the participant suggested breaking down a task of the ML objective perspective, since the concerns were not related at all.

"In the ML objective perspective there is something that does not make sense. The 'define objectives' task has independent concerns that could be part of separate tasks"

We identified limitations and opportunities for improvement of *PerSpecML v1* related to *providing additional guidance, improving the user interface, and improving the relationship between tasks and concerns*. Some of

them may be related with the participants' perceptions explored in *RQ4*. We addressed these limitations and capitalized on the opportunities for improvement, allowing to refine *PerSpecML v1* to better meet the needs and challenges identified by practitioners.

5.5.5.4

RQ4. To what extent do the participants perceive *PerSpecML v1* as easy to use, useful and usable in the future?

The participants' responses to a TAM questionnaire indicated varying degrees of agreement or disagreement with statements about ease of use, usefulness, and intended use. While the majority of participants totally agreed with the statements, there were a few participants who expressed partial agreement or disagreement. More specifically, one participant encountered some difficulties in using *PerSpecML v1*, two participants had reservations about its usefulness, and one participant was not fully confident in using it in the future. The TAM questionnaire results are shown in Figure 5.10.

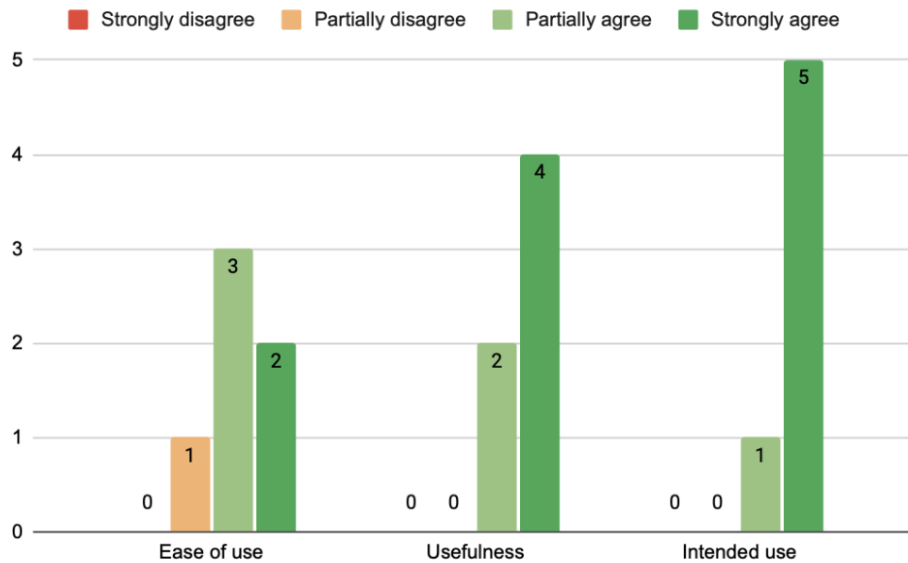


Figure 5.10: Frequencies of the TAM constructs for static validation industry.

These varied perceptions explained to some extent the feedback received in *RQ3* for identifying areas of improvement and addressing any concerns or challenges raised by participants. At the end of this validation, we decided to consider the feedback of the practitioners of the *ExACTa* initiative. In the following, we outline what was incorporated into *PerSpecML v1* from this static validation in industry.

5. We added the **domain expert role** to the *PerSpecML v1*' stakeholders, including it in tasks
6. The steps defined in the academic validation to apply *PerSpecML v1* turned into a **workflow diagram** to facilitate its application
7. We improved the *PerSpecML v1* documentation by creating a **Miro board** that summarizes the perspectives, tasks and concerns to be analyzed. We also added a **practical use case** and **explanations** of each *PerSpecML* component
8. We improved the user interface of both diagram and specification template by adding **colors** that identify each perspective
9. We simplified the specification template by **removing the representation of the relationships between concerns** (leaving them only in the perspective-based ML task and concern diagram, as they are used during the analysis)
10. We checked **terminology** and the **relationship between tasks and concerns** of each perspective to ensure its suitability

5.6

Dynamic Validation

Based on the valuable feedback and insights from the practitioners involved in the static validation, we made significant improvements to *PerSpecML v1*, resulting in a more robust and enhanced version called *PerSpecML v2* that served as the foundation for the subsequent validation conducted in this study. In this section, we evaluated *PerSpecML v2* by performing (i) requirement workshop sessions and (ii) interviews with practitioners who work for a large Brazilian e-commerce company known as Americanas that offers technology, logistics, and consumer financing services. We called this validation as dynamic, since it was performed by applying *PerSpecML v2* for specifying two real industrial ML projects.

5.6.1

Context

We performed the dynamic validation through two distinct case studies at Americanas, with each case involving the specification of a real ML-enabled system. These systems were purposefully crafted from scratch to enhance and optimize various facets of the company's business processes. Notably, due to

the absence of a formal method for specifying such systems within the e-commerce company, the utilization of *PerSpecML v2* was authorized, providing an opportunity to showcase its practical application. Each system was assigned a team comprising both novice and experienced practitioners. A description of the ML-enabled systems involved in this context is outlined in Table 5.19.

Table 5.19: ML-enabled systems involved in the dynamic validation.

System	ML domain	Description
Product Classification	Natural Language Processing	It classifies titles of products registered by sellers in the marketplace of the Americanas company into categories. Based on the correct category, basic attributes for registering the product details are then provided to the seller
Market	Recommendation System	It suggests products to customers that are likely to be of interest or relevance to them. Based on historical data and similarity measures, the products are recommended

Regarding the operation of the case studies, we assisted practitioners in the application of *PerSpecML v2* in requirements workshop sessions by providing the necessary materials and information in advance. This encompassed comprehensive documentation on *PerSpecML v2* along with illustrative use cases. Throughout these sessions, practitioners from each project collaboratively engaged in the analysis and specification of the ML-enabled systems using *PerSpecML v2*. The specifications were dynamically compiled by incorporating post-it notes into the interactive Miro board, a template initially crafted during the static validation. Subsequently, we conducted two additional sessions for interviews, engaging with experienced practitioners from each project who have knowledge of the domain problem and who have led the design and implementation of other ML-enabled systems within the company. These sessions focused on in-depth discussions about the resulting specifications. Finally, we distributed a follow-up questionnaire to all practitioners to critically evaluate *PerSpecML v2* and the specifications it generated. All mentioned artifacts are available in our online repository¹. Figure 5.11 shows the process diagram for the dynamic validation in industry.

5.6.2 Goal and Method

We detail the goal of the case studies of the dynamic validation in Table 5.20. We followed the GQM template to describe what we evaluated in

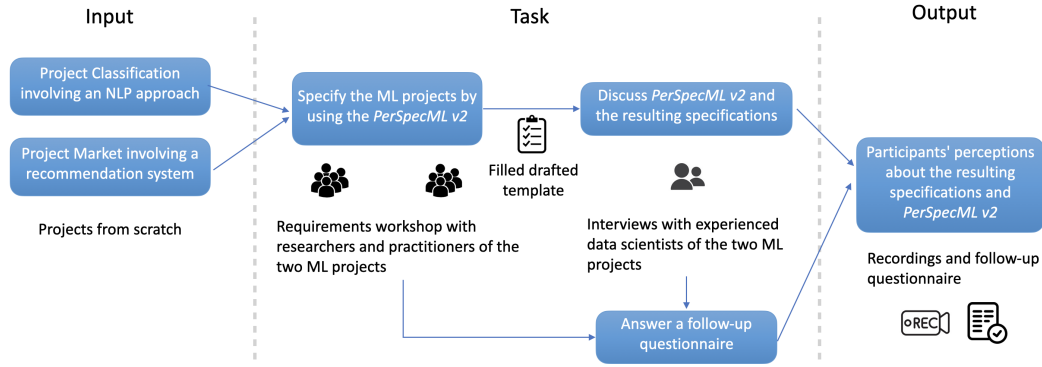


Figure 5.11: Process diagram for the dynamic validation in industry.

this second industrial validation. Here, we also describe the research questions.

Table 5.20: Study goal definition of the dynamic validation.

Analyze	<i>PerSpecML v2</i> (statically validated improved version) and its resulting specifications
for the purpose of	characterization
with respect to	the perceived quality of the specifications derived from <i>PerSpecML v2</i> , and ease of use, usefulness and intended use of <i>PerSpecML v2</i>
from the viewpoint of	practitioners
in the context of	two requirements workshop sessions involving 11 novice practitioners and three experienced practitioners who used <i>PerSpecML v2</i> to specify two ML projects from scratch, and two interviews with the three experienced practitioners who evaluated the resulting specifications derived from <i>PerSpecML v2</i>

Based on the presented goal, aligned to the purpose of a dynamic industrial validation, we defined the following research question to better understand the practical suitability of using *PerSpecML v2*.

RQ1 *What perception do practitioners have while specifying ML-enabled systems by using PerSpecML v2?*

For this research question, we conducted a comprehensive evaluation of practitioners' experiences while specifying ML-enabled systems using *PerSpecML v2*. During the requirements workshop sessions, we observed their interactions with *PerSpecML v2*, noted any challenges or difficulties they encountered, and gathered their feedback through discussions and direct feedback.

RQ2 *What perception do experienced practitioners have of the resulting specifications derived from PerSpecML v2?*

To answer this question, we interviewed three experienced practitioners who reviewed and discussed the specifications derived from *PerSpecML v2*. We selected them since experienced practitioners can better assess the efficiency and effectiveness of *PerSpecML v2* than novice, for instance, by comparing it to existing methods they have used in the past. During the interview, the experienced practitioners provided their feedback on the specifications. The goal was to gather valuable insights into how the experienced practitioners perceived the quality, completeness, and suitability of the specifications produced by using *PerSpecML v2*.

RQ3 *What are the limitations and opportunities for improvement of PerSpecML v2?*

To explore this research question, we considered the feedback and discussions from both the novice and experienced practitioners. The novice practitioners' firsthand experience with using *PerSpecML v2* shed light on challenges, difficulties, and limitations they encountered while applying the approach. Additionally, the insights provided by the experienced practitioners allowed us to identify areas for improvement and potential enhancements. With the feedback received, we further refined *PerSpecML v2* and came up to its final version.

