Thesis for The Degree of Doctor of Philosophy

Non-Functional Requirements for Machine Learning Systems

Khan Mohammad Habibullah

Division of Interaction Design & Software Engineering
Department of Computer Science & Engineering
Chalmers University of Technology and Gothenburg University
Gothenburg, Sweden, 2025

Non-Functional Requirements for Machine Learning Systems

Khan Mohammad Habibullah

ISBN 978-91-8115-395-8 (PRINT)
ISBN: 978-91-8115-396-5 (PDF)

Department of Computer Science & Engineering
Division of Interaction Design & Software Engineering
Chalmers University of Technology and Gothenburg University
Gothenburg, Sweden

This thesis has been prepared using LaTeX.
Printed by Chalmers Digitaltryck,
Gothenburg, Sweden 2025.

"Knowledge enlivens the soul."
- Imam Ali (RA)

Abstract

Background: Machine learning (ML) systems are increasingly being deployed in complex and safety-critical domains such as autonomous driving, healthcare, and finance. ML systems learn using big data and solve a wide range of prediction and decision-making problems that would be difficult to solve with traditional systems. However, increasing use of ML in different systems has raised concerns about quality requirements, which are defined as non-functional requirements (NFRs). Many NFRs, such as fairness, transparency, explainability, and safety, are critical in ensuring the success and acceptance of ML systems. However, many NFRs for ML systems are not well understood (e.g., maintainability), some known NFRs may become more important (e.g., fairness), while some may become irrelevant in the ML context (e.g., modularity), some new NFRs may come into play (e.g., retrainability), and the scope of defining and measuring NFRs in ML systems is also a challenging task.

Objective: The research project focuses on addressing and managing issues related to NFRs for ML systems. The objective of the research is to identify current practices and challenges related to NFRs in an ML context, and to develop solutions to manage NFRs for ML systems.

Method: This research follows a design science methodology and consists of a series of empirical and design-oriented studies. First, we conducted an interview study to explore practitioners' perceptions of NFRs and the challenges associated with defining and measuring them in ML systems. Then we conducted a subsequent survey study to validate and expand these findings with broader practitioner input. To complement these studies, we conducted a partial systematic mapping study to assess the coverage of NFRs in the academic literature, revealing discrepancies between research focus and industrial needs. Additionally, we conducted group interviews with domain experts in the automotive industry to uncover requirements engineering (RE) practices and challenges specific to ML-enabled perception systems. Based on these insights, we proposed a structured, five-step quality framework and evaluated it through practitioner interviews. Finally, we proposed revised maintainability metrics adapted to the unique structure of ML systems, and we evaluated them using ten real-world open-source ML projects.

Findings: We found that NFRs are crucial and play an important role in the success of the ML systems. However, there is a research gap in this area, and managing NFRs for ML systems is challenging. To address the research objectives, we have identified important NFRs for ML systems, such as accuracy, reliability, fairness, transparency, retrainability, and explainability. We also identified challenges in defining, scoping, and measuring NFRs, including domain dependence, lack of standardized metrics, and difficulty in tracing NFRs across ML system components. Furthermore, we found that practitioners face significant challenges in applying RE to ML systems—particularly in autonomous perception—due to uncertainty, evolving components, and lack of systematic approaches for managing quality trade-offs, data quality, and cross-organizational collaboration. To address these gaps, we proposed a five-

step NFR management framework, covering NFR selection, scoping, trade-off analysis, measurement planning, and structured specification using templates. Finally, given that maintainability is an important NFR for ML systems, we proposed scope-aware definitions and measurement strategies for maintainability in ML systems and demonstrated their usefulness through empirical evaluation.

Conclusion: NFRs are critical for ML systems, but they are difficult to define, allocate, specify, and measure due to challenges like unintended bias, non-deterministic behavior, and the high cost of thorough testing. Industry and research lack well-structured solutions to manage NFRs for ML systems effectively. This research addresses this critical gap by providing a comprehensive understanding of NFRs and the unique challenges they pose in the ML context. Through a combination of empirical studies and the development of a structured NFR management framework, this research offers a solution for identifying, prioritizing, scoping, measuring, and specifying NFRs across granular-level components of ML systems. Contributions also include scope-aware definitions and measurement metrics of maintainability for ML systems. These findings enrich the theoretical understanding of NFRs for ML systems, provide empirically grounded insights into their challenges, and introduce artifacts and metrics to support future research. These outcomes also provide valuable guidance for practitioners to build trustworthy, maintainable, and high-quality ML systems. This research will help practitioners make better engineering decisions, improve quality assurance processes, and provide a foundation for more systematic and accountable ML system development.

Keywords

Non-functional Requirements, NFRs, Machine Learning, Quality Requirements, Requirements Engineering

Acknowledgment

First and foremost, I am grateful to my main supervisor, Jennifer Horkoff, for her patience, motivation, support, and immense knowledge, which have been invaluable throughout my Ph.D. journey. I am also grateful to my co-supervisor, Gregory Gay, for his kind support, insightful feedback, and advice that helped me develop my research acumen. I feel lucky to have them as my supervisors, and it is my pleasure working with them.

Next, I would like to express my sincere gratitude to my examiner, Prof. Jan Bosch, and to the other members of the Ph.D. school for their valuable constructive feedback and their generous administrative support throughout this process.

I would also like to express my heartfelt gratitude to all my colleagues in the IDSE division for the wonderful time we have shared and for their unfailing kindness and support. Each of them has brought something unique and valuable into my life, and I am truly fortunate to have been surrounded by such inspiring, dedicated, and compassionate people. Their encouragement, humor, and camaraderie have made even the most challenging days enjoyable, and their friendship has been a constant source of motivation. They are more than just colleagues to me—they are the best companions and friends one could hope for, and I will always cherish the memories we have created together.

Most importantly, I am blessed to have my mother back home. I am in this position in my life and career because of her. I could never have come to this position without my mother, who is my courage and strength. I am also thankful to all my relatives and friends for their immense support, help, and good wishes.

Finally, my deepest gratitude goes to my beloved wife, Rejwana Siddiq. I am truly thankful for her unwavering love, endless support, and remarkable patience throughout this journey. Her sacrifices and positive energy have been my greatest strength, and I feel incredibly blessed to share my life with her.

My Ph.D. research is funded by the Swedish Research Council (VR) Project: Non-Functional Requirements for Machine Learning: Facilitating Continuous Quality Awareness (iNFoRM).

List of Publications

Appended publications

This thesis is based on the following publications:

[A] **K. M. Habibullah**, G. Gay, J. Horkoff "Non-functional requirements for machine learning: understanding current use and challenges among practitioners"
Requirements Engineering (2023), 28(2), pp.283–316.

[B] **K. M. Habibullah**, H. -M. Heyn, G. Gay, J. Horkoff, E. Knauss, M. Borg, A. Knauss, H. Sivencrona, P. Li. Jing "Requirements and software engineering for automotive perception systems: an interview study"
Requirements Engineering (2024), 29(1), pp.25-48.

[C] H. -M. Heyn, **K. M. Habibullah**, E. Knauss, J. Horkoff, M. Borg, A. Knauss, P. Li. Jing "Automotive perception software development: Data, annotation, and ecosystem challenges"
IEEE/ACM 2nd International Conference on AI Engineering–Software Engineering for AI (CAIN), (2023), pp. 13-24.

[D] **K. M. Habibullah**, G. Gay, J. Horkoff "Non-Functional Requirements for Machine Learning: An Exploration of System Scope and Interest"
IEEE/ACM 1st International Workshop on Software Engineering for Responsible Artificial Intelligence (SE4RAI) (2022), pp.29-36.

[E] **K. M. Habibullah**, G. Gay, J. Horkoff "A Framework for Managing Quality Requirements for Machine Learning-Based Software Systems"
International Conference on the Quality of Information and Communications Technology (2024), (pp. 3-20).

[F] **K. M. Habibullah**, J. G. Diaz, G. Gay, J. Horkoff "Maintainability Definition, Scoping, and Measurement for Machine Learning Systems"
Accepted to the International Conference on the Quality of Information and Communications Technology (2025).

Other publications

The following publications were published during my PhD studies. However, they are not appended to this thesis, due to contents overlapping that of appended publications or contents not related to the thesis. For example, journal paper [A] is an extension of conference paper [a], journal paper [B] is an extension of conference paper [b], the content of conference paper [F] overlaps with the content of poster [c]. The contents of [d], [f], and [g] are not relevant to this thesis and [e] is my licentiate thesis.

[a] **K. M. Habibullah**, J. Horkoff "Non-functional Requirements for Machine Learning: Understanding Current Use and Challenges in Industry" *2021 IEEE 29th International Requirements Engineering Conference (RE), Notre Dame, IN, USA, (2021), pp. 13-23, doi: 10.1109/RE51729.2021.00009*

[b] **K. M. Habibullah**, H. -M. Heyn, G. Gay, J. Horkoff, E. Knauss, M. Borg, A. Knauss, H. Sivencrona, P. Li. Jing "Requirements engineering for automotive perception systems" *29th International Working Conference on Requirement Engineering: Foundation for Software Quality, Springer, (2023)*

[c] **K. M. Habibullah**, J. G. Diaz, G. Gay and J. Horkoff "Scoping of Non-Functional Requirements for Machine Learning Systems" *2024 IEEE 32nd International Requirements Engineering Conference (RE), Reykjavik, Iceland, (2024), pp. 496-497, doi: 10.1109/RE59067.2024.00061*

[d] N. M. Johansson, R. Siddiq and **K. M. Habibullah** "The Role of Personality Traits in Shaping Trust in LLM-Generated Code Explanations" *In submission to International Conference on the Quality of Information and Communications Technology (2025)*

[e] **K. M. Habibullah** "Understanding and Managing Non-functional Requirements for Machine Learning Systems" *Licentiate thesis - Chalmers Library, (2023)*

[f] Umm-E-Habiba and **K. M. Habibullah** Explainable AI: A Diverse Stakeholder Perspective" *2024 IEEE 32nd International Requirements Engineering Conference (RE), Reykjavik, Iceland, (2024), pp. 494-495, doi: 10.1109/RE59067.2024.00060*

[g] **K. M. Habibullah** "Exploring Challenges and Solutions for Non-Functional Requirements for Machine Learning Systems" *REFSQ Workshops, (2023)*

Research Contribution

For Paper A, I was responsible for the conceptualization, development of the research methodology, data collection and analysis, and drafting the original manuscript. I also contributed significantly to the validation of results, visualization of findings, and preparation of the final publication.

