The Journal of Systems and Software 213 (2024) 112053

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

Identifying concerns when specifying machine learning-enabled systems: A perspective-based approach☆

Hugo Villamizar [a],[*], Marcos Kalinowski [a], Hélio Lopes [a], Daniel Mendez [b]

[a] Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, Brazil
[b] Blekinge Institute of Technology, Karlskrona, Sweden

ARTICLE INFO

Keywords:
Requirements engineering
Machine learning-enabled systems
Technology transfer
Case study

ABSTRACT

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 paper, we present *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 creation of *PerSpecML* involved a series of formative evaluations 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*.

Editor's note: Open Science material was validated by the Journal of Systems and Software Open Science Board.

1. Introduction

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. In this paper, we refer to them as ML-enabled systems, and their behavior is dictated by both 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); for instance, challenges related to covering quality properties such as fairness and explainability (Habibullah et al., 2023), or challenges related to collaboration and mismatched assumptions in ML projects given the required multidisciplinary teams (Nahar et al., 2022; Lewis et al., 2021). These particularities typically demand extra effort to successfully develop ML-enabled systems. It is, therefore, not surprising to us that Gartner reports only 53% of ML projects to make it into production (Gartner, 2020).

Within SE, Requirements Engineering (RE) is, in simple terms, the discipline that is meant to effectively translate stakeholder needs into requirements, constraints, and other information that defines what software systems should do under which conditions (Damian, 2007). 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, 2020). 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 conventional software systems where we need to identify representative stakeholders, validate specifications with customers, and address conflicting requirements. Nevertheless, establishing RE in ML projects can pose

☆ Editor: Heiko Koziolek.
[*] Corresponding author.
E-mail addresses: hvillamizar@inf.puc-rio.br (H. Villamizar), kalinowski@inf.puc-rio.br (M. Kalinowski), lopes@inf.puc-rio.br (H. Lopes), daniel.mendez@bth.se (D. Mendez).

https://doi.org/10.1016/j.jss.2024.112053
Received 25 August 2023; Received in revised form 18 December 2023; Accepted 2 April 2024
Available online 6 April 2024

challenges primarily due to (i) the inherent complexity of ML-enabled systems (ii) the interdisciplinary nature of ML involving collaboration among diverse teams (Nahar et al., 2022), (iii) the absence of a dedicated 'requirements engineer' role (Herrmann, 2013; Wang et al., 2018; Alves et al., 2023), and (iv) the perceived difficulty of RE activities in the context of ML projects (Ishikawa and Yoshioka, 2019; Kuwajima et al., 2020; Nahar et al., 2023; Alves et al., 2023).

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 and Niu, 2020). For instance, studies have emerged related to issues with data requirements (Challa et al., 2020), the understanding of the RE process in ML projects (Vogelsang and Borg, 2019), and non-functional requirements such as explainability, transparency and fairness (Habibullah et al., 2023; Cysneiros et al., 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 (de Souza 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 et al., 2021; Pei et al., 2022; Ahmad et al., 2023b). As a consequence, many ML-enabled systems lack requirements specifications (Lwakatare et al., 2019; Kuwajima et al., 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, we participated in real ML projects of a research and development (R&D) initiative (Kalinowski et al., 2020) and conducted a literature review on RE for ML (Villamizar et al., 2021) to better understand the intersection between these disciplines. This comprehensive approach, that followed a technology transfer model (Gorschek et al., 2006), which is recommended to foster successful transfer of technology from research to practice (Wohlin et al., 2012), allowed us to formulate an initial catalog comprising 45 distinct concerns (Villamizar et al., 2022), from which we proposed a candidate solution for specifying ML-enabled systems (Víllamizar et al., 2022). In this paper, we iteratively evaluated and improved both the catalog and the proposed solution by conducting three formative evaluations 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 an 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 resulted in *PerSpecML*, an approach for specifying ML-enabled systems that expands upon the initial catalog, now encompassing an analysis of 60 concerns related to 28 tasks that practitioners typically face in ML projects when defining and structuring this type of systems. These tasks are grouped 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. Throughout the evaluations, we 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 and usefulness.

The remainder of this paper is organized as follows. Section 2 presents the background and related work. In Section 3, we present *PerSpecML*. In Section 4, we detail the methodology used to conceive, evaluate, and evolve *PerSpecML*. In Sections 5 and 6, we describe the evaluation in academia and with industry representatives. Section 7 reports on the industrial case studies. Section 8 and Section 9 raise potential threats to validity and discuss our research findings. Finally, in Section 10, we conclude the paper, highlighting the main contributions, limitations and future work.

2. Background and related work

This section introduces a background on the core essence of ML and presents particularities and challenges when engineering ML-enabled systems that RE may address. We also describe related work.

2.1. ML projects

ML is the study of computer algorithms that explores data to determine the best way to combine the information contained in the representation (training data) into a model that generalizes to data it has not already seen (Mitchell et al., 1997). This component, unlike non-ML, base its behavior on external data instead of explicitly programming hard rules. However, data may not be adequate and lead to bad outcomes. The output of an ML model is a prediction, sometimes surprisingly accurate and sometimes surprisingly inaccurate. Technically, this supposes a change in the way of designing, developing and testing these type of systems.

From the viewpoint of management, formulating problems in ML projects is an uncertain and difficult process, since it requires various forms of discretionary work to translate high-level objectives or strategic goals into tractable problems, necessitating, among other things, the identification of appropriate target variables and proxies (Passi and Barocas, 2019). It is well-known that 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. As a result, ML projects are often executed as small proof-of-concept efforts that eventually make it into production. In this setting, a multidisciplinary team is required to complement the data science and engineering skills.

2.2. Quality of ML-enabled systems

Assuring the quality of ML-enabled systems is essential since these systems are increasingly becoming part of our daily life. However, this is not an easy task. Their quality goes beyond ML model performance metrics such as accuracy, precision or recall. Typically, these ML model performance metrics comprise the primary goal of data scientists during ML model development. However, this component is only a small part of a larger system (Sculley et al., 2015), which usually comprises other components for data collection, model consumption and infrastructure requirements. When designing an ML-enabled system it is important to understand the particular constraints on its operation. For example, where will the model run? What data will it have access to? How fast does it need to be? What is the business impact of a wrong prediction? How should the model be tuned to maximize business results? 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 et al., 2020; Ishikawa and Yoshioka, 2019; Belani et al., 2019). Thinking about possible strategies to address these concerns increases the chance of designing and developing an ML-enabled system that meets customer's needs.

2.3. RE for ML-enabled systems

Requirements Engineering (RE) constitutes approaches to understand the problem space and specify requirements that meet the stakeholders' needs. As such, it is concentrates on understanding what the actual problem is 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). RE is often considered a crucial and challenging stage of any software project. Indeed, most of the problems in software systems with and without ML components come from poor requirements rather than faulty implementation.

Practitioners argue that the incorporation of ML implies addressing additional qualities, setting more ambitious goals, dealing with a high degree of iterative experimentation, and facing more unrealistic assumptions (Nahar et al., 2023). It is therefore reasonable to assume that handling and resolving validation problems is (or should be) in scope of the role of a requirements engineer. We further argue that investing in RE can help to identify and mitigate problems early on. Nevertheless, establishing RE may be difficult due to the lack of guidance, tools, and techniques to support the engineering of ML-enabled systems (Ahmad et al., 2023b; Villamizar et al., 2021). It is not surprising that ML-enabled systems are rarely built based on comprehensive specifications (Lwakatare et al., 2019; Lewis et al., 2021) and that RE is seen by practitioners as the most difficult phase in ML projects (Ishikawa and Yoshioka, 2019; Kuwajima et al., 2020; Nahar et al., 2023).

Requirements specifications and ML have a special connection. For instance, Kästner (2020) states that 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-enabled system might be incorrect, even if the learning algorithm is implemented correctly. In SE, the specification is divided into software and system constraints. By analogy, the specification of an ML-enabled systems should be tailored to accommodate the unique characteristics of ML models, as well as the requirements for their seamless operation and maintenance within a broader system context. In this line, Berry (2022) 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 ML-enabled systems.

In the last years, the literature on RE for ML has focused on issues with data requirements (Challa et al., 2020), process of data-driven projects (Vogelsang and Borg, 2019), challenges of addressing non-functional requirements and particularities of certain quality attributes such as explainability, transparency and fairness (Habibullah et al., 2023; Cysneiros et al., 2018; Martínez-Fernández et al., 2022). Despite the important 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 (de Souza Nascimento et al., 2019), and only a limited number of studies have looked into how to specify and document requirements for ML-enabled systems (Pei et al., 2022; Villamizar et al., 2021; Ahmad et al., 2023b, 2021).

2.4. Related work

We subsequently highlight research that has investigated what quality attributes should be analyzed and how practitioners can specify and document requirements for ML-enabled systems. We further take a more holistic RE perspective where an ML model is merely part of a larger ML-enabled 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) play an important role and can make the difference between an ML-enabled system that fits customer's needs and one that does not.

In an effort to model a representation of data-driven systems, several works have been proposed. For instance, Chuprina et al. (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 behavior 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 et al. (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 modeling 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 facilitate effective 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 analysis and verification 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. (2023a) 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 do not match such attributes with the stakeholders who should be in charge. We consider these features can support the collaboration and communication between stakeholders.

From the industry perspective, several frameworks on human-centered AI development have been proposed by big tech companies, specifically Google's PAIR guidebook (Google Corporation, 2021), Apple's human interface guidelines for building ML applications (Apple Inc, 2020), and Microsoft's eighteen guidelines for human-centered AI interaction (Microsoft Corporation, 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 is 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.