RQ4 *To what extent do the practitioners perceive PerSpecML v2 as easy to use, useful and usable in the future?*

To address this research question, we provided to participants a follow-up questionnaire. We collected feedback from both novice and experienced practitioners regarding their perception of *PerSpecML v2* as an approach for specifying ML-enabled systems. The novice practitioners, who used *PerSpecML v2* during the requirements workshop session, provided their insights on the ease of use, usefulness, and usability of the approach. Additionally, the experienced practitioners shared their opinions on the practicality and potential future utility of *PerSpecML v2*. By analyzing their feedback, we gained a comprehensive understanding of how *PerSpecML v2* was perceived by practitioners across different experience levels.

5.6.3 Selection of Subjects

The dynamic validation involved two main groups of participants from Americanas: novice practitioners who specified two ML-enabled systems from scratch using *PerSpecML v2*, and experienced practitioners who also specified the systems, and additionally evaluated the resulting specifications. The profile of these practitioners was defined by the e-commerce company from the beginning of the ML projects since the scope of the projects involved training practitioners with limited ML experience, guided by ML experts who assumed leadership roles in the projects. In contrast to static validation, our dynamic validation adapts to the participants rather than following a predefined selection process.

The practitioners involved in this validation were characterized by having varied backgrounds, such as computer science, mathematics, physics, and others. The diversity in their educational background and experience helped validate the maturity of *PerSpecML v2*. Their feedback shed light on its suitability for real-world implementation and if it meets the expectations and requirements of industry professionals. In Table 5.21, we characterized the subjects by their role in the development of the ML-enabled systems involved in this study, educational background, and years of experience involved in ML projects.

Table 5.21: Subjects involved in the dynamic validation in industry.

Team	Id	Role	Background	Experience (Years)
Team A	P1	Developer	Computer science	1
	P2		Design	1
	P3		Computer science	1.5
	P4		Computer engineering	1
	P5	Scrum master	Physics	1.5
	P6	Data scientist	Computer science	1
	P7	Data scientist	Linguistic	8
Team B	P8	Developer	Electronic engineering	1
	P9		Computer engineering	1
	P10		Computer science	1
	P11		Mathematics	1
	P12	Scrum master	Computer science	2
	P13	Data scientist	Electrical engineering	4
	P14	Data scientist	Computer science	6

The subjects involved in specifying the ML-enabled systems from scratch were part of the two project teams. The allocation of participants was deter-

mined based on their roles in the ML projects. In the first one, which we call team A, we had six novice practitioners and one experienced practitioner responsible for *Product classification* system. In the second team that we call B, we had five novice practitioners and two experienced practitioners responsible for *Market* system. We highlighted the experienced practitioners who led each team with grey color in order to differentiate them from novice. Note that experienced practitioners are data scientists with a different educational background than computer science or engineering (except for P14), as expected for these positions (KIM et al., 2017; AHO et al., 2020).

5.6.4

Data Collection and Analysis Procedures

To address the research questions outlined in this dynamic validation, we employed three main data collection procedures: requirements workshop sessions, interviews, and a follow-up questionnaire.

5.6.4.1

Requirements Workshop Sessions

Workshop Design. We designed the requirements workshop sessions with a clear agenda and objectives, and outlined the tasks that the participants performed during the workshop, such as using *PerSpecML v2* to specify the two ML-enabled systems from scratch. This allowed to provide the input to respond to *RQ1*.

Data Collection. During the sessions, we collected data in the form of written specifications produced by the practitioners. These specifications included concerns on the five perspectives such as objectives, user experience, infrastructure, model, and data. Additionally, to ensure a comprehensive record of the specification task, we obtained explicit permission from all practitioners to record the sessions. These recorded sessions serve as valuable supplementary resources, allowing for a detailed review of the collaborative analysis and specification process and ensuring accuracy and completeness in our data collection.

Data Analysis: The author of this thesis and his advisor analyzed the recorded workshop sessions. Subsequently, they systematically extracted pertinent statements from the participants, focusing on their interactions with other participants during the workshop and their engagement with *PerSpecML v2*. This process was particularly significant for triangulating this data with information obtained through other research methods, ensuring a comprehensive

and multifaceted understanding of the dynamics and insights emerging from the workshop sessions.

Reporting: We summarized the findings and insights from the workshop sessions in a structured manner by including direct quotes and paraphrased statements from the practitioners to support the analysis and interpretations.

5.6.5

Interviews

Interview Design. We developed a semi-structured interview protocol for *RQ1*. The protocol included a set of open-ended questions that focus on the experienced practitioners' perception of the resulting specifications derived from *PerSpecML v2*. Questions explored aspects such as the quality, completeness, clarity, and effectiveness of the specifications. This shed light on answering *RQ1*.

Data Collection. We conducted interviews with experienced practitioners. During the interviews, we used the protocol to guide the discussions while allowing practitioners to share their thoughts and insights freely. We recorded the interviews in video format, with their consent, in order to ensure accurate capture of responses and allow for later review and analysis.

Data Analysis. The author of this thesis transcribed the video recordings of the interviews into text format in order to analyze the participants' responses, and then the same author applied coding techniques to categorize them into themes. In order to validate these themes, three research collaborators discussed and refined them before presenting our findings to the participants. In addition, we triangulated the analysis by comparing and cross-referencing the results from the different interviewees.

Reporting. We summarized the findings and insights from the interviews in a structured manner by including direct quotes and paraphrased statements from the practitioners to support the analysis and interpretations.

5.6.6

Questionnaire

Questionnaire design. The questionnaire included structured questions and rating scales designed to capture quantitative and qualitative data related to *RQ2* and *RQ3*, respectively. It addressed perceptions and feedback regarding the usefulness and ease of use of *PerSpecML v2*, and identified limitations or opportunities for improvement.

Data Collection. The questionnaire responses were collected electronically through an online survey platform, taking care of anonymity and confi-

dentiality. Participants were assured that their responses would be kept confidential, and all personal information was carefully protected and anonymized, ensuring that individual responses could not be linked back to specific participants. This approach was implemented to encourage participants to express their views without concerns about privacy.

Data Analysis. Quantitative data analysis techniques, such as descriptive statistics and inferential analysis, were used to analyze the questionnaire responses related to *RQ2*. These findings provided numerical insights and trends, allowing for a comprehensive understanding of participants' perceptions about the acceptance of *PerSpecML v2*. Qualitative data analysis techniques were also used to respond *RQ3*, involving coding and categorization. Here, we used the same procedures applied in the academic and static validation. For instance, we explored data, created initial codes and then reviewed them to report our findings to participants.

5.6.7 Results

5.6.7.1

RQ1. What perception do practitioners have while specifying ML-enabled systems by using *PerSpecML v2*?

During the workshop specification sessions, we observed the interactions of practitioners with *PerSpecML v2* to identify benefits or difficulties they encountered. The comments and discussions indicated that practitioners had a generally positive perception of *PerSpecML v2* as a supportive tool for guiding them through the specification process. For instance, novice practitioners P3 and P5 appreciated *the visual and intuitive interface of PerSpecML v2*:

“At first sight, I was able to identify each perspective, its tasks, and their concerns. This helps me to better understand the requirements and dependencies of the *Product Classification* system”

“I find the specification template and language constructs within *PerSpecML* beneficial in structuring the specifications effectively”

As the workshops progressed, practitioners recognized the *PerSpecML v2*'s role in *early identification and resolution of potential concerns* in ML projects, and its *ability to facilitate collaboration and communication* among different teams involved in ML projects. P11, P13, P1 and P3 expressed:

“Many times in our projects some of these concerns are only addressed when it is clearly too late. I see the diagram as a roadmap that allows me to identify components that would not be identified without its use”

“There are several tasks that at the beginning of the project do not concern our team, but that deserve to be analyzed for their relationships with others”

“*PerSpecML* summarizes the work of several ML teams in one diagram”

“Linking the model update task in the infrastructure perspective with the need to get user feedback in the user experience perspective makes sense. This encourages communication between teams involved in ML projects”

While some initial learning curve was observed, practitioners quickly grasped *PerSpecML v2*’s functionalities and became comfortable using the approach. Their perception of usability and effectiveness improved as they gained more hands-on experience during the workshop sessions. *RQ3* gave us more insights in this line.

5.6.7.2

RQ2. What perception do experienced practitioners have of the resulting specifications derived from *PerSpecML v2*?

The experienced practitioners expressed positive feedback regarding the resulting specifications derived from *PerSpecML v2* for the two ML projects. For instance, P13 and P14 appreciated the *clear and well-structured nature of the specifications*, and the *utility for specific users*:

“The specifications demonstrated a good understanding of the ML projects’ requirements, guiding the novice practitioners through the specification process”

“The diagram can be extremely helpful for novice data scientists or engineers to get an overview of the ML workflow”

However, P7 pointed out minor areas where specifications could be further refined to better align with specific project needs:

“I am not sure if at the end the specifications are already sufficiently clear, but I can state what has been raised is reasonable and useful. Coming up with a clear specification requires refinements and increments”

Indeed, the requirements workshop was supposed to be the first effort towards comprehensive specifications that should be further improved after the workshop. On the other hand, P7 and P14 (experienced practitioners from

separate workshops) both compared *PerSpecML v2* with the approach they used so far in their projects.

“*PerSpecML* provides a more comprehensive overview and is far better than the ML canvas to support specifying ML-enabled systems”

“Currently, we use *ML canvas* to describe ML systems, but *PerSpecML* covers more elements, and helps analyze their relationships”

Overall, the experienced practitioners were impressed with the novice practitioners’ efforts and saw *PerSpecML v2* as a valuable tool for fostering collaboration and understanding between different skill levels within the team.

5.6.7.3

RQ3. What are the limitations and opportunities for improvement of *PerSpecML v2*?

The open-ended responses in the follow-up questionnaire provided valuable insights into the limitations and opportunities for improvement of *PerSpecML v2*. For instance, P7 suggested adding a concern related to the *financial cost* associated with the infrastructure that is required to operate an ML-enabled system, while P3 recommended paying attention to the *versioning of libraries*.

“Based on my experience, ML systems can be expensive to maintain. Even large companies should carefully consider the costs of maintaining ML systems before implementing them. I would include this concern for sure”

“It is important to consider the versioning of the libraries that are typically used in the development of ML-enabled systems. On several occasions I have seen my teammates in trouble, for example, when the Python version is not compatible with the TensorFlow version. If there is a proper version management this could be avoided”

Moreover, P13 suggested complementing the model perspective with the phenomenon that occurs when the performance of ML models decreases over time, and that both data scientists and customers typically pass up.