For Paper B, I contributed to conceptualization, methodology, data collection, and formal analysis. I was primarily responsible for writing the original draft, visualization, and editing the manuscript.

For Paper C, I contributed to conceptualization, methodology development, and formal analysis. I also participated in validation, manuscript review, and editing the manuscript.

For Paper D, I was responsible for conceptualization, data collection, data analysis, validation, data curation, data visualization, and writing the original draft of the manuscript.

For Paper E, I was responsible for conceptualization of the study, the development of the framework, data collection, formal analysis, and the drafting of the original manuscript. I was also responsible for validation and result visualization.

For Paper F, I was responsible for the conceptualization, methodology, formal analysis, and validation of the study. I drafted the original manuscript and created visualizations to support the presentation of findings.

In all the articles, my co-authors contributed to the conceptualization, methodology, investigation, and validation processes, and they also provided substantial input by reviewing and editing the manuscript as well as refining the draft.

To define my contribution to the appended papers in this thesis, I use the CRediT (Contribution Roles Taxonomy) model, defined by Brand et al. [1].

Table 1: Contributions of Khan Mohammad Habibullah to the appended papers of this thesis

Role / Paper	**A**	**B**	**C**	**D**	**E**	**F**
Conceptualization	✓	✓	✓	✓	✓	✓
Methodology	✓	✓	✓	✓	✓	✓
Software						
Validation	✓	✓	✓	✓	✓	✓
Formal analysis	✓	✓	✓	✓	✓	✓
Investigation	✓	✓	✓	✓	✓	✓
Resources	✓	✓	✓	✓	✓	✓
Data Curation	✓	✓	✓	✓	✓	✓
Writing - Original Draft	✓	✓		✓	✓	✓
Writing - Review & Editing	✓	✓	✓	✓	✓	✓
Visualization	✓	✓	✓	✓	✓	✓
Supervision						
Project administration						
Funding acquisition						

Contents

Abstract v

Acknowledgement vii

List of Publications ix

Personal Contribution xi

1 Introduction 1
- 1.1 Research Goal and Research Questions . . . 3
- 1.2 Background and Related Work . . . 5
  - 1.2.1 Machine Learning (ML) . . . 5
  - 1.2.2 Requirements Engineering (RE) . . . 6
    - 1.2.2.1 Non-functional Requirements (NFRs) . . . 7
  - 1.2.3 RE for ML Systems . . . 8
    - 1.2.3.1 NFRs for ML Systems . . . 9
  - 1.2.4 Software and Systems Methods for ML Systems . . . 11
- 1.3 Research Methodology . . . 11
  - 1.3.1 Problem Space Exploration . . . 13
  - 1.3.2 Artifact Design . . . 16
  - 1.3.3 Evaluation of the Proposed Artifacts . . . 17
- 1.4 Results . . . 18
- 1.5 Threats to Validity . . . 34
- 1.6 Summary of Contributions . . . 37
- 1.7 Future Work . . . 40
- 1.8 Conclusion . . . 40

2 Paper A 43
- 2.1 Introduction . . . 44
- 2.2 Related Work . . . 46
- 2.3 Methodology . . . 50
  - 2.3.1 Interviews . . . 51
  - 2.3.2 Survey . . . 56
- 2.4 Results . . . 61
  - 2.4.1 NFR Importance, Scope, and Challenges . . . 61
    - 2.4.1.1 Perceived NFR Importance (RQ1) . . . 61
    - 2.4.1.2 Scope of NFRs (RQ2) . . . 65
    - 2.4.1.3 NFR and ML-related Challenges (RQ3) . . . 66

2.4.2 NFR Measurement Scope, Capture, and Challenges . . 70
2.4.2.1 NFR Measurements (RQ4) 70
2.4.2.2 NFR Measurement Scope (RQ5) 71
2.4.2.3 NFR Measurement Capture (RQ6) 73
2.4.2.4 NFR Measurement Challenges (RQ7) 73
2.4.3 Differences Between Industry and Academia (RQ8) . . . 75
2.4.3.1 Differences in Perceived NFR Importance (RQ1) 75
2.4.3.2 Differences in Scope of NFRs (RQ2) 78
2.4.3.3 Differences in NFR Challenges (RQ3) 79
2.4.3.4 Differences in NFR Measurements (RQ4, RQ5, RQ6) 81
2.4.3.5 Differences in NFR Measurement Challenges (RQ7) 82
2.5 Discussion and Future Work 83
2.5.1 Research Gaps 87
2.5.2 Threats to Validity 88
2.6 Conclusions 90

3 Paper B **91**
3.1 Introduction 92
3.2 Related Work 94
3.2.1 RE for ML 94
3.2.2 RE for Automotive and Driving Automation Systems . 95
3.2.3 Quality Assurance for ML 96
3.2.4 Software and Systems Methods for Machine Learning . 97
3.3 Methodology 98
3.3.1 Data Collection 98
3.3.2 Data Analysis 99
3.4 Results: Requirements Engineering (RQ1) 100
3.4.1 Operational Design Domain (ODD) 100
3.4.2 Scenarios and Edge Cases 102
3.4.3 Requirements Breakdown 103
3.4.4 Requirements Traceability 105
3.4.5 Requirements Specification 106
3.5 Results: Quality (RQ2) 107
3.5.1 System-level Quality 107
3.5.2 Safety 109
3.5.3 KPIs and Metrics 112
3.6 Results: Systems and Software Engineering (RQ3) 114
3.6.1 SE Methodology 114
3.6.2 Data Quality Methods 116
3.6.3 Verification and Validation Methods 116
3.7 Summary and Discussion 118
3.7.1 Requirements Engineering Topics and Challenges (RQ1) 119
3.7.2 Quality Topics and Challenges (RQ2) 120
3.7.3 Systems and Software Engineering Topics and Challenges (RQ3) 121
3.7.4 Future Directions in Research and Practice 121
3.7.5 Threats to Validity 123

3.8 Conclusion . . . 124

4 Paper C **125**
4.1 Introduction . . . 126
4.2 Related Work . . . 127
4.3 Method . . . 129
4.3.1 Preparation of interviews . . . 129
4.3.2 Data collection . . . 130
4.3.3 Data analysis . . . 131
4.3.4 Result validation . . . 132
4.4 Results . . . 132
4.4.1 RQ1: The ability to specify data for the development of automotive perception software . . . 132
4.4.1.1 Data collection . . . 133
4.4.1.2 Processes and Way of working . . . 134
4.4.1.3 Data quality . . . 135
4.4.2 RQ2: The ability to specify annotations for data used in automotive perception software . . . 136
4.4.2.1 Annotation costs . . . 136
4.4.2.2 Annotation quality . . . 137
4.4.2.3 Guidelines & Specification . . . 138
4.4.3 RQ3: Automotive industry's ecosystems and business models for data-intensive software developments . . . 139
4.4.3.1 Business environment . . . 139
4.4.3.2 Contracts & Infrastructure . . . 141
4.4.3.3 Shared responsibility . . . 141
4.5 Discussion . . . 142
4.5.1 Recommendations . . . 142
4.6 Threats to validity . . . 143
4.6.1 Threats to internal validity . . . 143
4.6.2 Threats to external validity . . . 144
4.7 Conclusion and Outlook . . . 144

5 Paper D **147**
5.1 Introduction . . . 148
5.2 Background and Related Work . . . 149
5.3 Methodology . . . 152
5.3.1 NFR Clustering . . . 152
5.3.2 Publication Volume Estimation . . . 153
5.3.2.1 Initial Paper Search . . . 153
5.3.2.2 NFR Selection . . . 154
5.3.2.3 Estimating the Number of Relevant Papers for Selected NFRs . . . 154
5.3.3 NFR Scope Determination . . . 155
5.4 Results and Discussion . . . 156
5.4.1 Threats to Validity . . . 161
5.5 Conclusions . . . 162

6 Paper E **163**
6.1 Introduction . . . 164
6.2 Running Example . . . 165
6.3 NFR Management Framework . . . 166
6.3.1 Step 1: Select and Prioritize NFR Types . . . 166
6.3.2 Step 2: Define NFR Types and Identify NFR Type Scope 168
6.3.3 Step 3: Balance NFR Type Trade-offs . . . 169
6.3.4 Step 4: Specify NFR Measurement Catalogue . . . 170
6.3.5 Step 5: Specify NFRs Using Template . . . 170
6.4 Preliminary Evaluation . . . 171
6.4.1 Methodology . . . 171
6.4.2 Impressions and Insights . . . 172
6.4.3 Potential Future Improvements . . . 173
6.4.4 Threats to Validity . . . 174
6.5 Related Work . . . 174
6.6 Conclusion . . . 176

7 Paper F **177**
7.1 Introduction . . . 178
7.2 Background and Related Work . . . 179
7.2.1 Modularity Measurement . . . 179
7.2.2 Maintainability Challenges for ML Systems . . . 180
7.3 Maintainability for ML Systems . . . 181
7.3.1 ML System Scoping . . . 181
7.3.2 Maintainability Definition for ML Systems . . . 182
7.3.3 Modified Modularity Metrics for Code . . . 183
7.4 Evaluation . . . 184
7.5 Results and Discussion . . . 186
7.5.1 Traditional and Revised Metrics, Whole System (RQ1) . 186
7.5.2 Factors that Contribute to Differences (RQ2) . . . 188
7.5.3 Validity of Conceptual Breakdown (RQ3) . . . 189
7.5.4 Differences Between Scopes (RQ4) . . . 189
7.6 Limitations and Threats to Validity . . . 191
7.7 Related Work . . . 192
7.8 Conclusion . . . 193

Bibliography **195**

Chapter 1

Introduction

Machine learning systems are software or systems that integrate or use machine learning (ML) to perform different tasks. ML has increasingly become a central component in a wide range of software systems. The integration of ML into software systems has revolutionized the way modern software is conceived, developed, and deployed. ML systems use algorithms that learn from large amounts of data, enabling the system to perform tasks that would be difficult to do manually or using traditional software [2]. ML has seen unprecedented growth in recent years, and ML is increasingly and extensively being used in many domains, including healthcare, finance, e-commerce, and most notably, complex and safety-critical domains such as autonomous vehicles and medical diagnostics to perform decision-making and prediction tasks, including object detection, image processing, and natural language processing. Despite the transformative potential of ML, the integration of such systems into the safety-critical and high-assurance domains raises significant engineering challenges, particularly in terms of quality assurance. There is growing concern about potential biases [3] and unintended consequences [4, 5] that may result from ML algorithms' influence on critical decisions and prediction operations. Additionally, the non-deterministic behavior of ML makes the development of ML systems more complex, expensive, and effort-intensive than traditional systems. As a consequence, ML systems require the fulfillment of certain quality requirements or deal with constraints such as fairness [3], transparency [4], privacy [6], security [7], and safety [5]. From a requirement engineering (RE) perspective, these quality aspects are known as non-functional requirements (NFRs) [8, 9].