Despite these remarkable contributions, 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, a more holistic RE approach should be used to write requirements for ML-enabled systems. *PerSpecML* advances over existing approaches by:

- Enabling the identification of requirements to model, specify and document ML-enabled systems.
- Allowing practitioners of ML projects to support the design of ML-enabled systems for addressing business, user, system, model, and data requirements.
- Modeling the link from business strategies and customer expectations to user preferences, model constraints, data preparations, and system operations.
- Modeling the operation and strategy knowledge of ML-enabled systems in the form of tasks and related concerns that are expressed in conceptual diagrams.
- Providing empirical evaluation of the proposed approach to strengthen the contributions and facilitate their adoption in practice.

3. *PerSpecML*

In this section we present *PerSpecML*, a perspective-based approach for specifying ML-enabled systems that involves analyzing 60 concerns related to typical tasks that practitioners face in ML projects when defining and structuring these software 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 the ML-enabled system. 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 an overview of ML-enabled systems concerns and their relationships, practitioners can make informed decisions 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.

Fig. 1 shows 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.

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 how to apply *PerSpecML*.

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 the software system. We also need to be concerned about how to improve the experience of end-users in

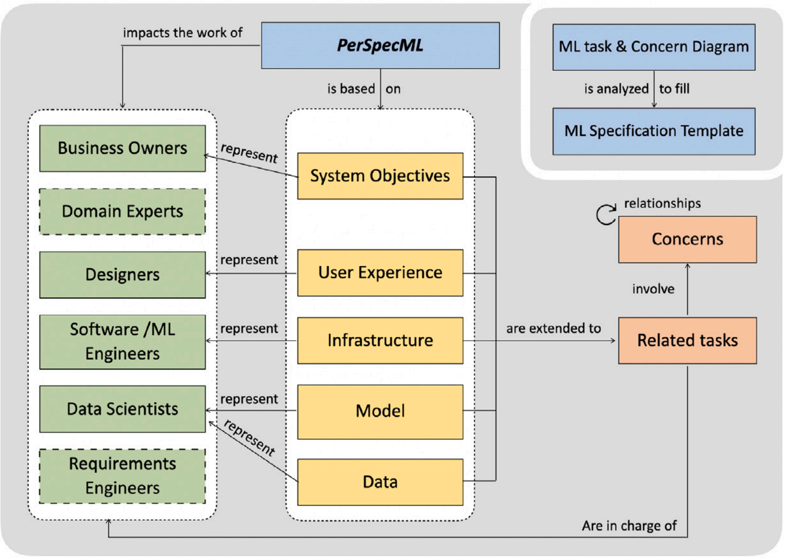


Fig. 1. An overview of the elements comprising *PerSpecML*.

order to deal with unrealistic assumptions, and align business value to ML technical activities in order to cover business requirements. Given this, our goal is that *PerSpecML* impacts 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 should business objectives be connected? What is the real cost involved in maintaining an ML-enabled system? What team and skills are needed to successfully build 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 interpret 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 will an ML outcome appear? How often will it appear? And also 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? Are online or batch predictions 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 should the ML model consider? What domain restrictions may apply? What should the complexity of the ML model be? Also, 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.

3.2. Concerns

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 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 components, the challenge is further amplified since it incorporates models that make predictions based on patterns and trends learned from data, which introduce unique considerations. All of these considerations, including ML components and deterministic (non-ML) components, become concerns for practitioners in charge of designing an ML-enabled system.

One of the main elements of *PerSpecML* are its concerns. In total, we identified 60 concerns including, for example, 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. The concerns, that can be seen as quality attributes, came from (i) own experiences as active participants in real ML projects, from (ii) literature reviews on RE for ML that have researched both academia and industry, and from (iii) practitioners who iteratively evaluated the concerns and recommended new ones to be considered. In *PerSpecML*, the concerns are part of tasks that stakeholders typically face throughout the development of ML-enabled systems.

Table 1
Description of the tasks to define the system objectives.

Task	Description
Understand the problem	understand the problem domain and the real-world 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 goals of the ML project at different levels in order to ensure that it meets the expectations of the stakeholders
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, and providing clarity will prevent disappointment and frustration

Table 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 (*e.g.*, frequency and forcefulness) in order to design user-friendly interfaces and workflows
Visualize predictions	present ML model outputs in a visually understandable format. Visual aids such as charts, and graphs can help users comprehend complex data and insights
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 the ML model's capabilities and potential inaccuracies

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 hyperparameter 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 1 details the tasks from a system objectives perspective.

A positive user experience is crucial for the successful adoption, acceptance, and utilization of ML-enabled systems. It enhances user engagement, improves user satisfaction, and ultimately contributes to the overall success of the ML project. Table 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 3 details the tasks of the infrastructure perspective.

A structured ML model development process fosters transparency, reproducibility, and accountability. It supports the creation of robust, reliable, and trustworthy ML solutions. Table 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 details the tasks to be done, mainly by data scientists and domain experts, to maintain high-quality data throughout the lifecycle of ML projects.

3.4. Perspectives

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 of different concerns and stakeholders' viewpoints. Perspectives have been effectively used in SE to model scenarios where team members work on a particular phenomena (Basili and Rombach, 1988). In *PerSpecML*, we modeled five perspective that are detailed as follows.

System Objectives Perspective: ML projects often have as much to do with the formulation of the problem as any property of the resulting ML model (Passi and Barocas, 2019). 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 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 ML trade-off to deal with customer expectations. Table 6 details the concerns when thinking on objectives for ML-enabled systems.

User Experience Perspective: A good ML-enabled system includes building better experiences of using ML. The goal of this perspective

Table 3
Description of the tasks to support the infrastructure of ML-enabled systems.

Task	Description
Transport data to the model	involves moving the relevant data from its source to the ML model for analysis, training, or prediction
Make the ML model available	refers to the process of deploying and exposing the trained ML model so that it can be accessed for making predictions
Update the ML model	refers to the process of making improvements or modifications to an existing ML model to enhance its performance
Store ML artifacts	involves the systematic storage and management of various artifacts generated throughout the ML development process
Observe the ML model	involves analyzing the performance, behavior, and outcomes of both the ML model and the software system
Automate End-to-End ML workflow	involves the design and implementation of a systematic and streamlined process that automates the ML workflow, from data preparation to model deployment and monitoring
Integrate the ML model	involves incorporating the trained ML model into the larger software system where it will be used for making predictions
Evaluate the financial cost involved with infrastructure	assess and analyze the expenses related to the computational resources, hardware, software, and services required to support the ML project

Table 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 combinations of hyperparameters to find the optimal configuration that yields the best performance
Train the ML model	create a ML model that captures the underlying patterns in the data and can make predictions on unseen examples
Validate the ML model	ensure that the trained ML model meets the desired criteria
Deploy the ML model	make the trained ML model available and 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 accuracy. Other quality characteristics are equally important for the model's overall performance, reliability, and suitability for real-world applications

Table 5
Description of the tasks to support data quality in ML projects.

Task	Description
Collect 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, validate, and test 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 in order to ensure that the data meets certain criteria and standards to produce accurate and 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 available data into separate subsets for training, validation, and testing purposes
Define a golden dataset	involves creating a high-quality dataset that represents the problem domain and serves as the ground truth for training and evaluating ML models

is to present the predictions of the ML model to users in a way that achieves the 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 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

Table 6
Description of each concern of the system objectives perspective.

Id	Concern	Addressing this concern involves specifying
O1	Context	the specific circumstances, environment, or conditions in which the ML-enabled system will operate
O2	Need	the requirement, desire, or gap that must be addressed to achieve a particular set of circumstances within a given context
O3	ML functionality	the nature of the learning problem and the desired outcome that the ML model is designed to achieve (*e.g.*, 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. *E.g.*, increase the revenue in X%, increase the number of units sold in Y%, number of trees saved
O6	System goals	what the system tries to achieve, with the support of an ML model, in terms of behavior or quality
O7	User goals	what the users want to achieve by using ML. *E.g.*, for recommendation systems this could involve helping users find content they will enjoy
O8	Model goals	metrics and acceptable measures the model should achieve (*e.g.*, for classification problems this could involve accuracy > X%, precision > Y%, recall > Z%)
O9	Leading indicators	measures correlating with future success, from the business' perspective. This includes the users' affective states when using the ML-enabled system (*e.g.*, customer sentiment and engagement)
O10	ML trade-off	the balance of customer expectations (*e.g.*, inference time vs accuracy, false positive vs false negative)

Table 7
Description of each concern of the user experience perspective.

Id	Concern	Addressing this concern involves specifying
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 (*e.g.*, automatic or assisted actions)
U3	Frequency	how often the system interacts with users (*e.g.*, whenever the user asks for it or whenever the system thinks the user will respond)
U4	Visualization	user-friendly interfaces to showcase the ML model's outputs and facilitate its integration into the customer's existing systems (*e.g.*, specifying dashboard and visualization prototypes for validation)
U5	Learning feedback	what interactions the users will have with the ML-enabled system in order to provide new data for learning, or human-in-the-loop systems where ML models require human interaction
U6	Acceptance	how well and how the 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-enabled system and how to interpret its outputs

model and data; the strategy to automate ML operations that allow to reproduce and maintain ML artifacts, and the integration the ML model will have with the rest of the system functionality. Table 8 details the concerns when thinking on the infrastructure for ML-enabled systems.

Model Perspective: Building an 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 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 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 10 details the concerns when thinking on data for ML-enabled systems.

Table 8
Description of each concern of the infrastructure perspective.