“Requirements specifications captures what the system is supposed to do, right? ML models tend to degrade over time due to several factors such as environmental and data changes. This behavior is typically not considered, therefore, it should be specified”

On the other hand, P12 added another interesting opportunity for improvement: classifying the concerns by importance to better cope with the number of concerns to be analyzed.

“When analyzing the diagram I see that the number of concerns is considerable. That’s not a bad, in fact, it shows everything to think when designing ML systems. For this reason, I think it would be interesting to classify each concern by its importance. This would somehow prioritize the specification process”

Finally, P14 mentioned the importance of automating *PerSpecML v2*:

“It would be good to automate the approach by decreasing human involvement in the execution of *PerSpecML* that are prone to errors. It is a matter of practicality. In short, you can automate the *PerSpecML*’ logical flow”

Overall, the feedback indicated that *PerSpecML v2* had potential for enhancement, and practitioners were eager to see future updates and features that could further elevate the tool’s usability and effectiveness.

5.6.7.4

RQ4. To what extent do the practitioners perceive *PerSpecML v2* as easy to use, useful and usable in the future?

Based on the TAM questionnaire that included four-point Likert scale ratings, we found that practitioners indicated a high level of acceptance and positive perception of *PerSpecML v2*. The summary of the responses is shown in Figure 5.12.

The majority of participants rated *PerSpecML v2* as easy to use, with a significant portion (12 out of 14) giving it a rating of 4 (strongly agree). The documentation, intuitive interface and clear instructions provided by *PerSpecML v2*—improvements that came up in static validation—contributed to its perceived ease of use, making it accessible and user-friendly for both novice and experienced practitioners. However, one participant expressed partial disagreement with the statement of ease of use. This response came from P14, an experienced data scientist who mentioned suggestions for improvements on this topic in the previous question.

Additionally, the practitioners found *PerSpecML v2* to be highly useful in the specification process. Excluding one who expressed partial agreement, all the participants gave it a rating of 4 for usefulness (strongly agree). Indeed, the discussions and the outputs of the workshop sessions showed that *PerSpecML v2* was especially valuable in guiding practitioners through the specification process and enhancing the overall clarity of the specifications.

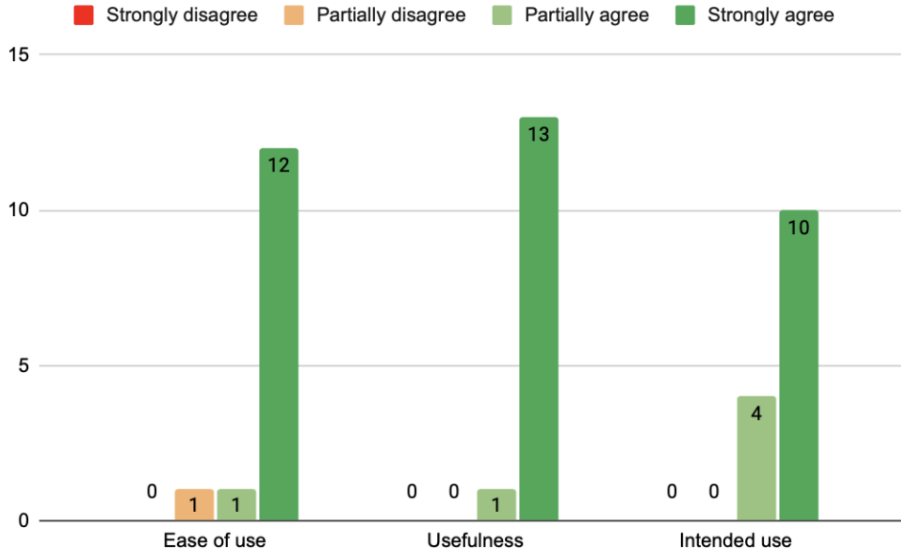


Figure 5.12: Frequencies of the TAM constructs for dynamic validation in industry.

Furthermore, the practitioners showed positive attitudes towards *PerSpecML v2*'s intended use. The majority of respondents (10 out of 14) expressed that they would be willing to use *PerSpecML v2* in future ML projects, indicating the approach's potential to become an essential part of their workflow for specifying ML-enabled systems.

Overall, the questionnaire results demonstrated a strong acceptance and positive perception of *PerSpecML v2*'s ease of use, usefulness, and future usability among the practitioners. When comparing these results with the static validation, we saw that the perception of ease of use improved considerably, indicating that the improvements from that evaluation had an effect.

At the end of this validation, we decided to consider the feedback of the practitioners of the Americanas company. In the following, we outline what was incorporated into *PerSpecML v2* from this dynamic validation in industry, which led to the final version of *PerSpecML*.

11. We added ‘**financial cost**’ as a new concern of the infrastructure perspective, since ML typically demand implementing several services that impact project budget
12. We added ‘**versioning**’ as a new concern of the model perspective, since this is essential for reproducibility, compatibility, and long-term maintainability of ML models
13. We added ‘**performance degradation**’ as a new concern of the model perspective, since it can lead to inaccurate predictions, which can cause problems for businesses and organizations
14. Based on a meta-review of the validations, we included ‘**education & training**’ in the user experience perspective, and ‘**anonymization**’ in the data perspective. The first new concern will help that users have a clear understanding of the ML model’s capabilities and potential inaccuracies ensure the system’s credibility and user satisfaction, and the second one will help to protect sensitive data when required while still maintaining the utility of the data for ML purposes
15. We refined the *PerSpecML v2*’ **logical flow** to explicitly include the relevance of the concerns into desirable, important or essential. This could help to prioritize the requirements of ML-enabled systems

5.7

Threats to Validity

Assessing the validity of study results is particularly important for ensuring the accuracy, reliability, and generalization of findings. In this study, we empirically evaluated *PerSpecML* by analyzing practitioners’ perceptions and experiences. In the following, we discuss potential limitations and challenges that could impact the trustworthiness and applicability of our research outcomes. To this end, we followed the categories suggested by (WOHLIN et al., 2012).

Construct validity. For our quantitative and qualitative analyses, we conducted a mix of data collection methods, such as the TAM questionnaire, focus groups, and interviews. These choices were based on the well-established theoretical foundation of such methods. For instance, the TAM model has been widely used in technology acceptance research (TURNER et al., 2010), and its questions were carefully designed to measure specific constructs related to the users’ attitudes and intentions towards adopting our approach. To gain insights

from these data collection methods, we used thematic analysis, a widely used qualitative research method for identifying, analyzing, and reporting patterns.

Internal validity. In the static validation, the practitioners' familiarity with the ML projects that were retroactively specified may have influenced their perception and performance during the validation process, leading to potential bias in the results. To mitigate this threat, we decided to retroactively specify the ML projects with the support of the product owner of each project, but without involving the practitioners. In this case, we wanted to take advantage of this situation since by knowing the ML projects, the practitioners could more easily evaluate the resulting specifications, *e.g.*, whether important aspects was missing.

External validity. We are aware that the generalization of the findings from the academic and static validation to real-world industrial scenarios may be limited. For instance, the toy scenario used in the academic setting and the specifications built retroactively in the static validation may not fully capture the complexity and challenges faced in actual industrial projects. Our intention with these artifacts was to use them to iteratively improve *PerSpecML* until it was mature and could be evaluated in a more realistic setting. Regarding the subject representativeness, the validation conducted in academia with students was a deliberate initial step in the evaluation process, serving as a foundational phase in the research. According to (FALESSI et al., 2018), using students as subjects is a valid simplification commonly needed in laboratory contexts.

Furthermore, the R&D initiative may also not represent a typical industrial setting. We recognize potential differences between practitioners in such settings and those in a more typical industrial environment. R&D settings often exhibit characteristics that align closely with academia, fostering an environment where practitioners may prioritize exploration, experimentation, and innovation. However, the R&D initiative involved in our study, closely works with industry partners from different domains such as energy & oil, and retail. Given this, practitioners from the R&D initiative were also actively engaged in practical hands-on work that involves the development of novel solutions for real industrial contexts.

We believe that including participants from different context constitutes a diverse setting that allowed for the examination of *PerSpecML* across different scenarios, thereby strengthening the generalization of the findings.

Conclusion validity. During the data collection and analysis procedures of the three evaluations, a single researcher conducted the thematic analyses. To mitigate this threat, three research collaborators reviewed and discussed the list of codes attached to the transcriptions. Two research collaborators

brought diverse expertise in SE, system architecture, and qualitative research methods, while another one brought expertise in data science. This helps to ensure a comprehensive and multidimensional analysis. In addition, as suggested by (KONTIO; LEHTOLA; BRAGGE, 2004), we presented the findings to a subset of participants from the academic, static and dynamic validation to review and provide feedback on the identified themes, ensuring that their perspectives were accurately represented. Moreover, we triangulated both qualitative and quantitative data by comparing findings from the focus group discussions with insights obtained from the follow-up questionnaire. This helped provide a more robust understanding of *PerSpecML*'s usability and effectiveness, supporting well-informed conclusions.

5.8 Discussion

In this section, we reflect on the outcomes of the validations and how they contribute to the understanding and improvement of *PerSpecML*, our perspective-based approach for identifying concerns when specifying ML-enabled systems. We explore the broader implications of the findings, other areas of study, and how our approach can positively impact the development of ML-enabled systems.

In terms of **rigor**, *PerSpecML* is the result of a series of validations that were conducted in different contexts, each contributing valuable insights and refining our approach to meet the diverse needs of practitioners involved in ML projects. Through careful evaluations encompassing academia and industry, *PerSpecML* has undergone iterative enhancements, ensuring its effectiveness and adaptability in guiding the specification of ML-enabled systems across various scenarios and project complexities. The combination of student validation, real-world discussions with experienced data scientists, and collaborative evaluations with both novice and experienced practitioners has culminated in a robust and user-friendly approach that empowers teams to collaboratively and comprehensively define ML-enabled systems from inception to completion.