For traditional software, NFRs such as performance, reliability, maintainability, and usability have relatively well-understood and established definitions, metrics, and methods for specification and validation. However, for ML solutions, many of these NFRs have different meanings, ambiguity, and are not yet well understood [10]. For example, the meaning of maintainability and adaptability is unclear in the ML context. Maintainability in an ML context may encompass retraining, data drift detection, or model lifecycle management—concerns that differ significantly from traditional software. Additionally, emergent NFRs such as fairness, explainability, and transparency have become critical in the context of ML due to ML's reliance on data and statistical

inference, while some NFRs such as compatibility and modularity may have reduced importance [3, 11]. Moreover, new NFRs, such as retrainability, may become relevant for ML systems. In addition, we observe common quality trade-offs among NFRs (e.g., security vs. performance) in traditional systems, but there are a few works that explored quality trade-offs in an ML context [3].

Furthermore, NFR-related challenges may become more critical when ML systems are deployed in safety-critical domains, such as automotive software, where perception systems rely heavily on ML to interpret sensor data and make real-time decisions in safety-critical environments and different edge scenarios. Failure to meet quality expectations can have severe consequences in such safety-critical domains. Recent research has shown that current RE approaches are insufficient to support the development of ML-based perception systems, and future research is needed to understand best practices and propose suitable approaches [12]. There is a lack of clarity regarding how to define, specify, trace, and validate NFRs throughout the ML pipeline, including the data, model, and system levels [12, 13]. Measuring NFRs for ML systems has also remained underexplored. For example, while accuracy is a widely discussed metric, it is unclear how to measure other NFRs comprehensively, such as robustness, fairness, or explainability, in ways that are actionable in practice [14]. Therefore, understanding and managing NFRs for ML can be challenging and requires a rigorous RE approach. Hence, researchers and practitioners working with ML and RE must recognize the importance of RE as a foundational element of quality assurance for ML and incorporate it to ensure the success of ML systems [15]. RE practices can help ensure that ML systems are designed, developed, and deployed with attention to their quality attributes, thereby improving overall performance, usability, and trust while mitigating the risk of failure.

ML is a part of a larger system [16], and ML can be decomposed into several granular levels, e.g., training data, ML model, and results. Therefore, different NFRs may apply to different aspects of the system. For example, some NFRs may be relevant to the algorithm used for learning. In contrast, others may apply to the training data or the model trained using that data, and some NFRs may apply to the results of applying the model or to the broader ML system that utilizes those results. Therefore, determining the scope of NFRs for ML systems, including identification, definition, and specification, remains challenging. Furthermore, measuring NFRs in an ML space and different granular levels of the system has not been explored, e.g., how to measure the accuracy of the ML algorithm or system as a whole.

Therefore, it is necessary to identify important NFRs for ML systems, NFR and NFR measurement-related challenges, and RE-related challenges in different example contexts. For a better understanding of the NFRs and NFR scopes, it is important to define specific NFRs for ML with generic definitions, identify NFRs for ML that received less attention in the literature, identify the initial scope of defining and measuring NFRs in ML systems, and cluster them based on shared characteristics. ML systems involve heterogeneous components, evolving models, and complex dependencies, that make ad-hoc handling of NFRs inconsistent and error prone. Therefore, we need to develop frameworks and/or solutions to manage NFRs as part of the ML systems development process and continue to evaluate, refine, and improve the frameworks and

solutions that guide practitioners in systematically specifying, prioritizing, measuring, and monitoring NFRs throughout the ML system development process. Although large language models (LLMs) and generative AI (GenAI) have reshaped many AI applications, traditional ML-based systems for analytics and decision-making still remain essential—especially in safety-critical and cyber-physical domains like autonomous driving and medical devices. As such, research on RE for ML systems (RE4ML) continues to be important, and much of effort to define, measure, and manage NFRs for ML will potentially apply and remain highly relevant in the GenAI era.

This thesis is organized as follows: Section 1.1 describes the research goal and formulated research questions to address those research goal. Section 1.2 discusses the background and studies related to this thesis. The research methodologies used to answer the research questions in order to fulfill research goal is discussed in Section 1.3. Summary of results of the Ph.D. research thus far is presented in Section 1.4. Threats to the validity of the studies conducted as a part of the thesis is described in Section 1.5. The summary of contributions of Ph.D. research is presented in Section 1.6. Further research plan and future work is described in Section 1.7. Section 1.8 concludes the thesis with a summary of the works. The appended publications are presented in Chapter 2, Chapter 3, Chapter 4, Chapter 5, Chapter 6, and Chapter 7.

1.1 Research Goal and Research Questions

The PhD study focuses on identifying the challenges related to NFRs for ML, developing and demonstrating artifacts as solutions, and evaluating those artifacts in practice. The overall research goal of this thesis is to **understand challenges and practices in NFRs for ML and create solutions to manage NFRs for ML systems.**

To guide our study on understanding and managing NFRs for ML systems, we formulate a number of research questions (RQs), as follows:

RQ1 What are some of the key RE topics and challenges for ML systems in industry?

Non-functional requirements (NFRs) are a type of requirement for systems and software that are identified and managed by the requirements engineering (RE) process. By answering this research question, the aim is to understand the current practices and challenges perceived by the practitioners working with ML and RE in the industry. To explore these aspects in depth, we focused on autonomous perception systems (APS) in the automotive domain as a representative example of ML systems. APS are a critical part of driving automation systems (DAS) and heavily rely on ML models for tasks such as perception, detection, and recognition – as part of large and complex software systems – making APS a suitable exemplar for studying RE topics and challenges for ML systems. For this RQ, we conducted a group interview study with practitioners in the autonomous vehicle industry who work with driving autonomous systems (DAS). The description of the study is elaborated in **Paper B** and **Paper C**. The results are also complemented by our published paper [17] [1].

[1]Which is not included as an appended paper because of overlapping content.

RQ2 How are NFRs for ML systems currently measured, and what are the current perceived NFR and NFR measurement-related challenges for ML systems in practice?
NFRs measurements are required to track and manage the quality of an ML system. We also aim to understand NFR-related and NFR measurement-related challenges experienced by the practitioners working with ML. We conducted an interview study and then a broader survey to answer this research question. **Paper A** contains the description of the study. The results are also complemented by our published paper [18][1].

RQ3 Which NFRs are more or less important for ML systems than they are for traditional systems?
The NFRs that are important for traditional systems may not be important for ML systems or may not have the same level of importance. Hence, it is crucial to identify and understand important NFRs for ML systems. An interview and a broader survey were conducted to answer this research question, and the studies are described in **Paper A**. The results are also complemented by our published paper [18][1].

RQ4 Which NFRs for ML systems have received the most—or least—attention in existing research literature?
After identifying important NFRs for ML, we are interested to understand, among important NFRs for ML, which NFRs received more attention and which ones received less attention in research. We performed a part of a systematic mapping study to answer this research question, which is described in **Paper D**.

RQ5 Over what aspects of an ML system are NFRs defined and measured?
ML systems can be decomposed into several smaller parts, and furthermore, ML is part of a larger system. Therefore, it is important to identify over which part of the system NFRs should be defined and measured. In the interview and survey study described in **Paper A**, we tried to understand the scope of defining and measuring NFRs for ML systems. In **Paper D**, we performed initial scoping of certain NFRs for ML systems, and in **Paper F**, we provided a breakdown of ML system elements and scoping of NFRs for ML systems.

RQ6 What structured approaches or framework(s) can support the identification, scoping, specification, and management of NFRs for ML systems?
The solutions to manage NFRs for ML are not well developed and organized, and their consideration is in the initial stage. Therefore, it is important to develop solutions to manage NFRs for ML in a structured way. We have begun to address this question in **Paper D** with an early conceptualization of NFRs for ML scoping and clustering and a more comprehensive and systematic treatment of NFR scoping is done and presented in **Paper F**. Furthermore, we extended these results and developed a quality framework to identify, specify, measure, and manage NFRs for ML systems, described in **Paper E**. We have performed a preliminary evaluation based on expert interviews.

RQ7 How to develop ML-specific measurements of NFRs for ML systems?

We conducted a study to define, scope, and measure maintainability, as an example NFR that is important for ML systems. ISO/IEC 25059 specifies eight high-level quality requirements for ML systems, including maintainability [19]. Maintainability comprises five sub-characteristics, including modularity, reusability, modifiability, analyzability, and testability [19]. In particular, in this study, we focus on modularity. We evaluated our proposed solution over ten open-source ML systems. The study is described in **Paper F**. The measurement approaches proposed in Paper F can be integrated into the solutions addressing RQ6.

The overview of this thesis is presented in Fig. 1.1, that maps the activities, research questions, and methods used in this thesis, and activities and research methods that will be used in future work to achieve the overall research goal.