Id	Concern	Addressing this concern involves specifying
I1	Data streaming	what data streaming strategy will be used (*e.g.*, real time data transportation or in batches)
I2	Model serving	how the ML model will be executed and consumed (*e.g.*, client-side, back-end, cloud-based, web service end-point)
I3	Incremental learning	the need for ML-enabled system abilities to continuously learn from new data, extending the existing model's knowledge
I4	Storage	where the ML artifacts (*e.g.*, models, data, scripts) will be stored
I5	Monitorability	the need to monitor the data and the outputs of the ML model to alert/detect when data drifts or changes
I6	Telemetry	what ML-enabled system data needs to be collected. Telemetry involves collecting data such as clicks on particular buttons and could involve other usage data
I7	Reproducibility	the need to repeatedly run an algorithm/ML process on certain datasets/experiments and obtain the same (or similar) results
I8	Maintainability	the need to modify ML-enabled systems to improve performance or adapt to a changed environment
I9	Integration	the integration that the model will have with the rest of the system functionality (*e.g.*, safety, security, privacy, fairness, legal)
I10	Hybrid decision intelligence	the essence of combining ML model outputs with rule-based or heuristic decision-making to create a more comprehensive and effective decision intelligence
I11	Cost	the financial cost involved in executing the inferences and with the infrastructure that could affect architectural decisions. Great models can be unusable due to the cost to run and maintain them

Table 9
Description of each concern of the model perspective.

Id	Concern	Addressing this concern involves specifying
M1	Algorithm & model selection	the set of algorithms that could be used/investigated, based on the ML problem and other concerns to be considered (*e.g.*, constraints regarding explainability or model performance, for instance, can limit the solution options)
M2	Algorithm tuning	the need to choose a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a parameter whose value is used to control the learning process
M3	Input & Output	the expected inputs (features) and outcomes of the model. Of course, the set of meaningful inputs can be refined/improved during pre-processing activities, such as feature selection
M4	Learning time	the acceptable time to train the model
M5	Performance metrics	the metrics used to evaluate the model (*e.g.*, precision, recall, F1-score, mean square error) and measurable performance expectations
M6	Baseline model	the optional simple model that acts as a reference. Its main function is 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 (*e.g.*, for decision trees there might be needs for pruning)
M9	Performance degradation	the awareness of performance degradation. Over time many models' predictive performance decreases as a given model is tested on new datasets within rapidly evolving environments
M10	Versioning	the versions of libraries, ensuring compatibility, and handling any conflicts or issues that may arise due to dependencies. This is important for maintaining reproducibility, portability, and ensuring that the ML model can be easily set up and executed on different systems
M11	Interpretability & Explainability	the need to understand reasons for the model inferences. The model might need to be able to summarize the reasons for its decisions. Other related concerns such as transparency, may apply
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

Table 10
Description of each concern of the data perspective.

Id	Concern	Addressing this concern involves specifying
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 type of the problem and the complexity of the algorithm
D6	Accuracy	the need to get correct data
D7	Completeness	the need to get data containing sufficient observations of all situations where the model will operate
D8	Credibility	the need to get true data that is believable and understandable by users
D9	Real usage	the need to get real data representing the real problem
D10	Bias	the need to get data fair samples and representative distributions
D11	Consistency	the need to get consistent data in a specific context
D12	Ethics	the need to get data to prevent adversely impacting society (*e.g.*, listing potential adverse impacts to be avoided)
D13	Anonymization	the need to anonymize or pseudonymize to protect individual identities while still maintaining the utility of the data for ML purposes
D14	Data operations & Modeling	what operations must be applied on the data (e.g., data cleaning and labeling) 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, validating and test data
D16	Golden dataset	the need for a baseline dataset approved by a domain expert that reflects the problem. It is employed to monitor other data acquired afterwards

3.5. *Relationship between concerns*

Identifying relationships that show influence and implications between the concerns of an ML-enabled system is of paramount importance 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 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 are required to know the reasons why the ML model made decisions then the explainability & interpretability concern arises. However, 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 perspective. For instance, defining the source to collect 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 the relationships between concerns can be found in our online repository.[1]

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 for approval and requirements management, we present the Perspective-Based ML Task and Concern Diagram that integrates the key components discussed earlier: concerns, tasks, perspectives, and stakeholders. Table 11 shows the notation we used to represent these components in the diagram.

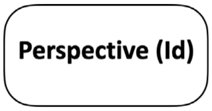
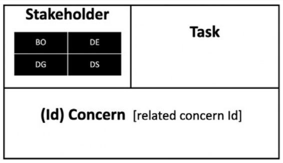
The perspective-based ML task and concern diagram shown in Fig. 2 serves as a visual representation of the interplay between these components and their relationships within the context of ML projects. It offers a comprehensive overview of how different perspectives shape the tasks at hand, while considering 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.

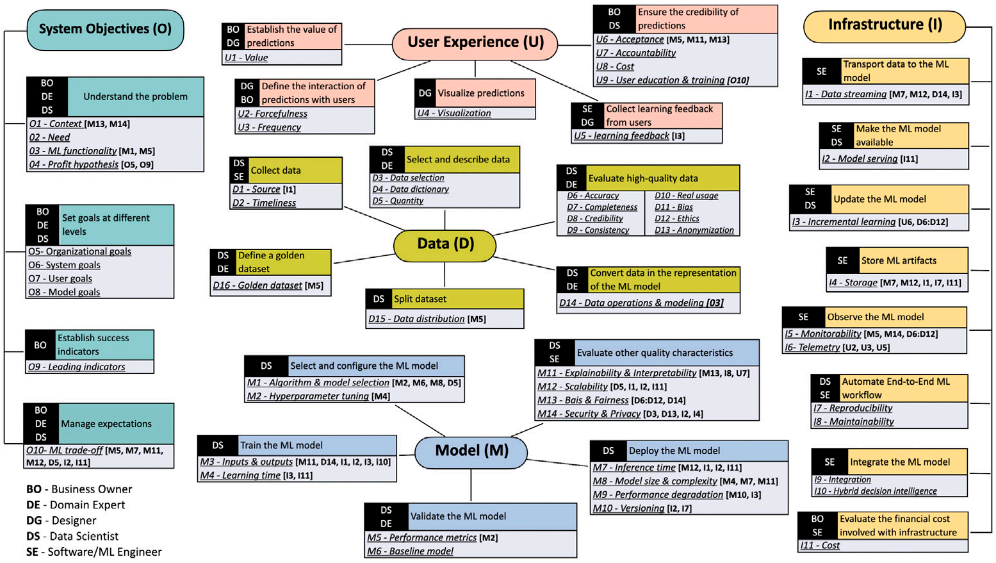
3.7. *Perspective-based ML specification template*

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

[1] https://doi.org/10.5281/zenodo.7705002

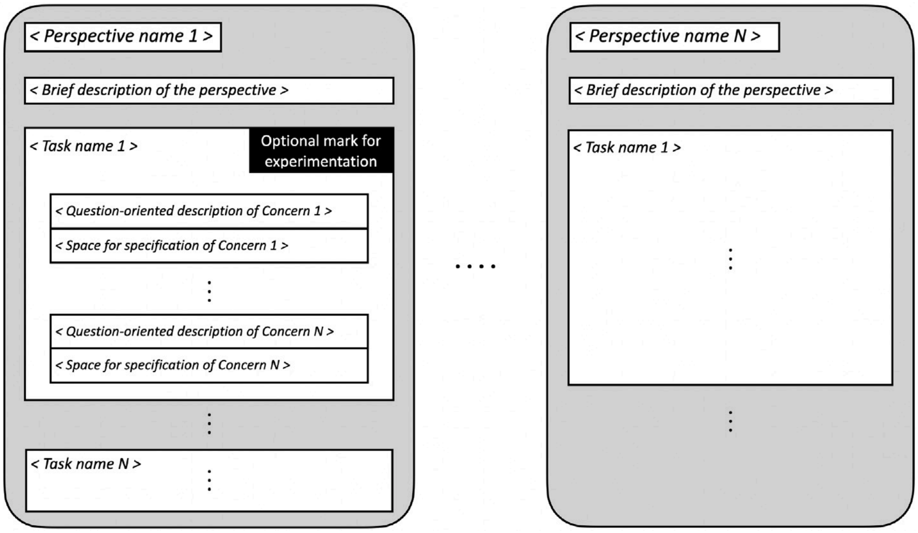
Table 11
Legend of the perspective-based ML task and concern diagram.