In terms of **scope and coverage**, *PerSpecML* was designed with the underlying assumption that the problem to be solved can benefit from ML, which is not always the case. Guidance to assess this assumption is out of our scope. While the focus of *PerSpecML* are requirements engineers, the specialists who provide a clear understanding of what needs to be built, other stakeholders such as project leaders can preside the application of *PerSpecML*. In addition, we are aware that not every ML-enabled system needs to address all the concerns we proposed and not every ML-enabled system needs to implement

them to the same degree. Beyond qualities of ML components, of course, we also care about qualities of the system as a whole, including response time, safety, security, and usability. That is, traditional RE for the entire system and its non-ML components is just as important. Note that when considering the overall system, general quality characteristics of software products such as the ones mentioned in the ISO/IEC 25010 standard (ISO/IEC, 2011), should also be analyzed.

In terms of **expected benefits**, the main purpose of *PerSpecML* is to support the specification of ML-enabled systems by analyzing the ML perspective-based diagram and filling out the ML specification template. Nevertheless, we believe *PerSpecML* may eventually be useful in various situations. First, to validate an already specified ML-enabled system. In this case, the concerns would be a reference since they came from diverse source of knowledge (literature review, practical experiences and an external industrial experience on building ML-enabled systems (HULTEN, 2019)). Second, *PerSpecML* may help design ML-enabled systems, since it includes (i) different components, including functional and non-functional properties, (ii) how they interact with each other, (iii) how they are deployed, and (iv) how they contribute with business requirements. Third, *PerSpecML* is applicable to the most common ML approaches from typical ML domains, such as classification or regression problems, to more complex domains, such as computer vision and natural language processing. In fact, in the validations we conducted, we used different type of ML domains.

5.9

Concluding Remarks

In this chapter we presented *PerSpecML*, a perspective-based approach for specifying ML-enabled systems, designed to identify which attributes, including ML and non-ML, are important to contribute to the overall system's quality. The approach empowers requirements engineers to analyze, with the support of business owners, domain experts, designers, software and ML engineers, and data scientists, 60 concerns related to 28 tasks that practitioners typically face in ML projects, grouping them into five perspectives: system objectives, user experience, infrastructure, model, and data.

We introduced two main artifacts of *PerSpecML*: (i) the Perspective-based ML Tasks and Concern Diagram that provides a holistic view of ML-enabled systems, and (ii) its corresponding specification template that provides a standardized format for documenting and organizing the applicable concerns. Together, these artifacts serve to guide practitioners in collaboratively

and comprehensively designing ML-enabled systems, enhancing their clarity, exploring trade-offs between conflicting requirements, uncovering hidden or overlooked requirements, and improving decision-making.

The creation of *PerSpecML* involved a series of validations conducted in diverse contexts, encompassing both academic and real-world scenarios as suggested in (GORSCHEK et al., 2006) for scaling proposals up to practice. The evaluation process began with a validation in academia, where students from two courses of SE for data science participated in specifying an ML-enabled system for a toy problem. This initial validation mainly showcased the promise of the approach and its potential for improvement in terms of ease of use. The static validation in an industry setting involved discussions with practitioners of a R&D initiative, analyzing specifications retroactively for two ready-made ML projects. This validation highlighted *PerSpecML*'s role as a road for identifying key components that could be missed without using the approach, but also identified opportunities for improvements related to usability. Finally, the dynamic validation engaged both novice and experienced practitioners of a Brazilian large e-commerce company, who specified two real ML-enabled systems from scratch using *PerSpecML*. The feedback from previous validations allowed the practitioners to focus on improvements related to the completeness of the concerns and how to use the approach. As a result of the diverse evaluations and continuous improvements, *PerSpecML* stands as a promising approach, poised to positively impact the specification of ML-enabled systems.

6

Contributions, Limitations, and Future Work

6.1

Contributions

The design and development of ML-enabled systems has proven to be complicated and challenging. Despite remarkable contributions in the field, many organizations continue to struggle with specifying such systems. This thesis developed and evaluated *PerSpecML*, a perspective-based approach for support the specification of ML-enabled systems. *PerSpecML* includes a catalog of concerns and tasks, a conceptual model, a specification template, and an available package that summarizes all the mentioned elements allowing our solution to be applied. This section discusses the contributions of this thesis.

Catalog of concerns and tasks. Creating and revising requirements for ML-enabled systems considering the five perspectives of *PerSpecML* can be a tedious activity. The catalogues aim to speed up the identification and specification process and reduce the required competency and technical experience for working with such systems. This element serves as a comprehensive repository that identifies and defines the key components and issues relevant to the development of ML-enabled systems. It outlines the various aspects, considerations, and challenges that need to be analyzed during the specification process. This catalog provides a structured and organized reference for practitioners, helping them understand the important dimensions of ML system development.

Conceptual model. We called this element as the Perspective-Based ML Task and Concern Diagram, which is a visual representation of the concerns and tasks outlined in the catalog. This diagram captures the relationships, dependencies, and interactions between different elements within the ML-enabled system. It offers a high-level view that aids in understanding the holistic structure of the larger system. This visual model simplifies complex ideas and helps stakeholders grasp the big picture. By using this element of *PerSpecML*, practitioners can identify trade-offs between conflicting objectives and requirements for ML projects.

Specification template. We called this element as the Perspective-Based ML Specification Template, which is a standardized document that outlines how to document the specific requirements, constraints, and design

considerations associated with an ML-enabled system. It offers a consistent format for describing individual components, ensuring that all necessary information is captured. By using this element of *PerSpecML*, practitioners can streamline the documentation process, making it more organized and accessible for stakeholders involved in the development and maintenance of the larger system.

Available package. This element is a set of resources that allows practitioners to effectively apply *PerSpecML*. This package includes the explanation and artifacts of the three elements mentioned above (catalog, conceptual model, template) as well as use cases and a usage guide, that facilitate the practical execution of *PerSpecML*. It helps bridge the gap between theory and practice, enabling teams to put the approach into action. We make these resources available in Miroverso¹.

These four elements collectively form our approach for supporting the identification of requirements for specifying ML-enabled systems, offering a structured and practical way to address the complexities and challenges associated with ML projects.

6.2

Limitations

As with any research, the approach proposed in this thesis has its limitations. Some of them concern threats to the validity of the conducted empirical studies, which we in Section 5.7. In the rest of this section, further limitations are discussed.

PerSpecML was developed with a focus on supervised learning, where the dataset includes labeled instances with clear distinctions between dependent and independent variables. While this perspective aligns well with scenarios where an ML model is trained using labeled data, it may not be directly applicable to other ML paradigms, such as unsupervised learning or reinforcement learning, where the labeling of data instances can be less straightforward or even absent. In these cases, additional perspectives, tasks, or concerns may be needed to cover the unique characteristics and requirements of the specific ML task at hand.

The main artifact of *PerSpecML*, the perspective-based ML task and concern diagram that summarizes the main elements of our approach, is based on a conceptual model. This artifact prioritizes simplicity and ease of use, making it accessible to a wide range of non-technical or non-RE expert stakeholders (*e.g.*, business owners, data scientists, domain experts).

¹<https://miro.com/miroverse/perspecml-machine-learning/>

However, this emphasis on simplicity may result in a limitation of addressing more complex requirements scenarios that could be adequately covered by other well-established RE modeling techniques, such as \hat{t}^* and GORE, which have been reported to be more challenging to learn among non-software engineers (NEACE; RONCACE; FOMIN, 2018; DIMITRAKOPOULOS et al., 2019).

Another limitation of this thesis lies in the scope of the catalog of concerns used for conceiving *PerSpecML*. The content of the catalog of concerns, presented in Chapter 4, was derived from a literature review, an iterative evaluation with practitioners, and our own knowledge and experience in the field. While this approach has allowed us to identify and include a wide range of concerns, it is important to acknowledge that the dynamic field of ML continues to introduce new concerns. Therefore, our catalog may not capture all the concerns within the ML domain. As the landscape of ML evolves, some novel concerns may not yet be included in the catalog. This limitation emphasizes the need for ongoing updates and revisions to ensure the catalog remains comprehensive and aligned with the current state of ML practice.

6.3

Future Work

While the validations of *PerSpecML* have yielded positive results and provided valuable insights, there remain several avenues for future work and enhancements to further enrich the approach and its applications in the field. For instance:

- Investigating ways to automatically generate detailed documentation from the specifications provided in *PerSpecML* artifacts could significantly streamline project management and maintainability. This would further bridge the gap between specification and implementation phases.
- Conducting other studies and soliciting continuous feedback from practitioners who actively use *PerSpecML* in real projects would offer valuable insights into its long-term benefits.
- Given the potentially conflicting nature of the concerns highlighted in *PerSpecML*, studying trade-offs in this context becomes even more promising, as it offers a pathway to address the complex particularities of ML-enabled systems.
- The exploration of *PerSpecML*'s educational potential for novice practitioners entering the field of ML is also promising. By exploring the

educational potential of *PerSpecML*, we can contribute to the development of a new generation of ML practitioners who are well-equipped to navigate the complexities of ML projects, ultimately leading to improved software quality.

- Addressing the limitations exposed in Section 6.2 and expanding the approach’s capabilities to accommodate a broader range of ML paradigms and complex requirements.

AHMAD, K. et al. Requirements engineering for artificial intelligence systems: A systematic mapping study. **Information and Software Technology**, Elsevier, p. 107176, 2023. Cited 5 times in pages 17, 18, 46, 49, and 60.

AHMAD, K. et al. Requirements engineering framework for human-centered artificial intelligence software systems. **Applied Soft Computing**, Elsevier, v. 143, p. 110455, 2023. Cited 2 times in pages 32 and 49.

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. Cited 4 times in pages 29, 46, 58, and 60.

AHO, T. et al. Demystifying data science projects: A look on the people and process of data science today. In: SPRINGER. **Product-Focused Software Process Improvement: 21st International Conference, PROFES 2020, Turin, Italy, November 25–27, 2020, Proceedings 21**. [S.l.], 2020. p. 153–167. Cited 4 times in pages 11, 26, 27, and 106.

ALVES, A. P. S. et al. Status quo and problems of requirements engineering for machine learning: Results from an international survey. In: SPRINGER. **Product-Focused Software Process Improvement: 24th International Conference, PROFES 2023, Dornbirn, Austria, December 10–13**. [S.l.], 2023. p. 153–167. Cited 4 times in pages 16, 17, 19, and 29.

APPLE. **Human-interface-guidelines for ML solutions**. 2020. Disponível em: <https://developer.apple.com/design/human-interface-guidelines/machine-learning>. Cited in page 33.