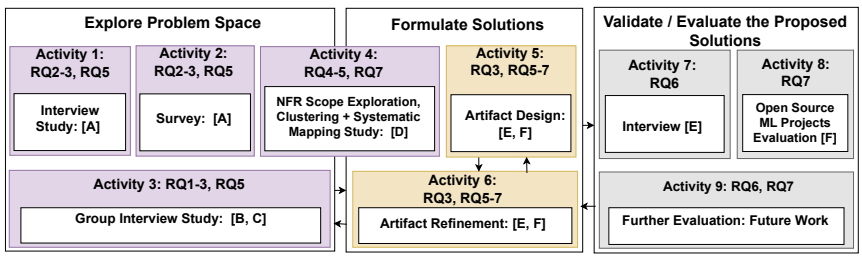

Figure 1.1: Overview of the thesis. The activities with purple backgrounds represent problem exploration activities, the yellow backgrounds represent work related to solutions, and the gray backgrounds represent solution evaluation and future work.

1.2 Background and Related Work

This section provides terminology and background information on the basic concepts, such as artificial intelligence (AI) and ML, RE, and NFRs used in this thesis. This section also provides an overview of the related work that pertains to the background information of this thesis.

1.2.1 Machine Learning (ML)

Machine learning (ML) is a sub-field of artificial intelligence (AI) that involves the study of algorithms and statistical models that allow software and computer systems to learn and make predictions or judgments based on data. By recognizing patterns in the data they are trained on, machine learning (ML) algorithms are developed to automatically improve over time [20]. ML has emerged as a paradigm-shifting technology in recent years that promotes innovation and growth in different industries, including healthcare, finance, transportation, and entertainment [21]. Machine learning has been used to develop applications for personalized recommendations, fraud detection, predictive maintenance, and image and speech recognition.

Although ML has many proven and potential benefits, there are also significant challenges associated with developing and deploying ML systems in different aspects and domains [22]. ML systems development challenges include a lack of identifying and understanding requirements, defined development processes, and data-related challenges [23]. Furthermore, one key challenge is to ensure that ML systems meet NFRs, such as performance, adaptability, safety, reliability, and security [24]. Though ML has the potential to transform many aspects of modern life, it is important to ensure that these systems are developed following standard guidelines, structures, and processes and meet certain quality aspects, such as performance, reliability, and security, so that they can be trusted and relied upon in practice.

Furthermore, the deployment of ML in real-world and safety-critical systems (e.g., autonomous vehicles, health care) has further amplified the need for rigorous quality assurance and engineering practices that account for both the variability and complexity of ML systems [25]. In such domains where human lives are at stake, ML systems must operate reliably, transparently, and safely in open, uncertain environments and different edge cases. Therefore, researchers and practitioners must rethink how software quality and assurance are ensured in the ML context [26].

Finally, the ML system development process no longer ends with code completion but extends to data collection, feature engineering, model selection, evaluation, deployment, and monitoring, which is more complex. As a result, traditional software engineering (SE) and requirements engineering (RE) methods require substantial adaptation to remain effective in ML contexts.

1.2.2 Requirements Engineering (RE)

Requirements are the specific services, capabilities, and qualities that a software system must have to meet the demands of stakeholders [27]. Requirements serve as a foundation for the design, development, testing, and maintenance activities of the software system and play a crucial role in achieving software quality [28]. A clear set of requirements ensures that the software meets business goals, provides the desired value, and meets the stakeholders' needs by following regulations. However, RE is one of the most critical and complex phases in software engineering and involves processes such as domain analysis, elicitation, specification, assessment, negotiation, documentation, and evolution [29]. High-quality RE directly contributes to the success of software projects by minimizing misunderstandings, reducing development costs, and improving the alignment between stakeholder expectations and the delivered system. In contrast, poor RE contributes to the software system's failure because of vague requirements, scope deviation, or misaligned goals [30].

There are two main types of requirements in software engineering:

- **Functional Requirements:** Functional requirements (FRs) are specifications of the specific tasks, functions, or operations that the software system must perform [31]. FRs specify the behavior of the system—what the system should do in response to specific inputs or events. For example, one of the FRs for a word processing system might be, "The system must allow users to produce, modify, and format text documents."

- **Non-Functional Requirements:** A non-functional requirement is an attribute of or a constraint on a system, where attributes are performance or quality requirements [9]. Accuracy, dependability, usability, and security are some examples of NFRs. In this thesis, we adopt a narrower notion of NFRs, focusing on quality requirements and excluding constraints, with a focus on managing quality attributes of ML systems. This perspective aligns with the interpretation of NFRs by Mylopoulos et al. and Rebertson & Robertson [32, 33].

Understanding, defining, and comprehending different types of requirements is very crucial and important as a part of software engineering because it guides and enables software developers to develop high-quality systems that satisfy the stakeholders' needs.

RE is the process of gathering, analyzing, documenting, validating, and maintaining a system's requirements [34]. In traditional or contractual RE processes, software engineers identify the needs of stakeholders and translate those needs into precise and understandable requirements to develop and test a system. However, in exploratory or agile development contexts, RE tends to be more iterative and lightweight, with requirements evolving incrementally rather than being fully specified upfront. The following steps are usually involved in a traditional RE process, and are also often applied to some extent in agile RE [35, 36]:

- Elicitation: In this step, the requirements engineers gather information from stakeholders about the system. Requirements elicitation can be done through interviews, surveys, seminars, workshops, or other techniques.
- Analysis: Requirements engineers examine the requirements and look for inconsistencies, ambiguities, and conflicts in the requirements.
- Specification: In the requirements specification step, requirements engineers list the FRs and NFRs in a specification document (e.g., SRS document). This document can act as a formal contract between the development team and the stakeholders.
- Validation: In this step, the development team and stakeholders evaluate the specified requirements to ensure that the requirements appropriately reflect the demands of stakeholders.
- Management: This step includes monitoring and managing changes in requirements (if any) to ensure that changes do not affect the overall project schedule or budget.

RE is an iterative process, and the requirements are monitored and managed throughout the development process.

1.2.2.1 Non-functional Requirements (NFRs)

NFRs are specifications that define a software system's qualities, attributes, or constraints [8, 9]. A software system's utility is usually determined by both its functionality and its non-functional characteristics, such as performance, usability, flexibility, accuracy, and security [37]. NFRs are considered essential

for the success of software and have been widely researched, but there is still a lack of standard guidelines for eliciting, defining, documenting, and validating NFRs [9]. There is also debate among the RE community about when NFRs should be considered in the RE process [8]. Montgomery et al. conducted a systematic mapping study and found that empirical research on requirements quality mainly focuses on improvement techniques, emphasizing attributes like ambiguity, completeness, consistency, and correctness, with a little research on evidence-based definitions and evaluations of quality attributes. The authors highlighted the need for more diverse, empirically grounded research [38]. Doerr et al. applied a systematic and experience-based method for eliciting, documenting, and analyzing NFRs, with the aim of creating a comprehensive set of traceable and measurable NFRs [39]. Ameller et al. conducted an interview-based study and revealed that model-driven development adaptation is a complex process with little or no support for NFRs [40]. Adams et al. claimed that identifying NFRs early in the design process is crucial to avoid increased cost later and then introduced a taxonomy of 27 NFRs that are organized into four categories to provide a framework for addressing them during early system design [41]. Sachdeva et al. conducted a case study that proposed a new solution to address performance and security NFRs in big data and cloud projects using Scrum. Their results illustrate that the proposed approach effectively balances performance and security needs, even when conflicts exist between them, within an agile methodology [42]. However, while most research on NFRs has focused on traditional software systems, there is a growing focus on NFRs for ML systems. However, while most research on NFRs has focused on traditional software systems, there is a growing focus on NFRs for ML systems.

1.2.3 RE for ML Systems

RE provides a systematic approach to identify and manage requirements for ML systems. By incorporating RE principles into the ML systems development process, practitioners can ensure that the ML system's design, development, and deployment meet the necessary stakeholders' requirements and quality aspects. Following RE principles can improve the overall performance and usability of ML systems and minimize the risk of failure.

The development and implementation of ML systems include diverse stakeholders' involvement, and RE can facilitate simplified communication and collaboration between them [43]. Easy and efficient collaboration and communication are crucial in ML systems development, as ML systems often involve complex interactions between multiple components and stakeholders, including data scientists, software developers, and end-users [44]. By using RE techniques, stakeholders can collaborate to ensure that the system meets the necessary quality requirements and satisfies the needs of all involved parties.

There have been many approaches and research on using ML to improve RE processes (e.g., model extraction [45, 46], prioritization [47], and categorization [48]), there has been a growing focus on RE research for ML systems [15]. However, researchers have recently begun identifying and highlighting challenges and solutions in RE for AI-based systems.

Yoshioka et al. identified RE-related research challenges for ML systems

and recommended the need for monitoring requirements for concept drift [49]. Ahmad et al. performed a systematic literature review and a mapping study and investigated current approaches in writing requirements for AI/ML systems [50, 51]. They analyzed the key tools and techniques used to specify and model requirements for AI/ML systems and highlighted that current RE applications are not sufficient to manage most AI/ML systems. They emphasized the need to provide new techniques and tools to support RE4AI and suggested further research in AI ethics, trust, and explainability. Vogelsang & Borg noted that the development process for ML systems is more complex, with the need to effectively use large quantities of data, and dependence on other quality requirements (NFRs) [15]. Belani et al. highlighted, discussed, and addressed issues for RE disciplines in constructing ML- and AI-based complex systems. They stated that one of the difficulties in developing ML-enabled software is identifying NFRs throughout the software lifecycle, not just in the first phases of dealing with requirements. ML-based systems require interventions to SE processes on different aspects, such as versioning of the ML models, dataset availability, and the whole system's performance [52]. Villamizar et al. conducted a systematic mapping study and proposed a catalogue of 45 concerns to be considered when specifying ML systems, covering five different perspectives they identified as relevant for such ML systems: objectives, user experience, infrastructure, model, and data [53]. Pei et al. performed a literature review and a step-by-step collaborative requirements analysis process to provide an overview of the collaboration among the different roles in RE for ML systems. Then they summarized the typical patterns for collaborations, and proposed high-level guidelines for evaluation and selection of viable patterns [54].

Heyn et al. reported challenges in defining data quality attributes, testing, monitoring, reporting, and human factors in AI context [55]. Nagadivya et al. explored ethical guidelines for the development of transparent and explainable AI systems, defined by various organizations. They found that transparency and explainability are related to several quality requirements, such as fairness, trustworthiness, understandability, traceability, auditability, and privacy [56]. They suggest a structured way for practitioners to define explainability as an NFR for AI systems. Further research focuses on specific types of requirements for AI, such as transparency (e.g., [57]) or legal requirements (e.g., [58]).