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
 Stakeholder BO DE DG DS Task (Id) Concern [related concern Id]	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



System Objectives (O)
BO DE DS Understand the problem
O1 - Context [M13, M14]
02 - Need
03 - ML functionality [M1, M5]
04 - Profit hypothesis [O5, O9]
BO DE DS Set goals at different levels
O5- Organizational goals
O6- System goals
O7 - User goals
O8 - Model goals
BO Establish success indicators
O9 - Leading indicators
BO DE DS Manage expectations
O10- ML trade-off [M5, M7, M11, M12, D5, I2, I11]
User Experience (U)
BO DG Establish the value of predictions
U1 - Value
DG BO Define the interaction of predictions with users
U2- Forcefulness
U3 - Frequency
DG Visualize predictions
U4 - Visualization
BO DS Ensure the credibility of predictions
U6 - Acceptance [M5, M11, M13]
U7 - Accountability
U8 - Cost
U9 - User education & training [O10]
SE DG Collect learning feedback from users
U5 - learning feedback [I3]
Data (D)
DS SE Collect data
D1 - Source [I1]
D2 - Timeliness
DS DE Select and describe data
D3 - Data selection
D4 - Data dictionary
D5 - Quantity
DS DE Evaluate high-quality data
D6 - Accuracy
D7 - Completeness
D8 - Credibility
D9 - Consistency
D10 - Real usage
D11 - Bias
D12 - Ethics
D13 - Anonymization
DS DE Define a golden dataset
D16 - Golden dataset [M5]
DS Split dataset
D15 - Data distribution [M5]
DS DE Convert data in the representation of the ML model
D14 - Data operations & modeling [03]
Model (M)
DS Select and configure the ML model
M1 - Algorithm & model selection [M2, M6, M8, D5]
M2 - Hyperparameter tuning [M4]
DS SE Evaluate other quality characteristics
M11 - Explainability & Interpretability [M13, I8, U7]
M12 - Scalability [D5, I1, I2, I11]
M13 - Bais & Fairness [D6:D12, D14]
M14 - Security & Privacy [D3, D13, I2, I4]
DS Train the ML model
M3 - Inputs & outputs [M11, D14, I1, I2, I3, i10]
M4 - Learning time [I3, I11]
DS Deploy the ML model
M7 - Inference time [M12, I1, I2, I11]
M8 - Model size & complexity [M4, M7, M11]
M9 - Performance degradation [M10, I3]
M10 - Versioning [I2, I7]
DS DE Validate the ML model
M5 - Performance metrics [M2]
M6 - Baseline model
Infrastructure (I)
SE Transport data to the ML model
I1 - Data streaming [M7, M12, D14, I3]
SE DS Make the ML model available
I2 - Model serving [I11]
SE DS Update the ML model
I3 - Incremental learning [U6, D6:D12]
SE Store ML artifacts
I4 - Storage [M7, M12, I1, I7, I11]
SE Observe the ML model
I5 - Monitorability [M5, M14, D6:D12]
I6- Telemetry [U2, U3, U5]
DS SE Automate End-to-End ML workflow
I7 - Reproducibility
I8 - Maintainability
SE Integrate the ML model
I9 - Integration
I10 - Hybrid decision intelligence
BO SE Evaluate the financial cost involved with infrastructure
I11 - Cost
BO - Business Owner
DE - Domain Expert
DG - Designer
DS - Data Scientist
SE - Software/ML Engineer

Fig. 2. An illustration of the perspective-based ML task and concern diagram.



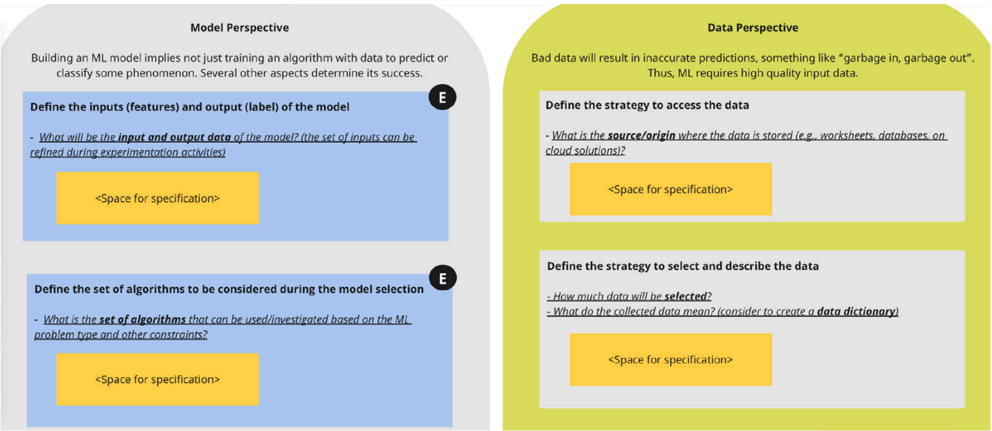
< Perspective name 1 >
< Brief description of the perspective >
< Task name 1 >
Optional mark for experimentation
< Question-oriented description of Concern 1 >
< Space for specification of Concern 1 >
< Question-oriented description of Concern N >
< Space for specification of Concern N >
< Task name N >
< Perspective name N >
< Brief description of the perspective >
< Task name 1 >

Fig. 3. Elements of the Perspective-Based ML Specification Template.

template (Villamizar et al., 2023) 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 Fig. 3.

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

1. **Perspective name:** Positioned at the top of the template, this element identifies the specific perspective under consideration.



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

- *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

- *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>

Fig. 4. Perspective-Based ML Specification Template for model and data perspectives.

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 name:** 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 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 allows 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 hyper-parameter 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.

The Perspective-Based ML Specification Template, which includes an illustrative example, is published online at Miro (Villamizar et al., 2023). Fig. 4 showcases an excerpt of the Perspective-Based ML Specification Template published in Miro, showing parts of the model and data perspectives.

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. Fig. 5 shows the workflow to ensure that *PerSpecML* is executed in a systematic and organized manner, leading to more successful outcomes.

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 the analysis to the provided 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 concern.
   - 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).

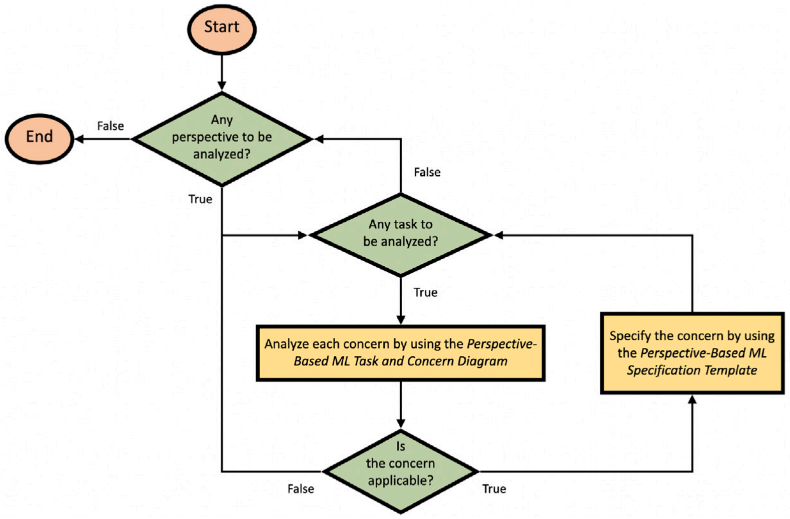

Start
Any perspective to be analyzed?
False
End
True
False
Any task to be analyzed?
True
Analyze each concern by using the *Perspective-Based ML Task and Concern Diagram*
Specify the concern by using the *Perspective-Based ML Specification Template*
Is the concern applicable?
False
True

Fig. 5. Workflow to execute *PerSpecML*.

Table 12
Specification of the concerns of the system objectives perspective.

Concern	Specification
Context	Customer feedback plays a significant role in the e-commerce industry. 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 in an automated manner in order to optimize the resources and 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 customer reviews
Organizational goals	Enhancing customer satisfaction and loyalty, Improving brand reputation and perception, increasing customer retention and reducing churn, and gaining competitive advantage in the market
System goals	Scalability to handle large volumes of customer reviews, real-time processing of reviews, ensuring customer data privacy, easy integration with existing systems and workflows
User goals	Promptly identifying and addressing customer concerns, understanding customer sentiment towards specific products, monitoring brand perception, 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, and mitigating biases and ensuring fairness in sentiment analysis
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

3.9. Application example of PerSpecML

This section provides a demonstration of how *PerSpecML* can be applied to systems that incorporate an ML component. For illustrative purposes, we explain the rationale of analyzing the concerns of the system objectives and the infrastructure perspectives. Another complete application example covering all the perspectives is also included in the online template published at Miro (Villamizar et al., 2023). 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 format with additional requirement descriptions to better understand the specification context.

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

Additional requirements description:

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* workflow in two perspectives.

The first step involves analyzing 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 application in this document, we show the specifications of these concerns in Table 12.

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 (see the Perspective-Based ML Task and Concern Diagram). The specification of these concerns holds the potential to impact various system aspects. For instance, the specification of *'Context'* influences *'ethical'* considerations within the data perspective, since critical domains such as medical diagnosis and financial predictions need to be carefully designed to avoid biases and unfair outcomes. Similarly, *'ML functionality'* specifications influence concerns within the model perspective, such as *Algorithm & 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 and measured.

After completing the analysis of concerns within the system objectives perspective, the subsequent perspective to be examined is the user experience perspective. However, we opt to break down the infrastructure perspective, given it encompasses a diverse set of concerns, making the *PerSpecML* application example more comprehensive. The specifications of the concerns of this perspective are detailed in Table 13.

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.

4. Methodology for conceiving *PerSpecML*

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 formative 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. Fig. 6 outlines the seven steps of the model, which we will describe sequentially hereafter (while following the terminology of the transfer model).

4.1. Step 1: Identify improvement areas based on industry needs

We followed the principle of constructivism (Fosnot, 2013) that advocates that a person needs to understand how something works before exploring the different ways to construct solution proposals. During the last four years, the first author has participated in research and development (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 (i) how important the domain and business settings are to align the stakeholder needs, requirements, and constraints with the engineering and data science activities, (ii) interdisciplinary teams typically involved in ML projects, and (iii) the lack of tools and documents that can capture key components when specifying ML-enabled systems.

4.2. 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 systematic mapping study on RE for ML (Villamizar et al., 2021), analyzed later literature reviews (Pei et al., 2022; Ahmad et al., 2023b, 2021) 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, (i) additional quality attributes of ML-enabled systems that practitioners should analyze (ii) the lack of studies focused on identifying key components of ML-enabled systems that may later be specified, and (iii) the lack of studies evaluated in practice to validate its effectiveness, feasibility and gather user feedback.

Table 13
Specification of the concerns of the infrastructure perspective.