ARPTEG, A. et al. Software engineering challenges of deep learning. In: IEEE. **2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)**. [S.l.], 2018. p. 50–59. Cited in page 29.

BARASH, G. et al. Bridging the gap between ml solutions and their business requirements using feature interactions. In: **Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering**. [S.l.: s.n.], 2019. p. 1048–1058. Cited 2 times in pages 39 and 68.

BASILI, V. R.; ROMBACH, H. D. The tame project: Towards improvement-oriented software environments. **IEEE Transactions on software engineering**, IEEE, v. 14, n. 6, p. 758–773, 1988. Cited 2 times in pages 50 and 84.

BEEDE, E. et al. A human-centered evaluation of a deep learning system deployed in clinics for the detection of diabetic retinopathy. In: **Proceedings of the 2020 CHI conference on human factors in computing systems**. [S.l.: s.n.], 2020. p. 1–12. Cited in page 16.

BELANI, H.; VUKOVIC, M.; CAR, Ž. Requirements engineering challenges in building ai-based complex systems. In: **International Requirements Engineering Conference Workshops (REW)**. [S.l.: s.n.], 2019. p. 252–255. Cited in page 28.

BERRY, D. M. Requirements engineering for artificial intelligence: What is a requirements specification for an artificial intelligence? In: **International Working Conference on RE: Foundation for Software Quality**. [S.l.: s.n.], 2022. p. 19–25. Cited 2 times in pages 18 and 29.

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>>. Cited in page 35.

CARLETON, A. D. et al. **Architecting the Future of Software Engineering: A National Agenda for Software Engineering Research and Development**. [S.l.], 2021. Cited 2 times in pages 17 and 18.

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. Cited in page 17.

CHUNG, L. et al. **Non-functional requirements in software engineering**. [S.l.]: Springer Science & Business Media, 2012. v. 5. Cited in page 24.

CHUPRINA, T.; MENDEZ, D.; WNUK, K. Towards artefact-based requirements engineering for data-centric systems. In: CEUR-WS. **Joint Workshops of the 27th International Conference on Requirements Engineering, REFSQ 2021-OpenRE, Posters and Tools Track, and Doctoral Symposium, Essen, Germany, 12 April 2021**. [S.l.], 2021. v. 2857. Cited in page 31.

CYSNEIROS, L. M.; LEITE, J. C. S. do P. Non-functional requirements orienting the development of socially responsible software. In: **Enterprise, Business-Process and Information Systems Modeling**. [S.l.]: Springer, 2020. p. 335–342. Cited in page 49.

CYSNEIROS, L. M.; RAFFI, M.; LEITE, J. C. S. do P. Software transparency as a key requirement for self-driving cars. In: IEEE. **2018 IEEE 26th international requirements engineering conference (RE)**. [S.l.], 2018. p. 382–387. Cited in page 17.

DALPIAZ, F.; NIU, N. Requirements engineering in the days of artificial intelligence. **IEEE Software**, IEEE, v. 37, n. 4, p. 7–10, 2020. Cited 2 times in pages 17 and 34.

DAMIAN, D. Stakeholders in global requirements engineering: Lessons learned from practice. **IEEE software**, IEEE, v. 24, n. 2, p. 21–27, 2007. Cited in page 24.

DAVIS, F. D. Perceived usefulness, ease of use, and user acceptance of information technology. **MIS quarterly**, JSTOR, p. 319–340, 1989. Cited in page 85.

DIMITRAKOPOULOS, G. et al. A capability-oriented modelling and simulation approach for autonomous vehicle management. **Simulation Modelling Practice and Theory**, Elsevier, v. 91, p. 28–47, 2019. Cited in page 120.

DORARD, L. **Machine Learning Canvas**. 2015. Disponível em: <<https://www.machinelearningcanvas.com/>>. Cited in page 30.

FALESSI, D. et al. Empirical software engineering experts on the use of students and professionals in experiments. **Empirical Software Engineering**, Springer, v. 23, p. 452–489, 2018. Cited 2 times in pages 86 and 114.

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. Cited 2 times in pages 25 and 49.

FOSNOT, C. T. **Constructivism: Theory, perspectives, and practice**. [S.l.]: Teachers College Press, 2013. Cited in page 50.

FRANCH, X.; JEDLITSCHKA, A.; MARTÍNEZ-FERNÁNDEZ, S. A requirements engineering perspective to ai-based systems development: A vision paper. In: SPRINGER. **International Working Conference on Requirements Engineering: Foundation for Software Quality**. [S.l.], 2023. p. 223–232. Cited in page 34.

FRY, H. **Hello World: How to be Human in the Age of the Machine**. [S.l.]: Random House, 2018. Cited in page 16.

GARTNER. **Gartner Identifies the Top Strategic Technology Trends for 2021**. 2020. Disponível em: <<https://www.gartner.com/smarterwithgartner/gartner-top-strategic-technology-trends-for-2021>>. Cited in page 16.

GIRAY, G. A software engineering perspective on engineering machine learning systems: State of the art and challenges. **Journal of Systems and Software**, Elsevier, v. 180, p. 111031, 2021. Cited 2 times in pages 29 and 46.

GOOGLE. **People + AI Guidebook**. 2021. Disponível em: <<https://pair.withgoogle.com/guidebook>>. Cited in page 33.

GORSCHKE, T. et al. A model for technology transfer in practice. **IEEE**, IEEE, v. 23, n. 6, p. 88–95, 2006. Cited 6 times in pages 11, 20, 58, 59, 61, and 117.

HABIBULLAH, K. M.; GAY, G.; HORKOFF, J. Non-functional requirements for machine learning: Understanding current use and challenges among practitioners. **Requirements Engineering**, Springer, p. 1–34, 2023. Cited 3 times in pages 17, 29, and 49.

HERRMANN, A. Requirements engineering in practice: There is no requirements engineer position. In: SPRINGER. **Requirements Engineering: Foundation for Software Quality: 19th International Working Conference, REFSQ 2013, Essen, Germany, April 8-11, 2013. Proceedings 19**. [S.l.], 2013. p. 347–361. Cited 2 times in pages 16 and 25.

HEYN, H.-M. et al. Requirement engineering challenges for ai-intense systems development. In: **1st Workshop on AI Engineering – Software Engineering for AI (WAIN2021)**. [S.l.: s.n.], 2021. Cited 2 times in pages 28 and 49.

HORKOFF, J. Non-functional requirements for machine learning: Challenges and new directions. In: **International Requirements Engineering Conference (RE)**. [S.l.: s.n.], 2019. p. 386–391. Cited in page 49.

HULTEN, G. **Building Intelligent Systems**. [S.l.]: Springer, 2019. Cited 6 times in pages 15, 49, 51, 60, 63, and 116.

ISHIKAWA, F.; YOSHIOKA, N. How do engineers perceive difficulties in engineering of machine-learning systems? In: **International Workshop on Conducting Empirical Studies in Industry (CESI)**. [S.l.: s.n.], 2019. p. 2–9. Cited 3 times in pages 17, 28, and 29.

ISO/IEC. **ISO/IEC 25010:Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models**. 2011. Disponível em: <<https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>>. Cited 2 times in pages 56 and 116.

ISO/IEC. **ISO/IEC 25012: Software engineering – software product quality requirements and evaluation (SQuaRE) – data quality model**. 2012. Disponível em: <<https://www.iso.org/standard/35736.html>>. Cited in page 70.

JORDAN, M. I.; MITCHELL, T. M. Machine learning: Trends, perspectives, and prospects. **Science**, American Association for the Advancement of Science, v. 349, n. 6245, p. 255–260, 2015. Cited 2 times in pages 25 and 26.

KALINOWSKI, M. et al. Lean r&d: An agile research and development approach for digital transformation. In: SPRINGER. **International Conference on Product-Focused Software Process Improvement**. [S.l.], 2020. p. 106–124. Cited 3 times in pages 20, 49, and 92.

KÄSTNER, C. **Machine Learning in Production: From Models to Products**. [S.l.: s.n.], 2022. Cited 4 times in pages 11, 16, 18, and 28.

KÄSTNER, C.; KANG, E. Teaching software engineering for ai-enabled systems. In: **Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering Education and Training**. [S.l.: s.n.], 2020. p. 45–48. Cited 2 times in pages 27 and 36.

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. Cited 3 times in pages 63, 86, and 106.

KITCHENHAM, B.; CHARTERS, S. Guidelines for performing systematic literature reviews in software engineering. **Keele University and Durham University Joint Report, Technical Report EBSE 2007-001**, 2007. Cited 2 times in pages 36 and 37.

KONTIO, J.; LEHTOLA, L.; BRAGGE, J. Using the focus group method in software engineering: obtaining practitioner and user experiences. In: IEEE. **Proceedings. 2004 International Symposium on Empirical Software Engineering, 2004. ISESE'04.** [S.l.], 2004. p. 271–280. Cited 3 times in pages 53, 95, and 115.

KUMENO, F. Software engineering challenges for machine learning applications: A literature review. **Intelligent Decision Technologies**, IOS Press, v. 13, n. 4, p. 463–476, 2019. Cited in page 35.

KUWAJIMA, H.; YASUOKA, H.; NAKAE, T. Engineering problems in ml systems. **Machine Learning**, Springer, v. 109, n. 5, p. 1103–1126, 2020. Cited 2 times in pages 17 and 28.

LAMSWEERDE, A. v. **Requirements engineering: from system goals to UML models to software specifications.** [S.l.]: John Wiley & Sons, Ltd, 2009. Cited in page 18.

LEONARDO, R. Pico: Model for clinical questions. **Evidence Based Medicine and Practice**, v. 3, n. 115, p. 2, 2018. Cited in page 37.

LEWIS, G. A.; BELLOMO, S.; OZKAYA, I. Characterizing and detecting mismatch in machine-learning-enabled systems. In: IEEE. **2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN).** [S.l.], 2021. p. 133–140. Cited 3 times in pages 15, 29, and 55.

LORENZONI, G. et al. Machine learning model development from a software engineering perspective: A systematic literature review. **arXiv preprint arXiv:2102.07574**, 2021. Cited in page 35.

LOUCOPOULOS, P.; KARAKOSTAS, V. **System requirements engineering.** [S.l.]: McGraw-Hill, Inc., 1995. Cited in page 24.