Along with the research discussed above, we focus on a wider view of NFRs for ML in research and in industry, collecting an overview of NFR perception from practitioners, and aim to address the challenges related to NFRs for ML systems.

1.2.3.1 NFRs for ML Systems

As the adoption of ML in software systems grows, addressing NFRs has become one of the key areas of research. This section highlights key studies that explore the challenges, directions, and emerging practices related to NFRs in ML systems.

Horkoff discussed the challenges of NFRs for ML and research direction, including how RE can be adjusted for solutions to address the challenges related to NFRs for ML systems [13]. Kuwajima et al. illustrated that ML models lack processes and methods in terms of requirements specification, design

specification, interpretability, and robustness [59]. The authors also compared the conventional system quality standard SQuaRE with the characteristics of ML models to identify quality models for ML systems, and the results revealed that the absence of requirements specification and robustness has the greatest impact on quality models. Similarly, Gruber et al. stated that less research has been done in the ML context on modeling NFRs, and research tends to focus on functional requirements more [60].

Vogelsang & Borg stated that RE practitioners need to understand ML performance measures to state good functional requirements for ML systems [15]. They also emphasized that RE for ML should focus on requirements over both data and the whole system. Khan et al. discussed the importance of documenting NFRs for ML systems, reviewed the relationship between RE and software architecture with respect to ML, and analyzed three methods (SysML extensions for functional and non-functional requirements, GORE-MLOps methodology, and methodology for specification, analysis, and verification in autonomous systems (SAV) for documenting and handling NFRs for delivering quality software systems [61]. Recently, NFRs are getting more attention in research, and researchers are focusing more on specific NFRs, such as bias and fairness in machine learning systems [62], transparency [63], uncertainty [64], explainability [65], and safety [66].

Villamizar et al. identified quality characteristics relevant to ML systems and NFR-related challenges, such as incomplete and fragmented understanding of NFRs for ML and lack of validated RE techniques to manage RE [12]. Martino et al. classified 30 NFRs for ML systems and identified 23 SE challenges in addressing them [67]. Bajraktari et al. proposed a template to document NFRs for ML systems, addressing the challenges in RE for such systems [68]. Martinez et al. performed a systematic mapping study and found that safety and dependability are the most studied properties of AI-based systems [69]. Previous studies have discussed the challenges and opportunities of addressing NFRs in ML system development. However, while there is some research on NFRs, there is limited research specifically focused on solutions to NFR challenges and on understanding the current practices and processes that professionals use to define, allocate, and measure these NFRs. Gezici et al. conducted a systematic literature review and provided a road map for researchers for deeper understanding of quality challenges, attributes, and practices in the context of software quality for AI-based software [70]. Ali et al. conducted a systematic mapping study to understand, classify, and critically evaluate existing quality models for AI systems, software, and components. The authors found quality characteristics (e.g., privacy, accuracy, fairness) for AI systems and software, but they did not find any quality characteristics and models for AI software components [71].

Prior research focused on understanding NFRs for ML systems, highlighting challenges, proposing taxonomies, and focusing on specific quality attributes. However, most existing work remains either conceptual, limited to particular NFRs (e.g., fairness, safety, transparency), or focused on abstract models and high-level frameworks. In contrast, this thesis focuses on a comprehensive and empirical approach to understand current industrial practices and challenges in RE for ML, identify critical NFRs and measurement-related challenges, and propose actionable, scope-aware solutions to manage NFRs for ML systems.

Unlike previous work, this thesis proposes a step-by-step framework for identifying, prioritizing, scoping, measuring, and specifying NFRs across different ML system components. Furthermore, it introduces adapted maintainability measurement metrics tailored to ML system components. By integrating detailed empirical evidence with practical solutions, this thesis will support both researchers and practitioners to understand NFR-related challenges and manage NFRs for ML systems.

1.2.4 Software and Systems Methods for ML Systems

Current systems and software development methods often do not account well for ML systems [72]. Wan et al. stated that adding ML to software systems significantly changes software development practices across requirements, design, testing, and process [73]. A recent study highlighted the challenges in developing ML systems using traditional SE practices, including the difficulties in integrating ML models into software systems, by emphasizing the need for improved practices [74]. Giray reported that the non-deterministic nature of ML systems complicates the SE aspects of engineering them. As a part of a systematic literature review, Giray also reported that there is a lack of mature tools and techniques to support the development and verification of ML systems [75].

Recently, researchers have started exploring the solutions and introducing some methods for the development of ML and AI systems. Indykov proposed a component-based approach to integrate ML functionality into complex systems by addressing challenges in system architecture and development workflows [76]. Lavin et al. proposed an ML technology readiness level framework that ensures robust, reliable, and responsible ML system development [77]. Hesenius et al. provided a structured engineering process framework named EDDA (engineering data-driven applications) that bridges existing gaps, supports data-driven application development, and ensures the required quality levels for critical components of ML systems [78]. Amershi et al. conducted a case study and described how various Microsoft software teams developed software applications with customer-focused AI features—integrating existing Agile software engineering processes with AI-specific workflows [79].

Although the aforementioned studies focus on adapting general software engineering processes, architectures, or readiness frameworks for ML systems, our work takes a more specific and empirically grounded approach, focusing on the challenges and solutions of NFRs for ML systems. In addition, our work includes in-depth insights into requirements and software engineering for autonomous perception systems and the complexities of managing data and annotation quality—areas largely overlooked in the literature.

1.3 Research Methodology

This Ph.D. thesis investigates and develops solutions for managing NFRs for ML systems by understanding existing challenges related to NFRs and ML systems. This thesis follows Design Science Research (DSR) as the primary methodology to answer the research questions outlined in Section 1.1. DSR is a systematic approach to designing, developing, and evaluating artifacts that are

intended to solve real-world problems [80]. DSR emphasizes the relationship between theoretical rigor and practical relevance, ensuring that the developed artifacts are knowledge-based and applicable to real scenarios [81]. We chose DSR as the primary research method as it provides us with a structured way to explore NFR for ML-related challenges and allowed us to iteratively design, evaluate, and refine our proposed solutions. The steps of the DSR method followed in this research are inspired by multiple well-established guidelines and adaptations of the methodology proposed in the literature [80, 82–87]. An overview of the adapted DSR cycles is presented in Fig. 1.2.

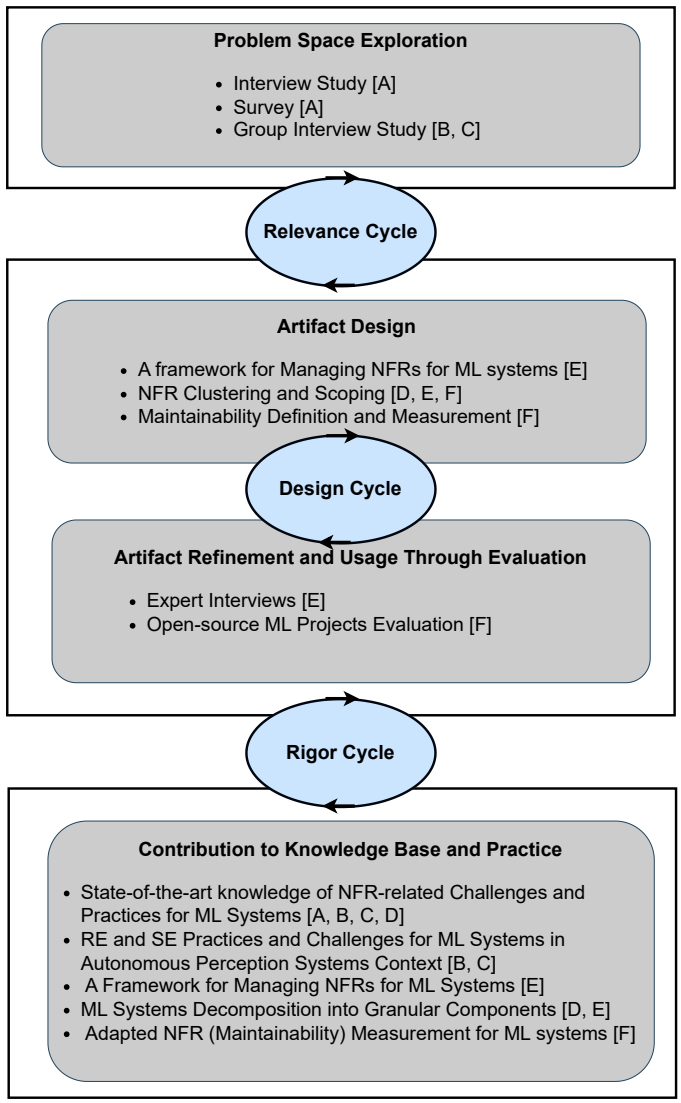

Figure 1.2: An overview of the adapted DSR cycles employed in this thesis.

Furthermore, Table 1.1 provides an overview of the research methodologies used across the included articles, along with the data collection methods and references to the papers that contributed to this thesis.

Table 1.1: Included articles with their research methods

Paper	Research Method	Data Source
A	Qualitative exploratory study	10 interviewees & 30 survey respondents
B	Qualitative exploratory study	7 group interviews, 19 interviewees
C	Qualitative exploratory study	7 group interviews, 19 interviewees
D	Part of systematic mapping study	Extracted information from literature
E	Artifact design and qualitative study	4 interviewees
F	Artifact design and Open-source ML projects evaluation	10 open-source ML projects

1.3.1 Problem Space Exploration

Interview Study (RQ2-3, RQ5): We conducted an interview study with 10 participants working with ML and RE to explore NFRs for ML-related challenges and how NFRs are perceived, specified, measured, and managed in practice. A detailed description of the research methodology, including instrument design, sampling, and analysis techniques can be found in **Paper A**. The study was driven by the primary research question: What is the perception and current treatment of NFRs in ML in industry? To guide the study and ensure the collection of rich and detailed data, we refined this question into more detailed sub-questions, such as, Which ML-related NFRs are more or less important in an ML context, and over what aspects of the system are those NFRs defined and measured in industry? How are NFRs for ML currently measured, and how are NFRs and their measurements captured in practice? Finally, what NFR- and NFR-measurement-related challenges are perceived?