Concern	Specification
Data streaming	The system shall set up mechanisms to ingest and process streaming data in real-time or near-real-time, ensuring timely analysis and response
Model serving	The system shall set up scalable and performant serving endpoints to handle high-volume prediction traffic and ensure low-latency responses
Incremental learning	The system shall provide the services to adapt and improve the ML model over time as new labeled data becomes available
Storage	Storage containers are needed for storing raw and processed data, model checkpoints, configuration files, and any other relevant data
Monitorability	The system shall set up monitoring tools and logging mechanisms to capture system metrics, errors, latency, throughput, and other relevant indicators for proactive maintenance and issue resolution
Telemetry	The system shall gather and analyze telemetry data such as user interactions of the sentiment analysis solution
Reproducibility	The system shall ensure the ML model is reproducible across different environments and deployments
Maintainability	The system shall follow software engineering best practices, modularizing the codebase, and providing clear documentation to enable efficient troubleshooting, bug fixing, and system enhancements
Integration	Establishing appropriate APIs and service interfaces is needed to enable seamless communication and interaction between the sentiment analysis component and the larger system
Hybrid decision intelligence	The system shall use the ML outputs of the ML model to create heuristics that reflect the context of the problem
Cost	The infrastructure necessary to execute and maintain the ML model must not exceed the budget allocated for it

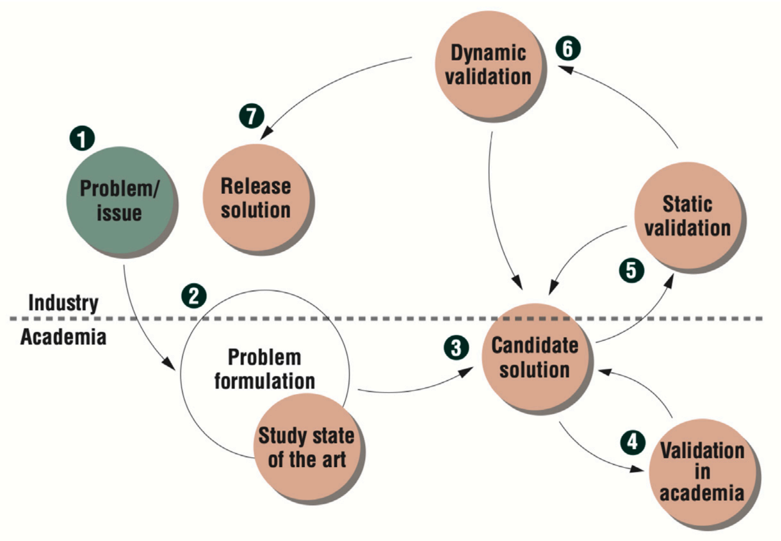


Fig. 6. Technology transfer model proposed by Gorschek et al. (2006).

4.3. 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 design of ML-enabled systems. As a first step, we proposed a catalog of 45 concerns to be analyzed by practitioners with the aim at identifying key components of ML-enabled systems (Villamizar et al., 2022). The 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 concerns as prominent, and allowed us to identify initial improvements. Fig. 7 shows the catalog.

Therefrom, we used this catalog to create a candidate solution for specifying ML-enabled systems (Víllamizar et al., 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 the ML-enabled system. Fig. 8 shows the diagram we proposed in a first effort to specify ML-enabled systems.

In this paper, 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 3.

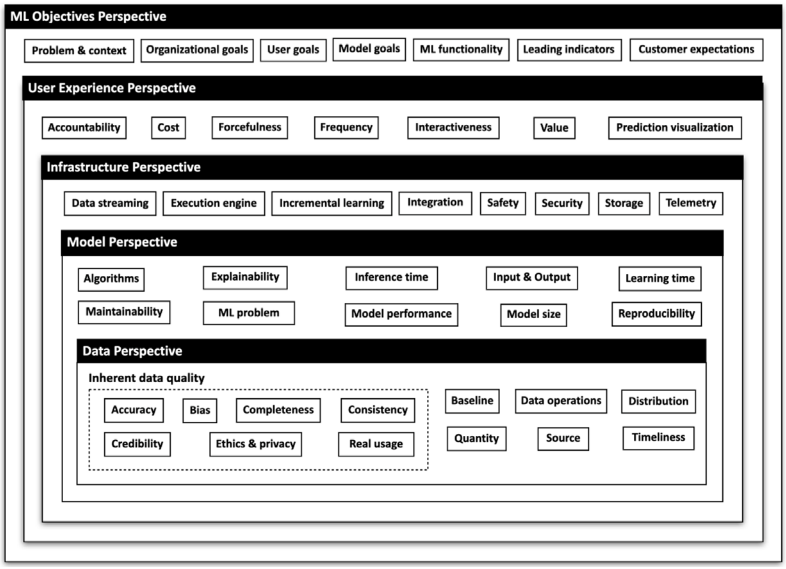


Fig. 7. Initial catalog of concerns (Villamizar et al., 2022).

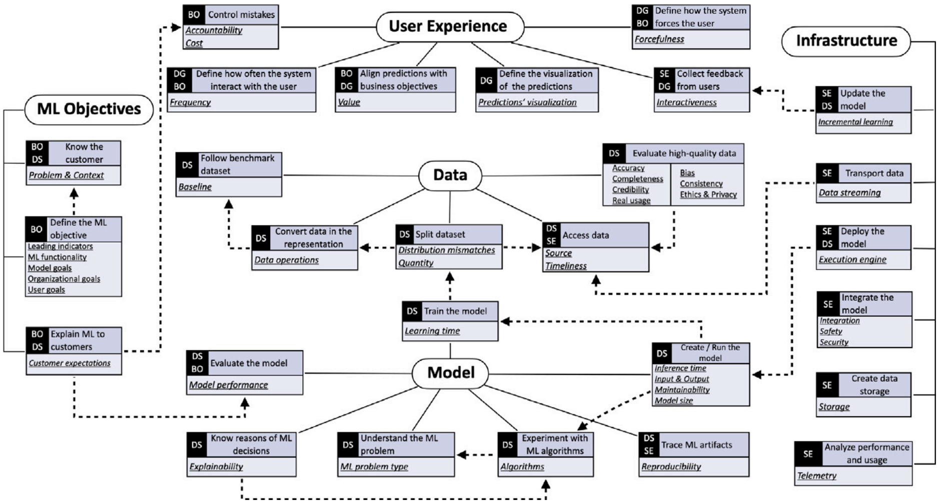


Fig. 8. Initial diagram for specifying ML-enabled systems (Villamizar et al., 2022).

4.4. 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 the technology transfer model (Gorschek et al., 2006): (i) with students from two courses on SE for data science specifying an ML-enabled system for a toy scenario (validation in academia), (ii) with practitioners working in a R&D initiative discussing specifications of ML-enabled systems built retroactively with stakeholders of real projects (static validation), and (iii) in two industrial case studies conducted with an e-commerce company, specifying real ML-enabled systems from scratch using the approach (dynamic validation). Note that, according to Gorschek 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, 6, and 7.

4.5. 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. 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 4.3 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.

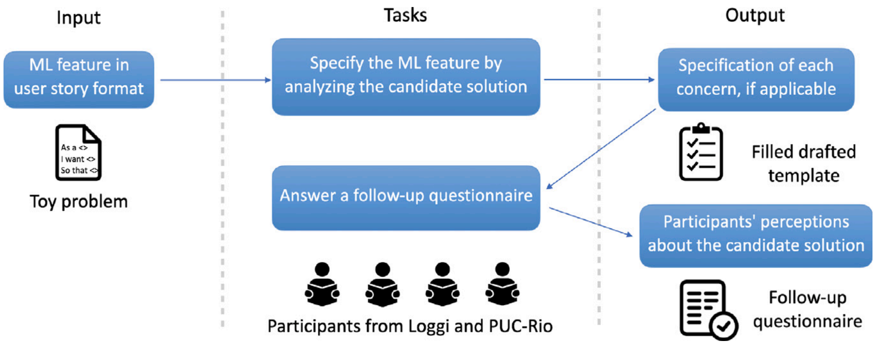

Input
Tasks
Output
ML feature in user story format
As a <>
I want <>
So that <>
Toy problem
Specify the ML feature by analyzing the candidate solution
Answer a follow-up questionnaire
Participants from Loggi and PUC-Rio
Specification of each concern, if applicable
Filled drafted template
Participants' perceptions about the candidate solution
Follow-up questionnaire

Fig. 9. Process diagram for the academic validation.

5.1. Context

The academic validation took place in the context of two courses on SE for data science with professionals, who are also students, from a Brazilian logistic company called Loggi[2] (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 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. On the other hand, for online participants, the time spent on the study was not regulated. All the material provided to the participants is available in our online repository.[1] Fig. 9 illustrates the academic validation.

5.2. Goal and method

We detail the goal of the validation in academia in Table 14. We followed the Goal-Question-Metric (GQM) goal definition template (Basili and 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.

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 do 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.

5.3. 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 15, we characterized the subjects by their educational background and average year of experience in ML projects.

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

[2] https://www.loggi.com

Table 14
Study goal definition of academic validation.

Analyze for the purpose of	the candidate solution's perspectives and concerns 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

Table 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

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. 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 one author to summarize the quantitative data. At the end, three authors 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, one author 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 authors, 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.

5.5. Results

5.5.1. 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. Fig. 10 presents the relevance of the candidate solution' perspectives based on their respective ratings.

We expected 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.