LWAKATARE, L. E. et al. A taxonomy of software engineering challenges for machine learning systems: An empirical investigation. In: SPRINGER, CHAM. **International Conference on Agile Software Development.** [S.l.], 2019. p. 227–243. Cited 2 times in pages 17 and 29.

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**, Elsevier, v. 127, p. 106368, 2020. Cited in page 34.

MAALEJ, W.; PHAM, Y. D.; CHAZETTE, L. Tailoring requirements engineering for responsible ai. **Computer**, IEEE, v. 56, n. 4, p. 18–27, 2023. Cited in page 30.

MAFFEY, K. R. et al. Mlteing models: Negotiating, evaluating, and documenting model and system qualities. **arXiv preprint arXiv:2303.01998**, 2023. Cited in page 32.

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. Cited 3 times in pages 17, 46, and 58.

MENDES, E. et al. When to update systematic literature reviews in software engineering. **Journal of Systems and Software**, Elsevier, v. 167, p. 110607, 2020. Cited in page 38.

MICROSOFT. **Microsoft's framework for building AI systems responsibly**. 2022. Disponível em: <<https://blogs.microsoft.com/wp-content/uploads/prod/sites/5/2022/06/Microsoft-Responsible-AI-Standard-v2-General-Requirements-3.pdf>>. Cited in page 33.

MOURAO, E. et al. On the performance of hybrid search strategies for systematic literature reviews in software engineering. **Information and Software Technology**, v. 123, p. 106294:1–12, 2020. ISSN 0950-5849. Cited 2 times in pages 37 and 47.

NAHAR, N. et al. A meta-summary of challenges in building products with ml components – collecting experiences from 4758+ practitioners. In: **2023 IEEE/ACM 2nd International Conference on AI Engineering – Software Engineering for AI (CAIN)**. [S.l.: s.n.], 2023. p. 171–183. Cited 2 times in pages 17 and 46.

NAHAR, N. et al. Collaboration challenges in building ml-enabled systems: Communication, documentation, engineering, and process. In: **Proceedings of the 44th International Conference on Software Engineering**. [S.l.: s.n.], 2022. p. 413–425. Cited in page 15.

NAKAMICHI, K. et al. Requirements-driven method to determine quality characteristics and measurements for machine learning software and its evaluation. In: **IEEE. 2020 IEEE 28th International Requirements Engineering Conference (RE)**. [S.l.], 2020. p. 260–270. Cited in page 31.

NALCHIGAR, S.; YU, E.; KESHAVJEE, K. Modeling machine learning requirements from three perspectives: a case report from the healthcare domain. **Requirements Engineering**, Springer, v. 26, n. 2, p. 237–254, 2021. Cited 2 times in pages 31 and 39.

NASCIMENTO, E. et al. Software engineering for artificial intelligence and machine learning software: A systematic literature review. **arXiv preprint arXiv:2011.03751**, 2020. Cited in page 35.

NASCIMENTO, E. de S. et al. Understanding development process of ml systems: Challenges and solutions. In: **Empirical Software Engineering and Measurement (ESEM)**. [S.l.: s.n.], 2019. p. 1–6. Cited in page 17.

NEACE, K.; RONCACE, R.; FOMIN, P. Goal model analysis of autonomy requirements for unmanned aircraft systems. **Requirements Engineering**, Springer, v. 23, p. 509–555, 2018. Cited in page 120.

NUSEIBEH, B.; EASTERBROOK, S. Requirements engineering: a roadmap. In: **Proceedings of the Conference on the Future of Software Engineering**. [S.l.: s.n.], 2000. p. 35–46. Cited in page 24.

PEI, Z. et al. Requirements engineering for machine learning: A review and reflection. In: IEEE. **2022 IEEE 30th International Requirements Engineering Conference Workshops (REW)**. [S.l.], 2022. p. 166–175. Cited 3 times in pages 17, 46, and 60.

PETERSEN, K.; VAKKALANKA, S.; KUZNIARZ, L. Guidelines for conducting systematic mapping studies in software engineering: An update. **Information and Software Technology**, v. 64, p. 1–18, 2015. Cited in page 36.

RAHIMI, M. et al. Toward requirements specification for machine-learned components. In: IEEE. **2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)**. [S.l.], 2019. p. 241–244. Cited in page 30.

SALDAÑA, J. **The coding manual for qualitative researchers, 4th Edition**. [S.l.]: SAGE Publications Limited, 2021. Cited in page 41.

SCHUH, G. et al. Identifying and analyzing data model requirements and technology potentials of machine learning systems in the manufacturing industry of the future. In: **International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS)**. [S.l.: s.n.], 2020. p. 1–10. Cited in page 35.

SERVICE, R. W. 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. Cited in page 95.

SIEBERT, J. et al. Construction of a quality model for machine learning systems. **Software Quality Journal**, Springer, v. 30, n. 2, p. 307–335, 2022. Cited in page 31.

TURNER, M. et al. Does the technology acceptance model predict actual use? a systematic literature review. **Information and software technology**, Elsevier, v. 52, n. 5, p. 463–479, 2010. Cited in page 113.

VILLAMIZAR, H.; ESCOVEDO, T.; KALINOWSKI, M. Requirements engineering for machine learning: A systematic mapping study. In: IEEE. **2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)**. [S.l.], 2021. p. 29–36. Cited 6 times in pages 17, 20, 22, 29, 34, and 58.

VILLAMIZAR, H.; KALINOWSKI, M.; LOPES, H. A catalogue of concerns for specifying machine learning-enabled systems. In: **2022 25th Workshop on Requirements Engineering (WER)**. [S.l.: s.n.], 2022. Cited 4 times in pages 11, 20, 49, and 52.

VÍLLAMIZAR, H.; KALINOWSKI, M.; LOPES, H. Towards perspective-based specification of machine learning-enabled systems. In: IEEE. **2022 48th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)**. [S.l.], 2022. p. 112–115. Cited 4 times in pages 11, 20, 60, and 61.

VILLAMIZAR, H. et al. Identifying concerns when specifying machine learning-enabled systems: A perspective-based approach. **arXiv preprint arXiv:2309.07980**, 2023. Cited in page 21.

VILLAMIZAR, H. et al. A systematic mapping study on security in agile requirements engineering. In: **Euromicro conference on software engineering and advanced applications (SEAA)**. [S.l.: s.n.], 2018. p. 454–461. Cited in page 41.

VOGELSANG, A.; BORG, M. Requirements engineering for machine learning: Perspectives from data scientists. In: IEEE. **2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)**. [S.l.], 2019. p. 245–251. Cited 2 times in pages 17 and 29.

WAGNER, S. et al. Status quo in requirements engineering: A theory and a global family of surveys. **ACM Transactions on Software Engineering and Methodology (TOSEM)**, ACM New York, NY, USA, v. 28, n. 2, p. 1–48, 2019. Cited in page 24.

WAN, Z. et al. How does machine learning change software development practices? **IEEE Transactions on Software Engineering**, IEEE, 2019. Cited 2 times in pages 28 and 30.

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**. [S.l.: s.n.], 2018. p. 1–10. Cited 2 times in pages 16 and 25.

WIERINGA, R. et al. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. **Requirements Engineering**, v. 11, n. 1, p. 102–107, 2006. Cited 2 times in pages 36 and 40.

WOHLIN, C. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: **International Conference on Evaluation and Assessment in Software Engineering (EASE)**. [S.l.: s.n.], 2014. p. 38. Cited 2 times in pages 37 and 39.

WOHLIN, C. et al. Guidelines for the search strategy to update systematic literature reviews in software engineering. **Information and Software Technology**, Elsevier, v. 127, p. 106366, 2020. Cited in page 37.

WOHLIN, C. et al. **Experimentation in software engineering**. [S.l.]: Springer Science & Business Media, 2012. Cited 6 times in pages 20, 40, 45, 58, 59, and 113.

A

RE for ML Papers Identified in the Literature Review

This appendix outlines the papers identified in the literature review we conducted in 2021 (see Chapter 3). In total, we identified 35 papers, some of them published in premier SE conferences and journals.

[P1] K. Nakamichi, K. Ohashi, I. Namba, R. Yamamoto, M. Aoyama, L. Joeckel, J. Siebert, and J. Heidrich, **“Requirements-driven method to determine quality characteristics and measurements for machine learning software and its evaluation,”** in International Requirements Engineering Conference (RE), 2020, pp. 260–270.

[P2] F. Ishikawa and Y. Matsuno, **“Evidence-driven requirements engineering for uncertainty of machine learning-based systems,”** in International Requirements Engineering Conference (RE), 2020, pp. 346–351.

[P3] C. Kästner and E. Kang, **“Teaching software engineering for ai-enabled systems,”** in International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET), 2020, pp. 45–48.

[P4] H. Liu, S. Eksmo, J. Risberg, and R. Hebig, **“Emerging and changing tasks in the development process for machine learning systems,”** in Proceedings of the International Conference on Software and System Processes (ICSSP), 2020, pp. 125–134.

[P5] H. Kuwajima, H. Yasuoka, and T. Nakae, **“Engineering problems in machine learning systems,”** Machine Learning, vol. 109, no. 5, pp. 1103–1126, 2020.

[P6] R. Akkiraju, V. Sinha, A. Xu, J. Mahmud, P. Gundecha, Z. Liu, X. Liu, and J. Schumacher, **“Characterizing machine learning processes: A maturity framework,”** in International Conference on Business Process Management (BPM), 2020, pp. 17–31.

[P7] K. Hamada, F. Ishikawa, S. Masuda, M. Matsuya, and Y. Ujita, **“Guidelines for quality assurance of machine learning-based artificial intelligence,”** in International Conference on Software Engineering & Knowledge Engineering

(SEKE), 2020, pp. 335–341.

[P8] N. Caporusso, T. Helms, and P. Zhang, **“A meta-language approach for machine learning,”** in International Conference on Applied Human Factors and Ergonomics, 2019, pp. 192–201.

[P9] D. Cirqueira, D. Nedbal, M. Helfert, and M. Bezbradica, **“Scenario- based requirements elicitation for user-centric explainable ai,”** in International Cross-Domain Conference for Machine Learning and Knowledge Extraction, 2020, pp. 321–341.

[P10] H. Belani, M. Vukovic, and Z^v. Car, **“Requirements engineering challenges in building ai-based complex systems,”** in International Requirements Engineering Conference Workshops (REW), 2019, pp. 252–255.