In the interview study, the sample selection was a combination of convenience, purposive, and snowball sampling. This strategy allowed us to ensure that participants possessed practical experience with ML system development and RE while also ensuring diversity across roles, organizations, and domains. Data was collected through semi-structured interviews, allowing the flexibility to adapt follow-up questions based on participants' responses. Each interview followed a set of predetermined open-ended questions and follow-up questions to gather detailed information. We interviewed a total of 10 engineers and researchers who have varying levels of experience working with ML in different sectors of the ML industry, including automotive, healthcare, and information technology. Based on the interviewees' demographic information, we believe that the selected interviewees are representative of the practitioners who work in the data science and ML field, including their knowledge of NFRs.

With the interviewees' consent, we recorded each interview session, and for analysis, all interviews were transcribed and anonymized. The collected data was qualitative in nature, and we used thematic analysis and coding for

data analysis that is inspired by [88, 89]. The coding process involved starting with high-level codes aligned with our research questions, and refining and modifying them as we analyzed the transcripts.

Survey (RQ2-3, RQ5): To validate and expand the findings of the interview study and gain broader empirical insights, we conducted a survey, targeting professionals and researchers working with ML and RE. **Paper A** discusses the survey in detail. The primary objective of the survey was aligned with the interview study, focusing on the perception, importance, scope, and challenges related to NFRs for ML system development. Additionally, the survey was designed to explore potential differences in perspectives between participants from different professional backgrounds—specifically comparing those working in industry, academia, or across both domains. The survey allowed us to identify whether the participant's background influences how practitioners prioritize important NFRs and NFR-related challenges for ML systems.

We used a combination of purposive and convenience sampling to select survey participants, including practitioners from both academia and industry with experience in ML and requirements engineering. We used email to distribute the online survey to our contacts. To increase reach and diversity, we also distributed the survey through social media platforms such as Facebook, Twitter, and LinkedIn. We posted in groups and communities focused on AI, machine learning, and software engineering. The survey remained open from September 22, 2021, to April 7, 2022, which provided ample time for participation across multiple geographic regions and professional contexts. In total, 42 individuals responded to at least part of the survey, with 30 responses analyzed based on the demographic information provided and completion of the questions. The survey was designed with semi-structured questions to allow participants to express their opinions freely while collecting in-depth information.

The survey was divided into three main sections. The first section gathered demographic information that includes participants' current role, years of experience, primary domain (industry or academia), and familiarity with ML and NFRs. Demographic data provided a foundation for interpreting the results and identifying patterns across subgroups. In the second set of questions, we collected participants' general impressions of NFRs, whether the participants think NFRs play an important role in ensuring the quality of ML systems, the degree of importance of each NFR, and the scope of which part of the ML systems NFRs should be defined and measured. We provided a list of important NFRs (25 NFRs) identified as important in the interview study and their general definition of each NFR to help respondents answer the questions. This helped ensure that all respondents interpreted the terms similarly and could reflect more precisely on their relevance and scope. In the third set of questions, we collected information on NFR challenges, including whether respondents agreed that the identified challenges could affect the development of ML systems. Not all questions were mandatory, which allowed participants to focus on areas most relevant to their experience. The respondents were also given the space to write qualitative comments for most questions. We conducted a test survey with one Ph.D. student, one postdoctoral researcher, and one associate professor to improve the reliability, validity, and quality

of the survey questionnaires. Most of the data collected was quantitative and analyzed using descriptive statistics. Qualitative data was also collected through comments made by a few participants.

Group Interview Study (RQ1-3, RQ5): In a further study, we conducted a group interview study that focuses on examining NFRs, and broader SE and RE in ML in a specific application domain: autonomous perception systems (APS). APS are a critical part of driving automation systems (DAS). These systems rely heavily on ML to perform tasks such as environment perception, object detection and recognition, and situational awareness. **Paper B** and **Paper C** describe the group interview study in detail.

The goal of the study was to explore and examine the SE- and RE-related topics, particularly the treatment of NFRs and challenges faced by practitioners in the development process of ML-based APS. Along with meeting functional requirements, these systems must also achieve demanding quality attributes such as safety, robustness, reliability, and explainability. In this interview study, we aim to understand how these requirements are currently captured, traced, validated, and communicated, and to identify recurring gaps in existing development practices. To explore these questions, we conducted interviews with 19 participants from five automotive companies.

In order to maintain the flexibility to add follow-up questions, we employed semi-structured group interviews with a series of preset open-ended questions. Each session was guided by a series of open-ended questions designed to probe into participants' experiences with RE, SE, and NFRs in ML systems while leaving room for spontaneous elaboration, clarification, and follow-up questions. The interviews lasted between 1 hour 30 minutes and two hours. We used Microsoft Teams to conduct the interviews between December 2021 and April 2022. With all participants' consent, we recorded every interview session. After transcribing, we anonymized the recordings for analysis. At least three researchers were present in each interview session, with the same two researchers participating in all sessions.

We used thematic analysis to analyze the collected qualitative data, inspired by [88, 90]. We used a mixed form of coding. We started with a number of high-level deductive codes, then identified inductive codes while going through interview transcripts. At least three researchers coded each transcript together to ensure reliability and reduce individual bias. The researchers discussed and resolved any disagreements during each coding session. In a second round, a new group of at least two researchers reviewed the interview transcripts and verified the codes. This multi-stage coding and cross-validation approach enhanced the rigor of our analysis and ensured that the resulting themes were well supported by the data.

Preliminary Systematic Mapping Study (RQ4-5, RQ7): We performed an exploratory study to establish an initial scoping of the academic treatment of specific NFRs and an initial estimation of the level of research performed on specific NFRs. This investigation aimed to identify which NFRs have received notable scholarly attention, which have been comparatively underexplored, and how the academic discourse aligns—or fails to align—with practitioner concerns identified in our previous empirical studies. We performed a preliminary

systematic mapping of the selected NFRs for ML systems. We utilized Scopus, a comprehensive meta-database that includes research from peer-reviewed journals and conferences from various publishers, such as IEEE, ACM, and Elsevier. We developed search strings for the database search by identifying relevant terms and synonyms from related literature and our discussions. We split the major terms into more specific terms and concatenated them to form the search strings. The goal of this mapping was understanding the current research landscape related to NFRs for ML systems, highlighting knowledge gaps, and guiding the design of further empirical work and framework development. **Paper D** describes the partial systematic mapping study in detail.

To estimate the number of relevant publications for each selected NFR, we applied a sampling-based relevance screening procedure. We first retrieved the total number of publications returned by each search string and then randomly sampled 50 papers per NFR for manual screening. Three researchers independently evaluated the relevance of each paper based on established inclusion and exclusion criteria. We resolved discrepancies through our discussion, utilizing the inclusion and exclusion criteria to create a final list. We calculated the final estimation by multiplying the total number of identified publications by the percentage of the relevant sample. Although approximate, this estimation provided valuable insights into the relative academic attention devoted to different NFRs in ML contexts.

1.3.2 Artifact Design

Initial Scoping and Clustering (RQ5): Using the result from the mapping study (Paper D) and the interview and survey studies (Paper A), we ask whether ML system NFRs can be grouped into clusters based on shared features, and what scopes (e.g., data, model, system) NFRs can be defined over in an ML system.

We selected important NFRs for ML from the interview study, and defined these NFRs based on our previous experience and a review of literature from research papers, websites, blogs, and forums. To categorize these NFRs into manageable clusters, we employed a group discussion approach to group NFRs that shared similar meanings or purposes. **Paper D** describes the clustering in detail.

We proposed an initial scoping of NFRs for ML systems in **Paper D**. To identify the scope of NFRs for ML systems, we identified the key elements of an ML system. We then utilized our prior definitions and experience, along with the titles and abstracts of relevant studies to determine the applicability of each NFR to these system elements. We improved the initial scoping and proposed the final version in **Paper F**. We first propose that ML systems can be divided into "ML components"—responsible for supporting and performing ML operations—as well as components that interface with ML and components with no relationship to ML. We can then further break down these groups of components. Through a series of meetings, we reached a consensus through discussion, addressing any disagreements by providing concrete examples of how the NFR is applied to the respective ML system element.

A Framework for Managing NFRs for ML Systems (RQ3, RQ5-6): We proposed a quality framework to specify, allocate, measure, and manage NFRs for ML systems. The proposed quality framework consists of five steps—each consisting of one or more concrete tasks. Although the steps are presented in an ordered sequence, practitioners can jump between steps and tasks, as it is expected that NFR identification, definition, scoping, trade-off, measurement, and specification will be iterative in nature. We discuss NFR management at two levels. Steps 1–4 (identifying, prioritizing, defining, scoping, balancing trade-offs between *NFR types*, and measuring NFRs for ML systems) of the framework relate to broad high-level *types* of NFRs (e.g., 'performance"). Step 5 of the framework relates to *specific instances* of those types (e.g., "PE003: the object detection function should detect objects defined in the operational design domain within 0.001 seconds.").

Maintainability Definition, Scoping, and Measurement (RQ5, RQ7): We conducted an artifact-focused study described in **Paper F**. We designed the research to develop a scoping-aware definition of maintainability tailored for ML systems, propose revised modularity metrics that reflect dependencies beyond traditional structured code, and empirically evaluate these metrics on real-world ML systems.

We began by analyzing the limitations of existing definitions and measurements and found that they do not account for heterogeneous components such as datasets, models, or ML scripts. We proposed an updated definition of maintainability that explicitly recognizes ML-specific components and scopes.

Modularity, a sub-characteristic of maintainability, can be measured using coupling and cohesion. We introduced revised coupling and cohesion metrics that capture dependencies between code files, datasets, models, and ML library functions.

1.3.3 Evaluation of the Proposed Artifacts

We evaluated our proposed artifacts and solutions to manage NFRs for ML using different empirical methods such as interviews and open source ML project evaluations. The further evaluation and refinements of our developed artifacts will be done in an iterative process.