5.5.2. 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 dividing the explainability concern into both explainability and interpretability. 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"

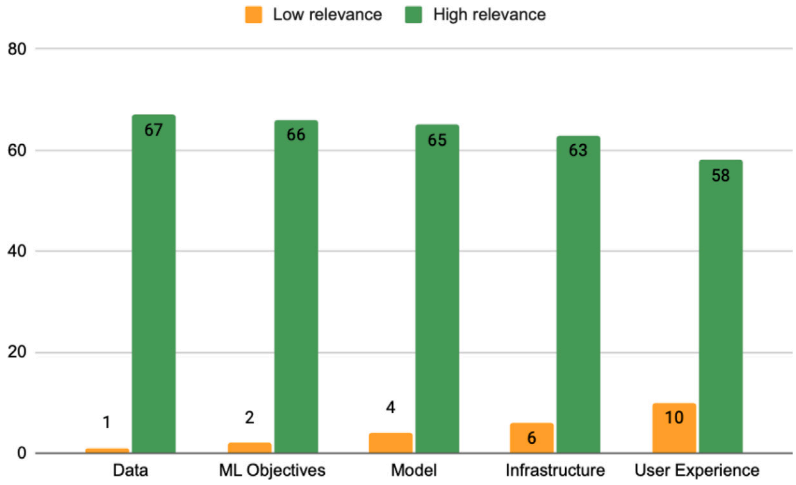


Fig. 10. Frequencies of the relevance of each perspective of the candidate solution.

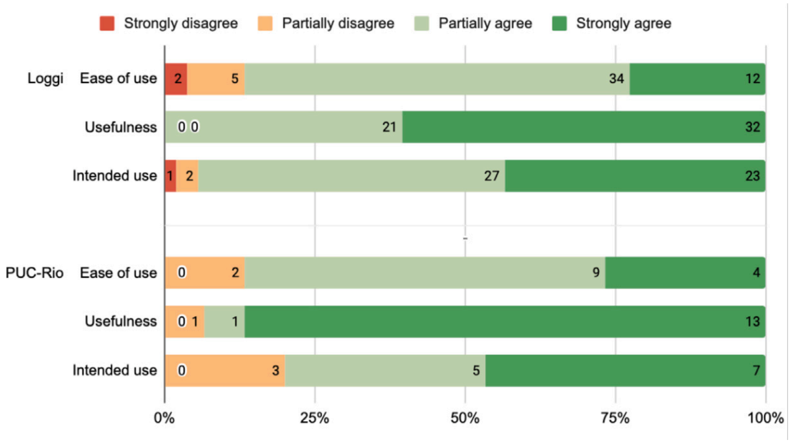


Fig. 11. Frequencies of the TAM constructs for academic validation.

> "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"

5.5.3. RQ3. To what extent do 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 Fig. 11.

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' guidance will imply an improvement in the perception of intended use.

5.5.4. 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"

Table 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

> "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

6. Static validation in industry

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 used *PerSpecML v1* to retroactively specify two ready-made ML projects. We referred to this evaluation as static since it was performed without executing *PerSpecML v1* in a real or simulated environment.

6.1. Context

The static validation in industry involved practitioners of a R&D initiative called *ExACTa*[3] 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 named. Table 16 details these projects.

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 one of the authors 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 repository.[1] Fig. 12 shows the process diagram for the static validation in industry.

6.2. Goal and method

We detail the goal of the static validation in Table 17. We followed the GQM template to describe what we evaluated in this first industrial validation. Here, we also describe the research questions.

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 and enhanced *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.

6.3. Selection of subjects

We invited six practitioners who have been actively working with the development of ML-enabled systems in the *ExACTa* initiative. The practitioners were selected based on their roles 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 18 shows an overview of the participant characterization.

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

[3] https://exacta.inf.puc-rio.br

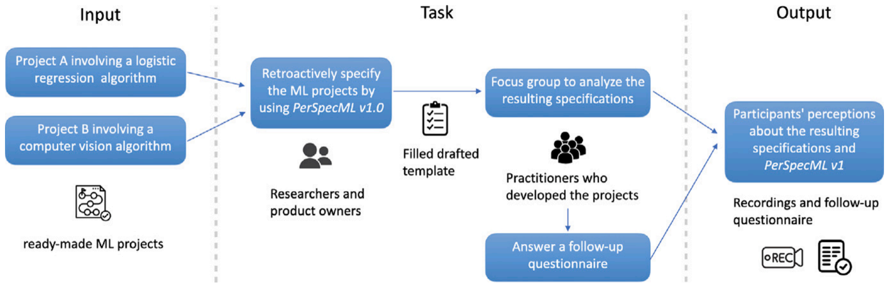

Input
Task
Output
Project A involving a logistic regression algorithm
Project B involving a computer vision algorithm
ready-made ML projects
Retroactively specify the ML projects by using *PerSpecML v1.0*
Researchers and product owners
Filled drafted template
Focus group to analyze the resulting specifications
Practitioners who developed the projects
Answer a follow-up questionnaire
Participants' perceptions about the resulting specifications and *PerSpecML v1*
Recordings and follow-up questionnaire
REC

Fig. 12. Process diagram for the static validation in industry.

Table 17
Study goal definition of the static validation.

Analyze	*PerSpecML v1* (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 *PerSpecML v1* with six experienced software practitioners involved in the development of these systems

Table 18
Subjects involved in the static validation in industry.

Id	Position	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

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.

6.4. 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.

6.4.1. Focus group

We conducted a focus group for promoting in-depth discussion on *RQ1* and *RQ2* (Kontio et al., 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 one 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 one author 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 authors. In the end, the final set of categories were analyzed by three authors to address the research questions. The transcriptions and all codes are available in our online repository.[1] Examples of codes are highlighted when presenting the results.

6.4.2. 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.

6.5. Results

6.5.1. 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*

6.5.2. 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.*

6.5.3. 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*

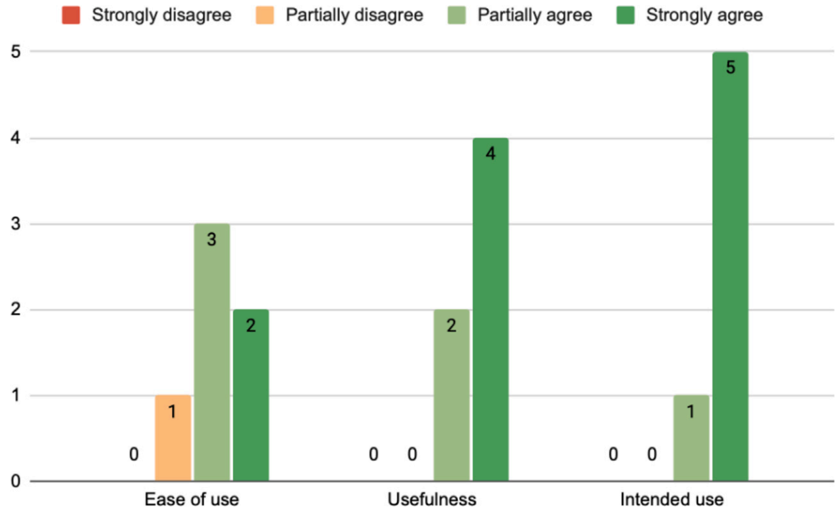


Fig. 13. Frequencies of the TAM constructs for static validation industry.

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.

6.5.4. 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 Fig. 13.

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**[4] 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 and their concerns
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

7. Dynamic validation in industry

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 executing *PerSpecML v2* for specifying two real ML projects from scratch.

7.1. 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 19.

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

[4] https://miro.com/miroverse/perspecml-machine-learning/

Table 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

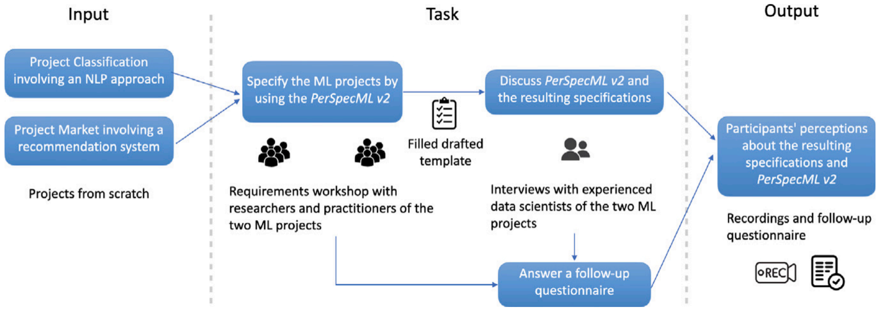

Input
Task
Output
Project Classification involving an NLP approach
Project Market involving a recommendation system
Projects from scratch
Specify the ML projects by using the *PerSpecML v2*
Filled drafted template
Requirements workshop with researchers and practitioners of the two ML projects
Discuss *PerSpecML v2* and the resulting specifications
Interviews with experienced data scientists of the two ML projects
Answer a follow-up questionnaire
Participants' perceptions about the resulting specifications and *PerSpecML v2*
Recordings and follow-up questionnaire
REC

Fig. 14. Process diagram for the dynamic validation in industry.

Table 20
Study Goal Definition of the Dynamic Validation.

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

specifications it generated. All mentioned artifacts are available in our online repository.[1] Fig. 14 shows the process diagram for the dynamic validation in industry.

7.2. Goal and method

We detail the goal of the case studies of the dynamic validation in Table 20. We followed the GQM template to describe what we evaluated in this second industrial validation. Here, we also describe the research questions.

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 and insights of 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 the novice

Table 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

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.

7.3. 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. The scope of the projects involved both novice practitioners and 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 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.

The subjects involved in specifying the ML-enabled systems from scratch were the real participants of the two project teams. 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, which 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 gray 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).

7.4. 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.

7.4.1. 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: Two authors analyzed the recorded workshop sessions. Subsequently, the same authors 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.

7.4.2. 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 the 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 allows for later review and analysis.

Data Analysis: One author 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 authors 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.

7.4.3. 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 confidentiality. 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.