[P11] A. Vogelsang and M. Borg, **“Requirements engineering for machine learning: Perspectives from data scientists,”** in International Requirements Engineering Conference Workshops (REW), 2019, pp. 245–251.

[P12] M. Rahimi, J. L. Guo, S. Kokaly, and M. Chechik, **“Toward requirements specification for machine-learned components,”** in International Requirements Engineering Conference Workshops (REW), 2019, pp. 241–244.

[P13] E. de Souza Nascimento, I. Ahmed, E. Oliveira, M. P. Palheta, I. Steinmacher, and T. Conte, **“Understanding development process of machine learning systems: Challenges and solutions,”** in International Symposium on Empirical Software Engineering and Measurement (ESEM), 2019, pp. 1–6.

[P14] J. Horkoff, **“Non-functional requirements for machine learning: Challenges and new directions,”** in International Requirements Engineering Conference (RE), 2019, pp. 386–391.

[P15] B. Ahmed, T. Dannhauser, and N. Philip, **“A lean design thinking methodology (ldtm) for machine learning and modern data projects,”** in 2018 10th Computer Science and Electronic Engineering (CEECE), 2018, pp. 11–14.

[P16] H. Kuwajima and F. Ishikawa, **“Adapting square for quality assessment of artificial intelligence systems,”** in International Symposium on Software Reliability Engineering Workshops (ISSREW), 2019, pp. 13–18.

- [P17] F. Ishikawa and N. Yoshioka, **“How do engineers perceive difficulties in engineering of machine-learning systems?-questionnaire survey,”** in International Workshop on Conducting Empirical Studies in Industry (CESI) and 6th International Workshop on Software Engineering Research and Industrial Practice (SER&IP), 2019, pp. 2–9.
- [P18] B. C. Hu, R. Salay, K. Czarnecki, M. Rahimi, G. Selim, and M. Chechik, **“Towards requirements specification for machine-learned perception based on human performance,”** in 7th International Workshop on Artificial Intelligence for Requirements Engineering (AIRE), 2020, pp. 48–51.
- [P19] S. Amershi, A. Begel, C. Bird, R. DeLine, H. Gall, E. Kamar, N. Nagappan, B. Nushi, and T. Zimmermann, **“Software engineering for machine learning: A case study,”** in International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), 2019, pp. 291–300.
- [P20] Z. Wan, X. Xia, D. Lo, and G. C. Murphy, **“How does machine learning change software development practices?”** IEEE Transactions on Software Engineering, 2019.
- [P21] H. Washizaki, H. Uchida, F. Khomh, and Y.-G. Gueheneuc, **“Studying software engineering patterns for designing machine learning systems,”** in 2019 10th International Workshop on Empirical Software Engineering in Practice (IWSEEP), 2019, pp. 49–495.
- [P22] M. Hall, D. Harborne, R. Tomsett, V. Galetic, S. Quintana-Amate, A. Nottle, and A. Preece, **“A systematic method to understand requirements for explainable ai (xai) systems,”** in Proceedings of the IJCAI Workshop on eXplainable Artificial Intelligence (XAI 2019), Macau, China, 2019.
- [P23] Q. V. Liao, D. Gruen, and S. Miller, **“Questioning the ai: informing design practices for explainable ai user experiences,”** in Proceedings of the CHI Conference on Human Factors in Computing Systems, 2020, pp. 1–15.
- [P24] A. Banks and R. Ashmore, **“Requirements assurance in machine learning,”** in The AAAI’s Workshop on Artificial Intelligence Safety (SafeAI), 2019.
- [P25] H. Challa, N. Niu, and R. Johnson, **“Faulty requirements made valuable: On the role of data quality in deep learning,”** in 7th International Workshop

on Artificial Intelligence for Requirements Engineering (AIRE), 2020, pp. 61–69.

[P26] B. Jahic, N. Guelfi, and B. Ries, **“Specifying key-properties to improve the recognition skills of neural networks,”** 2020.

[P27] S. Nalchigar, E. Yu, and K. Keshavjee, **“Modeling machine learning requirements from three perspectives: a case report from the healthcare domain,”** Requirements Engineering, pp. 1–18, 2021.

[P28] A. Pereira and C. Thomas, **“Challenges of machine learning applied to safety-critical cyber-physical systems,”** Machine Learning and Knowledge Extraction, vol. 2, no. 4, pp. 579–602, 2020.

[P29] C. Wilhjelmsen and A. A. Younis, **“A threat analysis methodology for security requirements elicitation in machine learning based systems,”** in International Conference on Software Quality, Reliability and Security Companion (QRS-C), 2020, pp. 426–433.

[P30] N. Balasubramaniam, M. Kauppinen, S. Kujala, and K. Hiekkanen, **“Ethical guidelines for solving ethical issues and developing ai systems,”** in International Conference on Product-Focused Software Process Improvement (PROFES), 2020, pp. 331–346.

[P31] M. Anisetti, C.A. Ardagna, E. Damiani, and P.G. Panero, **“A methodology for non-functional property evaluation of machine learning models,”** in Proceedings of the 12th International Conference on Management of Digital EcoSystems (MEDES), 2020, pp. 38–45.

P[32] M. Hesenius, N. Schwenzfeier, O. Meyer, W. Koop, and V. Gruhn, **“Towards a software engineering process for developing data-driven applications,”** in International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE), 2019, pp. 35–41.

P[33] R. Lukyanenko, A. Castellanos, J. Parsons, M. C. Tremblay, and V. C. Storey, **“Using conceptual modeling to support machine learning,”** in International Conference on Advanced Information Systems Engineering (CAISE), 2019, pp. 170–181.

P[34] L. M. Cysneiros and J. C. S. do Prado Leite, **“Non-functional require-**

ments orienting the development of socially responsible software,” in Enterprise, Business-Process and Information Systems Modeling. Springer, 2020, pp. 335–342.

P[35] G. Barash, E. Farchi, I. Jayaraman, O. Raz, R. Tzoref-Brill, and M. Zalmancovich, **“Bridging the gap between ml solutions and their business requirements using feature interactions,”** in Proceedings of the Joint Meeting of the European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE), 2019, pp. 1048–1058.

B

Details of the Relationship of Concerns by Perspective

This appendix details the relationships of each concern considered by *PerSpecML*. These relationships are grouped by perspective but some of them can be related to concerns of other perspectives.

Table B.1: Relationships between system objectives perspective concerns.

Relationship		Description
Context	Bias & Fairness	The context of which the ML-enabled system operates can have ethical implications, especially in sensitive domains. <i>E.g.</i> , ML models for making critical decisions need to be carefully designed to avoid biases and unfair outcomes
	Security Privacy	The context of which the ML-enabled system operates can have also security & privacy implications. ML models can be vulnerable to various types of attacks. It's crucial for stakeholders to be aware of these threats and implement strategies to secure their models, especially in applications where security and trust are paramount
ML funct	Algorithm and model selection	The ML functionality (ML funct) guides the selection of appropriate ML algorithms. Different tasks (<i>e.g.</i> , classification, regression) require specific algorithms that are suitable for the task at hand
	Performance metrics	The ML functionality affects how the model's performance is evaluated and measured. The choice of performance metrics depends on the specific task and the priorities of the application. <i>E.g.</i> , in a medical diagnosis task, recall might be more critical to minimize false negatives
Hypoth	Organizational goals and leading indicators	The profit hypothesis directly aligns the ML system's goals with the broader business objectives. It helps establish specific, measurable targets for generating value, whether it's increased revenue, cost savings, customer retention, or other key performance indicators
ML trade-offs	Performance metrics	The choice of trade-offs can affect the performance of the ML model. <i>E.g.</i> , prioritizing speed over accuracy might lead to the selection of simpler models that can make predictions more quickly but with potentially lower accuracy
	Explainability Interpretability	Trade-offs can affect the interpretability of the ML model. More complex models might offer higher accuracy but be harder to interpret
	Scalability	The trade-offs chosen impact the computational resources. More complex models may require more memory and processing power, affecting the scalability and efficiency of the solution
	Data quantity and quality	The trade-offs can impact the volume and quality of data required for training. Some ML models might perform well with smaller datasets, while others might need larger datasets to generalize effectively
	Inference time and model serving	The trade-offs can impact the response time of the system. Choosing trade-offs that prioritize low latency can influence the choice of model architecture and deployment strategy
	Cost	The trade-offs can impact the costs associated with the ML project. More complex models might lead to higher infrastructure costs and operational expenses

Table B.2: Relationships between user experience perspective concerns.

	Relationship	Description
Acceptance	Performance metrics	The choice of performance metrics allows to ensure that the ML model is being evaluated in a way that is meaningful to the users
	Explainability Interpretability	By emphasizing explainability and interpretability, you cater to stakeholders' desire to comprehend how the model arrives at its decisions, increasing their acceptance and willingness to adopt the model's predictions
	Bias & Fairness	Models that are biased or unfair can have negative impacts on user experience. Defining bias and fairness helps create positive and inclusive user experiences
E&T	ML trade-off	Education and training (E&T). By understanding these trade-offs, users and stakeholders can make informed decisions throughout the ML project lifecycle. This awareness helps in achieving a balance that aligns with project goals, resource constraints, and ethical considerations

Table B.3: Relationships between infrastructure perspective concerns.