Quality Framework Evaluation (RQ6): We conducted semi-structured interviews to collect qualitative data that contained the perceptions of domain experts about our proposed quality framework. A detailed description of the evaluation process is provided in **Paper E**. The sampling method was a mix of convenience and purposive sampling. We contacted people in the industry who have RE and ML experience and then asked them if they knew any qualified candidates we could contact. We interviewed five practitioners who have six to 25 years of experience working with RE and ML. We believe that our interviewees are representative of those working with RE and ML. The interviews lasted 50 to 60 minutes and were conducted between September and October 2023 via Microsoft Teams. We recorded all interviews with the permission of interviewees, then transcribed and anonymized them for further analysis.

We used semi-structured interviews with a set of predetermined open-ended questions so that we could have the freedom to add follow-up questions to collect in-depth information. The collected data were qualitative, and we used thematic analysis as a data analysis method. We used a mixed form of coding, where we started with several high-level codes based on our interview guide, then refined and adapted those codes when we went through the transcripts. One author started to code one transcript, then refined the codes based on feedback from two other researchers, and coded the rest of the transcripts. At last, all the authors discussed the codes and refined those based on the discussion.

Maintainability Measurement Approach Evaluation (RQ7): To evaluate the applicability and usefulness of our proposed maintainability measurement approach for ML systems, we selected ten popular open-source ML projects from GitHub. We selected the projects based on the systems tagged with "ML" and ranking highest by star count. Our selected ML projects span various domains, such as image processing, conversational AI, and facial recognition. We ensured that the selected systems represent realistic, non-trivial ML-based software solutions that include different granular-level components.

We manually decomposed each system into four functional scopes: Data Acquisition, Training Pipeline, ML Interfacing, and Non-ML Components. This decomposition allowed for a scope-aware evaluation of maintainability, based on the assumption that maintainability varies across these heterogeneous components. We applied both traditional maintainability metrics—Coupling Between Object Classes (CBO) and Loose Class Cohesion (LCC)—and our proposed ML-specific extensions— CBO^{ML} and LCC^{ML} over the scopes.

We calculated the metrics per file and then aggregated at the system and scope levels. We used descriptive statistics (i.e., average values and distribution characteristics) to compare CBO, CBO^{ML}, LCC, and LCC^{ML} values between projects and scopes within and across projects. This measurement allowed us to compare how maintainability characteristics differ among systems and among different ML-specific scopes. We also performed qualitative inspection of selected codebases to explore the causes of high or low maintainability scores, focusing on how data, ML models, and code structuring affect the metrics. This mixed-methods evaluation provided empirical support that the revised metrics offer more accurate and context-sensitive insights into the maintainability of ML-based systems than traditional approaches. A detailed description of the evaluation process is provided in **Paper F**.

1.4 Results

In this section, we present the results and answer the research questions based on the research conducted to date. Detailed results can be found in our published research articles: Paper A [91], Paper B [92], Paper C [93], Paper D [94], Paper E [95], and Paper F [96].

RQ1: *What are some of the key RE topics and challenges for ML systems in industry?*

In ML systems, RE involves identifying the problem domain, specifying the system's functional and non-functional requirements, and validating these requirements throughout the development life cycle. We chose autonomous perception systems (APS) as a representative of such ML systems in our study, as ML is an integral part of APS. Based on thematic analysis of the group interview data, RE-related sub-themes, and topics are summarized in Figure 1.3.

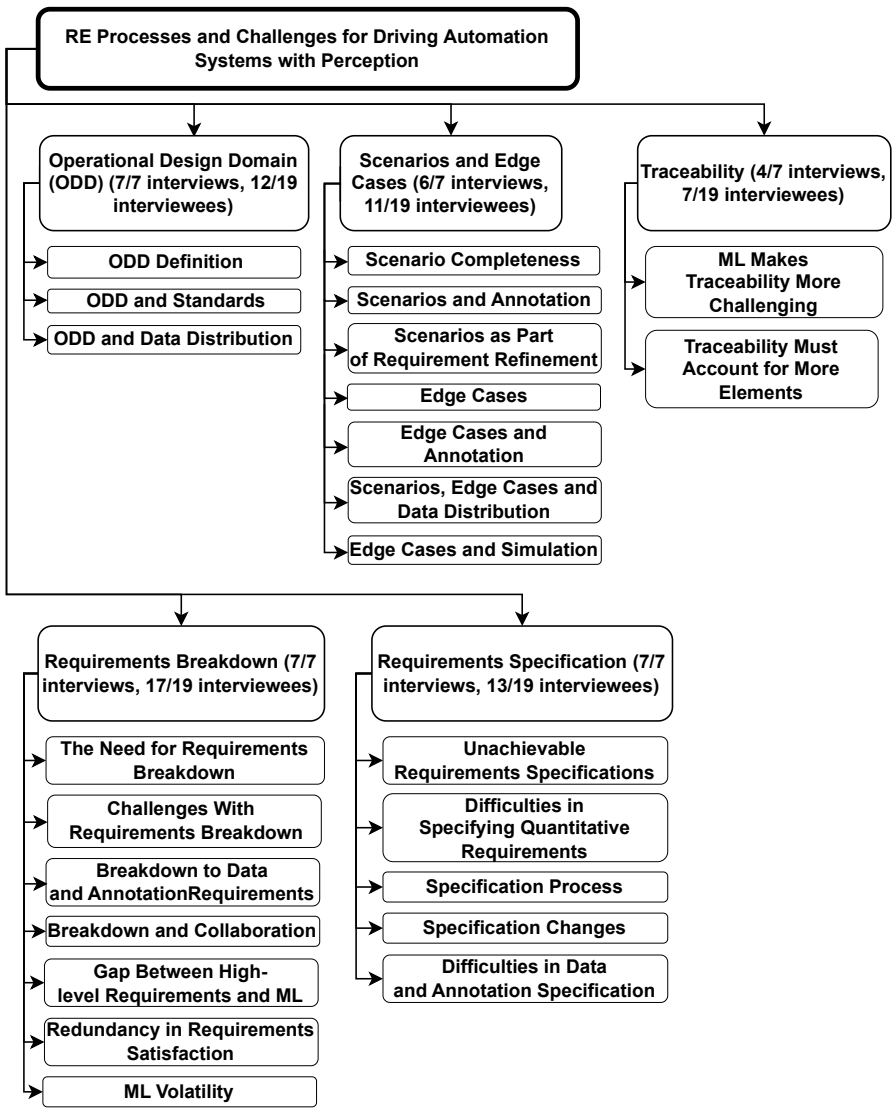

Figure 1.3: Mind map illustrating identified RE topics and challenges for driving automation systems with perception.

Through the group interview study, we identified many RE-related challenges practitioners face in the development of these systems due to the complexity and uncertainty inherent in ML-based perception systems. One recurring theme was the difficulty of defining complete and precise requirements upfront. Instead of conventional requirement specifications, practitioners often rely on representations such as operational design domains (ODDs) and scenario-based modeling as foundational RE artifacts.

Several specific RE challenges emerged in relation to the definition and management of ODDs, the decomposition of high-level goals into detailed, actionable requirements, and the difficulty of tracing these requirements throughout the ML pipeline. RE-related challenges for autonomous perception systems also include detection and exit detection of ODD, the specification of plausible scenarios and edge cases, decomposition of requirements, traceability, quantification of quality requirements, and the creation of specifications for data and annotations. Another key RE task highlighted was the development of clear and testable specifications for both data and annotations—an emerging area of RE artifacts that is critical in ML workflows.

Practitioners also identified important NFRs specific to autonomous perception systems, such as system-level, mentioned performance, comfort, integrity, trust, reliability, robustness, and explainability are the most important NFRs. At the function level, the interviewees mentioned performance, accuracy, and suitability. Quality-related sub-themes and topics are summarized in Figure 1.4. In addition, trade-offs among different NFRs were frequently discussed, such as safety vs. cost, accuracy vs. usability, and cost vs. comfort. These trade-offs highlight the complex balancing acts practitioners must manage, especially when deploying ML systems in safety-critical environments.

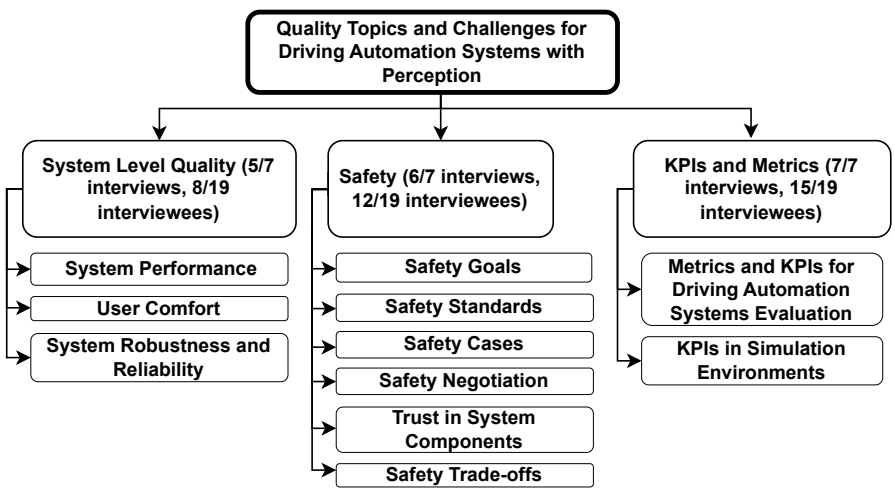

Figure 1.4: Mind map illustrating relevant quality topics and challenges for driving automation systems with perception.

Moreover, large annotated datasets are required for the development of such ML systems, specifically for the training and validation of the ML components. Therefore, maintaining data quality is very important to ensure the overall

quality of autonomous perception systems. Data requirements can also entail specific data quality aspects or data-related NFRs. The most important data quality aspects mentioned by the interviewees do not describe physical properties of data, such as pixel density, contrast, resolution, brightness, etc., but instead focus on the represented information in the data. The important data qualities mentioned by practitioners are bias, data correctness, data reusability, and data maintainability.

On the other hand, collaborations between original equipment manufacturers (OEMs) and their suppliers of software components, data, and annotations are hampered by the widespread challenges in defining data and annotation requirements. The lack of common metrics defining data variance as a way of conveying data quality, the lack of process guidelines, and non-transparent data selection as part of the data gathering process have a negative impact on the ability to specify data needs. The most critical challenges we found are inconsistent manual annotations and missing specifications and guidelines for annotation processes.