7.5. Results

7.5.1. 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.

7.5.2. 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.

7.5.3. 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.

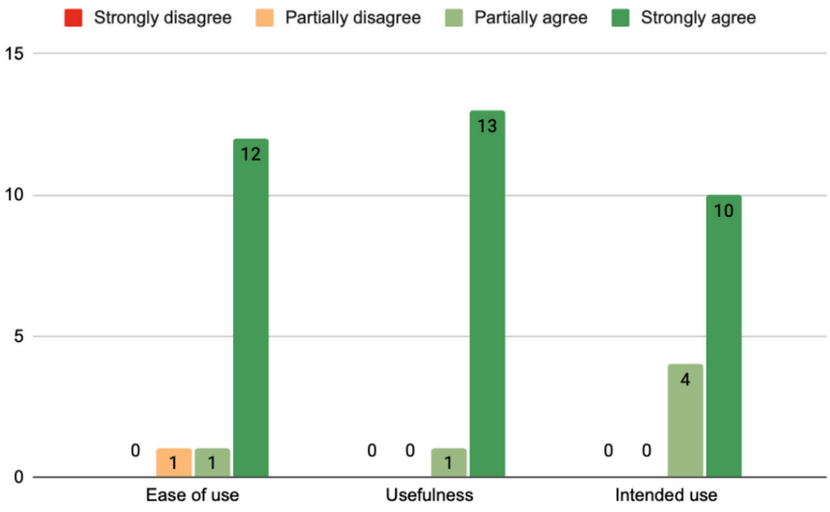


Fig. 15. Frequencies of the TAM constructs for dynamic validation in industry.

> “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.

7.5.4. *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 Fig. 15.

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.

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, ‘**anonymization**’ in the data perspective, and ‘**hybrid decision intelligence**’ in the infrastructure 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. The second one will help to protect sensitive data when required while still maintaining the utility of the data for ML purposes, and the last one will help to provide context to the outputs of ML models

8. 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 human factors, such as practitioners' perceptions and experiences. In the following, we critically examine 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 (Service, 2009), a widely used qualitative research method for identifying, analyzing, and reporting patterns. We also peer-reviewed each analysis step and made our data available online for auditing to improve reliability.

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. In addition, according to Falessi et al. (2018), using students as subjects remains a valid simplification of real-life settings commonly applied 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 authors reviewed and discussed the list of codes attached to the transcriptions. Two of these authors brought diverse expertise in SE, system architecture, and qualitative research methods, while the other one brought expertise in data science. This helped to ensure a comprehensive analysis. In addition, as suggested by Kontio et al. (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.

9. 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 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. However, we recognize that further evaluations should be conducted involving more exhaustive real-world scenarios.

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.

10. Concluding remarks

The design and development of ML-enabled systems has proven to be challenging. Despite remarkable contributions in the field, many organizations continue to struggle with specifying such systems. In this paper 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 formative evaluations 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 SE courses 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. Lastly, 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.

In the following, we conclude our work by explicitly highlighting the contributions, acknowledging the limitations, and outlining avenues for future development in the context of *PerSpecML*.

10.1. Contributions

Through this work, we provided a catalog of concerns and tasks, a conceptual model, a specification template, and an available package, allowing *PerSpecML* to be applied.

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 available these resources in Miroverso.[5]

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-enabled systems.

10.2. 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 8. 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). 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 i^* and GORE, which have been reported to be more challenging to learn among non-software engineers (Neace et al., 2018; Dimitrakopoulos et al., 2019).

Another limitation of this work lies in the scope of the catalog of concerns used for conceiving *PerSpecML*. The content of the catalog of concerns 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

[5] https://miro.com/miroverse/perspecml-machine-learning/

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.

10.3. 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 10.2 and expanding the approach's capabilities to accommodate a broader range of ML paradigms and complex requirements.

CRediT authorship contribution statement

Hugo Villamizar: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Marcos Kalinowski:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Resources, Supervision, Validation, Writing – original draft, Writing – review & editing. **Hélio Lopes:** Conceptualization, Investigation, Methodology, Project administration, Supervision, Validation, Writing – original draft, Writing – review & editing. **Daniel Mendez:** Conceptualization, Investigation, Methodology, Project administration, Supervision, Validation, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

I have shared the link to my data as a footnote in the article.

Acknowledgments

We would like to thank the employees of Loggi, of the ExACTa Initiative at PUC-Rio and of Americanas S.A. Thanks also for the financial support of the Brazilian CAPES and CNPq agencies (grant 312827/2020-2).

References

Ahmad, Khlood, Abdelrazek, Mohamed, Arora, Chetan, Baniya, Arbind Agrahari, Bano, Muneera, Grundy, John, 2023a. Requirements engineering framework for human-centered artificial intelligence software systems. Appl. Soft Comput. 143, 110455.

Ahmad, Khlood, Abdelrazek, Mohamed, Arora, Chetan, Bano, Muneera, Grundy, John, 2023b. Requirements engineering for artificial intelligence systems: A systematic mapping study. Inf. Softw. Technol. 107176.

Ahmad, Khlood, Bano, Muneera, Abdelrazek, Mohamed, Arora, Chetan, Grundy, John, 2021. What's up with requirements engineering for artificial intelligence systems? In: 2021 IEEE 29th International Requirements Engineering Conference. RE, IEEE, pp. 1–12.

Aho, Timo, Sievi-Korte, Outi, Kilamo, Terhi, Yaman, Sezin, Mikkonen, Tommi, 2020. Demystifying data science projects: A look on the people and process of data science today. In: Product-Focused Software Process Improvement: 21st International Conference, PROFES 2020, Turin, Italy, November 25–27, 2020, Proceedings 21. Springer, pp. 153–167.

Alves, Antonio Pedro Santos, Kalinowski, Marcos, Giray, Görkem, Mendez, Daniel, Lavesson, Niklas, Azevedo, Kelly, Villamizar, Hugo, Escovedo, Tatiana, Lopes, Helio, Biffl, Stefan, et al., 2023. Status quo and problems of requirements engineering for machine learning: Results from an international survey. In: International Conference on Product-Focused Software Process Improvement. Springer, pp. 159–174.

Apple Inc, 2020. Human-interface-guidelines for ML solutions.

Barash, Guy, Farchi, Eitan, Jayaraman, Ilan, Raz, Orna, Tzoref-Brill, Rachel, Zalmanovici, Marcel, 2019. 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. pp. 1048–1058.

Basili, Victor R., Rombach, H. Dieter, 1988. The TAME project: Towards improvement-oriented software environments. IEEE Trans. Softw. Eng. 14 (6), 758–773.

Belani, Hrvoje, Vukovic, Marin, Car, Željka, 2019. Requirements engineering challenges in building AI-based complex systems. In: International Requirements Engineering Conference Workshops. REW, pp. 252–255.

Berry, Daniel M., 2022. Requirements engineering for artificial intelligence: What is a requirements specification for an artificial intelligence? In: International Working Conference on RE: Foundation for Software Quality. pp. 19–25.

Carleton, Anita D, Harper, Erin, Robert, John E, Klein, Mark H, De Niz, Dionisio, Desautels, Edward, Goodenough, John B, Holland, Charles, Ozkaya, Ipek, Schmidt, Douglas, et al., 2021. Architecting the Future of Software Engineering: A National Agenda for Software Engineering Research and Development. Technical Report, Carnegie-Mellon University Pittsburgh.

Challa, Harshitha, Niu, Nan, Johnson, Reese, 2020. Faulty requirements made valuable: on the role of data quality in deep learning. In: 2020 IEEE Seventh International Workshop on Artificial Intelligence for Requirements Engineering. AIRE, IEEE, pp. 61–69.

Chuprina, Tatiana, Mendez, Daniel, Wnuk, Krzysztof, 2021. Towards artefact-based requirements engineering for data-centric systems. In: 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, vol. 2857, CEUR-WS.

Cysneiros, Luiz Marcio, Raffi, Majid, do Prado Leite, Julio Cesar Sampaio, 2018. Software transparency as a key requirement for self-driving cars. In: 2018 IEEE 26th International Requirements Engineering Conference. RE, IEEE, pp. 382–387.

Dalpiaz, Fabiano, Niu, Nan, 2020. Requirements engineering in the days of artificial intelligence. IEEE Softw. 37 (4), 7–10.

Damian, Daniela, 2007. Stakeholders in global requirements engineering: Lessons learned from practice. IEEE Softw. 24 (2), 21–27.

Davis, Fred D., 1989. Perceived usefulness, ease of use, and user acceptance of information technology. MIS Q. 319–340.

de Souza Nascimento, Elizamary, Ahmed, Iftekhar, Oliveira, Edson, Palheta, Márcio Piedade, Steinmacher, Igor, Conte, Tayana, 2019. Understanding development process of ML systems: Challenges and solutions. In: Empirical Software Engineering and Measurement. ESEM, pp. 1–6.

Dimitrakopoulos, George, Kavakli, Evangelia, Loucopoulos, Peri, Anagnostopoulos, Dimosthenis, Zographos, Theodoros, 2019. A capability-oriented modelling and simulation approach for autonomous vehicle management. Simul. Model. Pract. Theory 91, 28–47.

Dorard, Louis, 2015. Machine learning canvas.

Falessi, Davide, Juristo, Natalia, Wohlin, Claes, Turhan, Burak, Münch, Jürgen, Jedlitschka, Andreas, Oivo, Markku, 2018. Empirical software engineering experts on the use of students and professionals in experiments. Empir. Softw. Eng. 23, 452–489.

Fosnot, Catherine Twomey, 2013. Constructivism: Theory, Perspectives, and Practice. Teachers College Press.