	Relationship	Description
Data streaming	Inference time	Data streaming minimizes latency by processing and acting on data in near real time, crucial for time-sensitive applications
	Data operations	data streams may require on-the-fly pre-processing and feature extraction, demanding efficient techniques
	Scalability	Data streaming often involves handling high data volumes. Designing ML systems to handle data streams efficiently requires scalability
	Incremental learning	With data streaming, models can be updated and retrained in real time as new data becomes available, ensuring models remain relevant and accurate
MS	Cost	Different model serving (MS) architectures have different cost implications. Choosing the right architecture can help optimize the infrastructure costs
IL	Acceptance	Incremental learning (IL) may require human oversight to ensure that updates align with performance metrics, business goals and fairness considerations
	High quality data	IL relies on the assumption that new data is of good quality and representative of the problem domain
Storage	Inference time	The choice of storage solutions affects the speed at which data can be retrieved and processed, impacting the overall system's latency and throughput
	Scalability	Scalability is influenced by storage solutions that can accommodate growing data volumes without compromising performance
	Data streaming	Data streaming projects require storage solutions that can handle continuous and high-velocity data influx while maintaining low latency
	Reproducibility	Intermediate storage of cleaned and pre-processed data ensures reproducibility and reduces the need for repeated processing
	Cost	Storage costs can be a significant part of ML project budgets
Monitorability	Performance metrics	Monitorability enables to track the performance metrics of ML models. This helps ensure that ML models are providing accurate predictions over time
	Security Privacy	Keep an eye on any unusual or unauthorized activities related to the ML system's data and models
	High quality data	Tracking the quality of incoming data and identifying potential issues or anomalies is a good practice since poor data might affect the model's performance
Telemetry	Frequency Forcefulness	Capture telemetry data related to user interactions, preferences, and behaviors. This can provide insights into how users are engaging with the ML system
	Learning feedback	Telemetry facilitates a feedback loop by capturing user interactions and responses to model predictions. This feedback can be used to improve model accuracy and relevance

Table B.4: Relationships between model perspective concerns.

	Relationship	Description
Alg. selection	Hyper-parameter tuning	Different algorithms have different parameters that can be set to improve the performance
	Baseline model	When experimenting with multiple ML models comparing against the baseline helps to choose ML models that outperform the initial
	Model size & complexity	More complex ML algorithms might provide higher accuracy but could be harder to interpret, while simpler algorithms are often more interpretable
	Data quantity	Some ML algorithms require larger datasets to perform well, while others can work effectively with smaller amounts of data
Input-Output	Exp & Int	It's easier to interpret models when inputs and outputs have clear meanings
	Data operations	The inputs guide the selection and engineering of relevant features
	Data streaming	The inputs and outputs must be known at hand to set this service
	Model serving	The inputs determine how it will provide data to the model during deployment. The outputs determine how predictions are presented to users
	Incremental learning	The outputs generated by the model can serve as feedback to improve the model's performance
	Hybrid decision Intelligence	The outputs generated by the model can be combined with heuristics to create a more comprehensive solution
HT	Learning time	Longer learning times can provide the opportunity to perform more extensive hyperparameter tuning (HT), leading to better-tailored models
LT	Incremental learning	the learning time (LT) can influence the speed at which ML models iterate and deploy updated versions
	Cost	Longer learning times might require more computational resources, potentially leading to higher costs
PM	Hyper-parameter tuning	Performance metrics (PM) guide model optimization. Adjusting parameters to optimize a specific metric may be necessary
Exp & Int	Bias & Fairness	An emphasis on explainability enables the identification of bias sources within the model
	Maintainability	When a model's outputs are explainable and interpretable, it becomes easier to identify and rectify errors
	Accountability	Defining explainability and interpretability cultivates trust in the model's predictions by making its decision-making process transparent
Scalability	Data quantity & Streaming	Scalability considerations influence how well the model can handle increasing volumes of data, and ensure that the ML model remains effective
	Cost	Scalability involves efficient use of computational resources like memory, processing power, and storage. Properly defined scalability minimizes wastage and ensures optimal resource utilization
	Model serving	Scalability considerations impact how quickly the model can process incoming data and provide predictions within acceptable timeframes
	Learning time	Scalability affects the time it takes to train an ML model
B & F	High quality data	Defining bias and fairness (B&F) guides the data collection process to ensure that representative and unbiased data is used for training the model
	Data operations	B&F considerations influence which features are selected and how they are used to avoid introducing biases in the ML model
S & P	Data selection	Security and privacy (S & P) considerations influence the collection of data, ensuring that sensitive information is not inadvertently included
	Anonymization	Sensitive data may need to be anonymized to protect individual identities while still maintaining the utility of the data
	Model serving	S & P considerations influence how models are deployed and updated, ensuring, <i>e.g.</i> , that endpoints are properly secured against attacks
	Storage	Decisions regarding how and where data is stored are guided by security and privacy concerns to prevent unauthorized access and potential breaches
Inference time	Scalability	Efficient inference times enable the model to handle a larger volume of requests concurrently, ensuring scalability for high user demand
	Data streaming	In scenarios where ML models are integrated into data processing pipelines, fast inference times ensure smooth and timely data flow
	Model serving	For certain deployments, <i>e.g.</i> , on edge devices fast inference times are crucial due to limited computational resources and bandwidth
	Cost	Faster inference times can lead to lower operational costs by reducing the computational resources, especially in cloud-based deployments
MC	Learning and inference time	The time required to train a model depends on the algorithm's complexity or model complexity (MC), and computational demands
PD	Incremental learning	Performance degradation (PD) may signal the need for model adaptation, fine-tuning, or retraining on new data to restore or improve its accuracy
	Versioning	PD insights influence decisions about deploying new model versions to replace outdated ones
V	Model serving	Versioning (V) plays a critical role in managing the release and deployment of models to production environments
	Reproducibility	By documenting each ML model version's configuration and training data, experiments can be reproduced and results can be validated

Table B.5: Relationships between data perspective concerns.

Relationship		Description
DD	Performance metrics	Certain metrics, especially in the context of imbalanced datasets, can highlight bias in predictions. Understanding how metrics perform across different data distributions (DD) helps identify potential fairness issues
GD	Performance metrics	A golden dataset (GD) provides a reliable benchmark for evaluating the performance of ML models. It ensures that the model's accuracy, precision, recall, and other metrics are measured against a trusted standard
S	Data streaming	To implement a data streaming project it is necessary to define the data source (S)
DO	ML functionality	Data operations (DO). The selection, transformation and creation of features is influenced by the desired functionality

C

List of Publications

This thesis is based on the following publications:

- (A) Hugo Villamizar, Tatiana Escovedo, and Marcos Kalinowski. **Requirements engineering for machine learning: A systematic mapping study**. In 2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pages 29–36. IEEE, 2021.
- (B) Hugo Villamizar, Marcos Kalinowski, and Hélio Lopes. **A catalogue of concerns for specifying machine learning-enabled systems**. In 2022 25th Workshop on Requirements Engineering (WER), 2022.
- (C) Hugo Villamizar, Marcos Kalinowski, and Hélio Lopes. **Towards perspective-based specification of machine learning-enabled systems**. In 2022 48th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pages 112–115. IEEE, 2022.
- (D) Hugo Villamizar, Marcos Kalinowski, Hélio Lopes, and Daniel Mendez. **Identifying concerns when specifying machine learning-enabled systems: A perspective-based approach**. arXiv preprint arXiv:2309.07980.
- (E) Gabriel Menegon, Hugo Villamizar, Tatiana Escovedo, and Marcos Kalinowski. **Specifying a machine learning enabled system to predict the market price of steel plates using *PerSpecML*: A case study**. (Submitted).
- (F) Antonio Pedro Santos Alves, Marcos Kalinowski, Gökem Giray, Daniel Mendez, Niklas Lavesson, Kelly Azevedo, Hugo Villamizar, Tatiana Escovedo, Helio Lopes, Stefan Biffl, Jürgen Musil, Michael Felderer, Stefan Wagner, Teresa Baldassarre, Tony Gorschek. **Status quo and problems of requirements engineering for machine learning: Results from an international survey**. In 2023 24st International Conference on Product-Focused Software Process Improvement (PROFES), Dornbirn, Austria, December 10–13.
- (G) Marcos Kalinowski, Tatiana Escovedo, Hugo Villamizar, and Hélio Lopes. **Engenharia de software para ciência de dados: Um guia**

de boas práticas com ênfase na construção de sistemas de machine learning em Python. Casa do Código; 2023 May 3.

I was the main contributor regarding the design, planning, execution, and writing of this thesis, and Paper A, Paper B, Paper C, and Paper D. Paper D and E are currently under review. In Paper E, I contributed to writing the paper. In Paper F, I contributed to designing and distributing the survey and writing the paper. I collaborated as a co-author on the first book on software engineering for data science, where I incorporated the primary findings of this thesis.

During my doctoral studies, I also collaborated with other researchers at the *ExACTa Lab* with content not related to the thesis. The details of our collaborative work are outlined in the following papers.

- (H) Marcos Kalinowski, Hélio Lopes, Alex Furtado Teixeira, Gabriel da Silva Cardoso, André Kuramoto, Bruno Itagyba, Solon Tarso Batista, Juliana Alves Pereira, Thuener Silva, Jorge Alam Warrak, Marcelo da Costa, Marinho Fischer, Cristiane Salgado, Bianca Teixeira, Jacques Chueke, Bruna Ferreira, Rodrigo Lima, Hugo Villamizar, André Brandão, Simone Barbosa, Marcus Poggi, Carlos Pelizaro, Deborah Lemes, Marcus Waltemberg, Odnei Lopes, and Willer Goulart. **Lean r&d: An agile research and development approach for digital transformation.** Product-Focused Software Process Improvement: 21st International Conference, PROFES 2020, Turin, Italy, November 25–27, 2020, Proceedings 21. Springer International Publishing, 2020.
- (I) Hugo Villamizar, Marcos Kalinowski, Alessandro Garcia, and Daniel Mendez. **An efficient approach for reviewing security-related aspects in agile requirements specifications of web applications.** Requirements Engineering, 25, pp.439-468, 2020.
- (J) Raphael Oliveira, Marcos Kalinowski, Maria Teresa Baldassarre, Hugo Villamizar, Tatiana Escovedo, and Hélio Lopes. **Investigating the Impact of SOLID Design Principles on Machine Learning Code Understanding.** In 2024 3rd International Conference on AI Engineering (CAIN) – Software Engineering for AI.
- (K) Eduardo Zimelewicz, Antonio Pedro Santos Alves, Marcos Kalinowski, Tatiana Escovedo, Daniel Mendez, Görkem Giray, Kelly Azevedo, Hugo Villamizar, Maria Teresa Baldassarre, Michael Felderer, Stefan Biffel, Jürgen Musil, Stefan Wagner, Niklas Lavesson, and Tony Gorschek. **ML-Enabled Systems Model Deployment and Monitoring: Status Quo and Problems.** In 2024 Software Quality Days Conference (SWQD).

- (L) Gabriel Busquim, Hugo Villamizar, Maria Julia Lima, and Marcos Kalinowski. **On the Interaction between Software Engineers and Data Scientists when building Machine Learning-Enabled Systems**. In 2024 Software Quality Days Conference (SWQD).