Gartner, 2020. Gartner identifies the top strategic technology trends for 2021.

Google Corporation, 2021. People + AI guidebook.

Gorschek, Tony, Garre, Per, Larsson, Stig, Wohlin, Claes, 2006. A model for technology transfer in practice. IEEE 23 (6), 88–95.

Habibullah, Khan Mohammad, Gay, Gregory, Horkoff, Jennifer, 2023. Non-functional requirements for machine learning: Understanding current use and challenges among practitioners. Requir. Eng. 1–34.

Herrmann, Andrea, 2013. Requirements engineering in practice: There is no requirements engineer position. In: Requirements Engineering: Foundation for Software Quality: 19th International Working Conference, REFSQ 2013, Essen, Germany, April 8-11, 2013. Proceedings 19. Springer, pp. 347–361.

Hulten, Geoff, 2019. Building Intelligent Systems. Springer.

Ishikawa, Fuyuki, Yoshioka, Nobukazu, 2019. How do engineers perceive difficulties in engineering of machine-learning systems? In: International Workshop on Conducting Empirical Studies in Industry. CESI, pp. 2–9.

ISO/IEC, 2011. ISO/IEC 25010:Systems and software quality requirements and evaluation (SQuaRE) — System and software quality models.

ISO/IEC, 2012. ISO/IEC 25012: Software engineering – software product quality requirements and evaluation (SQuaRE) – data quality model.

Kalinowski, Marcos, Lopes, Hélio, Teixeira, Alex Furtado, da Silva Cardoso, Gabriel, Kuramoto, André, Itagyba, Bruno, Batista, Solon Tarso, Pereira, Juliana Alves, Silva, Thuener, Warrak, Jorge Alam, et al., 2020. Lean r&d: An agile research and development approach for digital transformation. In: International Conference on Product-Focused Software Process Improvement. Springer, pp. 106–124.

Kästner, Christian, 2020. Machine learning is requirements engineering—on the role of bugs, verification, and validation in machine learning. Medium post, Accessed April 25.

Kim, Miryung, Zimmermann, Thomas, DeLine, Robert, Begel, Andrew, 2017. Data scientists in software teams: State of the art and challenges. IEEE Trans. Softw. Eng. 44 (11), 1024–1038.

Kontio, Jyrki, Lehtola, Laura, Bragge, Johanna, 2004. Using the focus group method in software engineering: obtaining practitioner and user experiences. In: Proceedings. 2004 International Symposium on Empirical Software Engineering, 2004. ISESE'04, IEEE, pp. 271–280.

Kuwajima, Hiroshi, Yasuoka, Hirotoshi, Nakae, Toshihiro, 2020. Engineering problems in ML systems. Mach. Learn. 109 (5), 1103–1126.

Lewis, Grace A., Bellomo, Stephany, Ozkaya, Ipek, 2021. Characterizing and detecting mismatch in machine-learning-enabled systems. In: 2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI. WAIN, IEEE, pp. 133–140.

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

Maffey, Katherine R, Dotterrer, Kyle, Niemann, Jennifer, Cruickshank, Iain, Lewis, Grace A, Kästner, Christian, 2023. Mlteing models: negotiating, evaluating, and documenting model and system qualities. In: 2023 IEEE/ACM 45th International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER). IEEE, pp. 31–36.

Martínez-Fernández, Silverio, Bogner, Justus, Franch, Xavier, Oriol, Marc, Siebert, Julien, Trendowicz, Adam, Vollmer, Anna Maria, Wagner, Stefan, 2022. Software engineering for AI-based systems: a survey. ACM Trans. Softw. Eng. Methodol. (TOSEM) 31 (2), 1–59.

Microsoft Corporation, 2022. Microsoft's framework for building AI systems responsibly.

Mitchell, Tom M., et al., 1997. Mach. Learn. 45 (37), 870–877.

Nahar, Nadia, Zhang, Haoran, Lewis, Grace, Zhou, Shurui, Kästner, Christian, 2023. 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, pp. 171–183.

Nahar, Nadia, Zhou, Shurui, Lewis, Grace, Kästner, Christian, 2022. Collaboration challenges in building ml-enabled systems: Communication, documentation, engineering, and process. In: Proceedings of the 44th International Conference on Software Engineering. pp. 413–425.

Nakamichi, Koji, Ohashi, Kyoko, Namba, Isao, Yamamoto, Rieko, Aoyama, Mikio, Joeckel, Lisa, Siebert, Julien, Heidrich, Jens, 2020. Requirements-driven method to determine quality characteristics and measurements for machine learning software and its evaluation. In: 2020 IEEE 28th International Requirements Engineering Conference. RE, IEEE, pp. 260–270.

Nalchigar, Soroosh, Yu, Eric, Keshavjee, Karim, 2021. Modeling machine learning requirements from three perspectives: a case report from the healthcare domain. Requir. Eng. 26 (2), 237–254.

Neace, Kerry, Roncace, Robert, Fomin, Pavel, 2018. Goal model analysis of autonomy requirements for unmanned aircraft systems. Requir. Eng. 23, 509–555.

Passi, Samir, Barocas, Solon, 2019. Problem formulation and fairness. In: Proceedings of the Conference on Fairness, Accountability, and Transparency. pp. 39–48.

Pei, Zhongyi, Liu, Lin, Wang, Chen, Wang, Jianmin, 2022. Requirements engineering for machine learning: A review and reflection. In: 2022 IEEE 30th International Requirements Engineering Conference Workshops. REW, IEEE, pp. 166–175.

Rahimi, Mona, Guo, Jin L.C., Kokaly, Sahar, Chechik, Marsha, 2019. Toward requirements specification for machine-learned components. In: 2019 IEEE 27th International Requirements Engineering Conference Workshops. REW, IEEE, pp. 241–244.

Sculley, David, Holt, Gary, Golovin, Daniel, Davydov, Eugene, Phillips, Todd, Ebner, Dietmar, Chaudhary, Vinay, Young, Michael, Crespo, Jean-Francois, Dennison, Dan, 2015. Hidden technical debt in machine learning systems. Adv. Neural Inf. Process. Syst. 28.

Service, Robert W., 2009. Book review: Corbin, J., & Strauss, A. (2008). Basics of qualitative research: Techniques and procedures for developing grounded theory. Organ. Res. Methods 12 (3), 614–617, Thousand Oaks, CA: Sage.

Siebert, Julien, Joeckel, Lisa, Heidrich, Jens, Trendowicz, Adam, Nakamichi, Koji, Ohashi, Kyoko, Namba, Isao, Yamamoto, Rieko, Aoyama, Mikio, 2022. Construction of a quality model for machine learning systems. Softw. Qual. J. 30 (2), 307–335.

Turner, Mark, Kitchenham, Barbara, Brereton, Pearl, Charters, Stuart, Budgen, David, 2010. Does the technology acceptance model predict actual use? A systematic literature review. Inf. Softw. Technol. 52 (5), 463–479.

Villamizar, Hugo, Escovedo, Tatiana, Kalinowski, Marcos, 2021. Requirements engineering for machine learning: A systematic mapping study. In: 2021 47th Euromicro Conference on Software Engineering and Advanced Applications. SEAA, IEEE, pp. 29–36.

Villamizar, Hugo, Kalinowski, Marcos, Lopes, Hélio, 2022. A catalogue of concerns for specifying machine learning-enabled systems. In: 2022 25th Workshop on Requirements Engineering. WER.

Víllamizar, Hugo, Kalinowski, Marcos, Lopes, Hélio, 2022. Towards perspective-based specification of machine learning-enabled systems. In: 2022 48th Euromicro Conference on Software Engineering and Advanced Applications. SEAA, IEEE, pp. 112–115.

Villamizar, Hugo, Kalinowski, Marcos, Lopes, Helio, 2023. PerSpecML - machine learning - online published miro template.

Vogelsang, Andreas, Borg, Markus, 2019. Requirements engineering for machine learning: Perspectives from data scientists. In: 2019 IEEE 27th International Requirements Engineering Conference Workshops. REW, IEEE, pp. 245–251.

Wagner, Stefan, Fernández, Daniel Méndez, Felderer, Michael, Vetrò, Antonio, Kalinowski, Marcos, Wieringa, Roel, Pfahl, Dietmar, Conte, Tayana, Christiansson, Marie-Therese, Greer, Desmond, et al., 2019. Status quo in requirements engineering: A theory and a global family of surveys. ACM Trans. Softw. Eng. Methodol. (TOSEM) 28 (2), 1–48.

Wang, Chong, Cui, Pengwei, Daneva, Maya, Kassab, Mohamad, 2018. 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. pp. 1–10.

Wohlin, Claes, Runeson, Per, Höst, Martin, Ohlsson, Magnus C, Regnell, Björn, Wesslén, Anders, 2012. Experimentation in Software Engineering. Springer Science & Business Media.

Hugo Villamizar is a software engineer and researcher of the ExACTa initiative (Agile Experimentation, Co-Creation, and Digital Transformation).

Marcos Kalinowski is an Associate Professor of Software Engineering at the Department of Informatics at Pontifical Catholic University of Rio de Janeiro (PUC-Rio) and coordinator of the ExACTa initiative (Agile Experimentation, Co-Creation, and Digital Transformation).

Hélio Lopes is an Associate Professor of Data Science at the Department of Informatics at Pontifical Catholic University of Rio de Janeiro (PUC-Rio) and coordinator of the ExACTa initiative (Agile Experimentation, Co-Creation, and Digital Transformation).

Daniel Mendez is Full Professor in Empirical Software Engineering at the Blekinge Institute of Technology, Sweden, and head of the Requirements Engineering research division at the fortiss research and transfer institute, Germany.