Although our study focused on autonomous perception systems, many of the identified RE topics and challenges are likely to be relevant in other ML-intensive domains, particularly where data plays a central role in system behavior. Our findings suggest that traditional RE practices require substantial adaptation to accommodate the uncertainties, dependencies, and emergent properties inherent in ML systems. We suggest practitioners include iterative and continuous requirements refinement, improved methods for specifying and validating data-related requirements, and enhanced tools to support communication and traceability across interdisciplinary teams. Detailed results regarding these topics are discussed in **Paper B** and in **Paper C**.

RQ2: *How are NFRs for ML systems currently measured, and what are the current perceived NFR and NFR measurement-related challenges for ML systems in practice?*

In the interview and survey study as described in **Paper A**, all interviewees stated they measure or need to measure NFRs over ML-enabled software, but the measuring technique varies depending on the functionalities of the software. For example, NFRs can be measured based on response time, statistical analysis, different performance metrics, or user feedback. Measurement can be done by machine and human judgment combined, along with statistical analysis (e.g., precision, recall, and F1 score). According to the interviewees, many NFRs, such as explainability, fairness, and robustness, are difficult to measure, as they are not quantifiable. We asked the interviewees how NFR measurements were captured for ML-enabled systems, such as using a tool or via some documentation. Interviewees were able to name some methods and tools to capture NFR measurements (e.g., checklists, custom code, traceability), but answers varied, and participants often found this question difficult to answer.

Through the interview study, we also gained an understanding of the perceptions and challenges related to NFRs in an ML systems context, described in detail in **Paper A**. We found that practitioners working on ML systems consistently face different challenges in how NFRs should be defined, prioritized, and assessed. Several NFRs were identified as particularly challenging (e.g.,

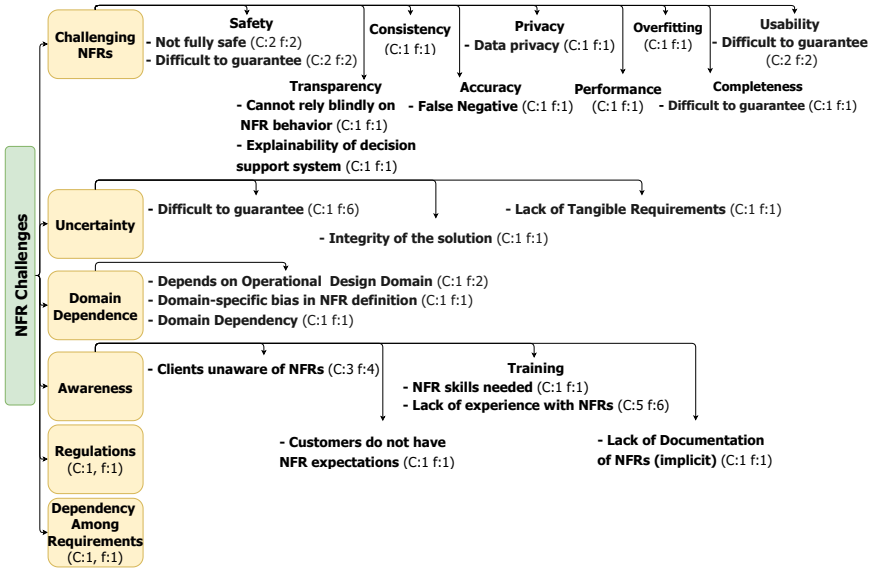

Figure 1.5: NFR-Related Challenges with ML Systems.

safety, transparency, accuracy, consistency, privacy, and completeness), but additional challenges included uncertainty, dependence on domains, and a lack of knowledge of NFRs and regulations. Practitioners also described how the lack of shared understanding or precise definitions—especially for NFRs, such as transparency and robustness—leads to difficulties in translating abstract goals into concrete engineering tasks.

NFR-related challenges for ML systems are presented in Fig. 1.5, where leaf-level challenges include interviewee counts (c) and frequencies (f).

Furthermore, unlike traditional systems, where NFRs can often be directly specified and verified through statistical analysis or formal testing, the dynamic behavior of ML components introduces uncertainty that affects both the elicitation and validation of NFRs for such systems. For instance, the safety and reliability of ML systems depend on the model's accuracy and the quality and representativeness of the data on which the model was trained, which are often hidden or poorly specified. Similarly, achieving consistency or fairness in ML model outputs does not depend solely on code correctness but also depends on how well the training data align with real-world operational scenarios and contexts. Although some of the challenges we identified (e.g., domain dependence, dependency among requirements) are also exist in traditional systems, their impact is amplified in ML systems due to non-deterministic behavior, evolving data distributions, and the tight coupling between data and system behavior.

We also found many challenges regarding the measurement of NFRs in ML systems, including a lack of knowledge, complexity, costly rigorous testing, and finding data. Fig. 1.6 summarizes NFR measurement-related challenges experienced by the interviewees, where leaf-level challenges include interviewee counts (c) and frequencies (f). Although many challenges, such as domain dependence, could apply to both NFR challenges and NFR measurement challenges, the issues discussed here specifically arise during the measurement of NFRs.

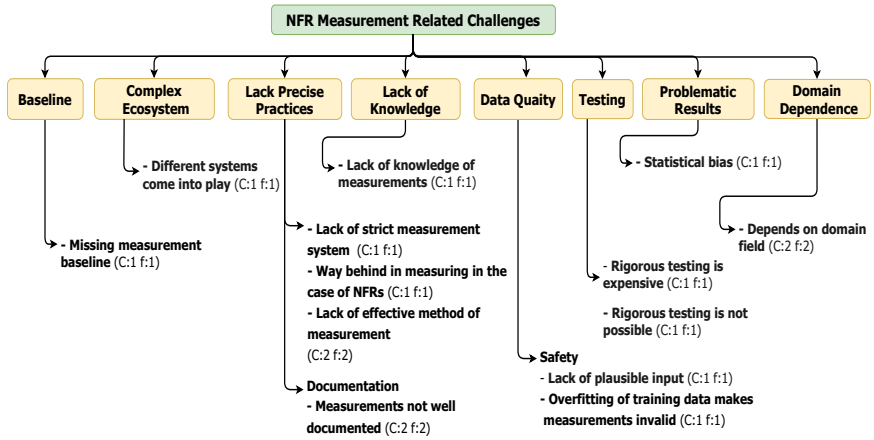

Figure 1.6: NFR Measurement-Related Challenges

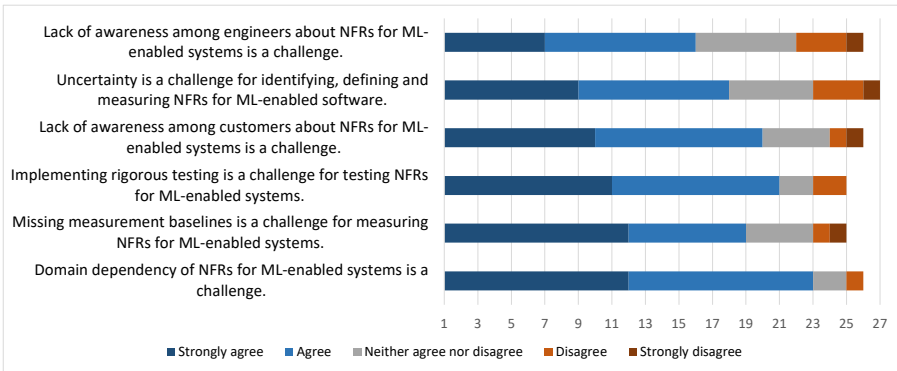

Figure 1.7: Opinions of survey participants on specific NFR and NFR-measurements related challenges.

We also received insights regarding NFR and NFR measurement-related challenges from survey participants. To validate the challenges identified through interviews, we asked survey participants for their opinion on the listed challenges, and the result is presented in Fig. 1.7. Sixteen participants (62%) agreed that lack of awareness among engineers is a challenge, while four (15%) disagreed. Lack of awareness among customers about NFRs is also a challenge—20 participants agreed (77%), while two disagreed (8%). Similarly, we could confirm challenges found in the interviews related to the uncertainty of defining and measuring NFRs for ML systems, the domain dependency of NFRs for ML systems, and implementing rigorous testing of NFRs for ML systems. Most of the participants agreed on these statements, while very few disagreed. Specific challenges may not emerge in all projects. However, 76% of survey respondents have encountered at least one of these challenges in their ML projects.

RQ3: *Which NFRs are more or less important for ML systems than they are for traditional systems?*

Our interview and survey studies offer perspectives on which NFRs are perceived as more or less important for ML systems compared to traditional systems. The identified important and less important NFRs for ML systems are described in detail in **Paper A**. According to the interviewees, most NFRs as defined for traditional software are still relevant and important in an ML context, while only a few become less prominent. Important NFRs, according to our interviews, include fairness, flexibility, usability, accuracy, efficiency, correctness, reliability, and testability. Fig. 1.8 illustrates the important and less important NFRs for ML systems. These NFRs are crucial to ensuring the quality and success of ML systems, particularly in high-stakes or safety-critical domains like autonomous perception systems, where the stakes of model failure are substantial. It is also important to note that there was a disagreement among the interviewees about which NFRs are less important.

For example, some interviewees suggested that portability is less important for ML systems, as ML systems are often developed with specific hardware or runtime environments in mind. Some others considered portability an important NFR, considering that models can be reused across platforms or transferred into new deployment environments. The NFRs (yellow-colored background) in Fig. 1.8 indicate precisely this kind of disagreement, with the same NFR being viewed as important by some and less important by others.

The survey study validated the qualitative insights from the interview study with broader quantitative data. In the survey study, participants strongly agreed that NFRs play an important role in ensuring the quality of ML systems, and there is a difference in how NFRs are defined and measured between traditional systems and ML systems. Participants from a blended context (both academia and industry) placed a higher importance on fairness, transparency, explainability, justifiability, and privacy than other groups. They also placed the highest average importance on NFRs but had the largest variance as well. They placed a lower emphasis on fault tolerance, portability, and simplicity. We also compared the results for those with a more industrial or academic background. For example, accuracy, completeness, integrity, and reliability are the most important NFRs for ML among the academic participants. On the other hand, reliability, accuracy, integrity, and justifiability are the most important NFRs from the industrial participants' perspective.

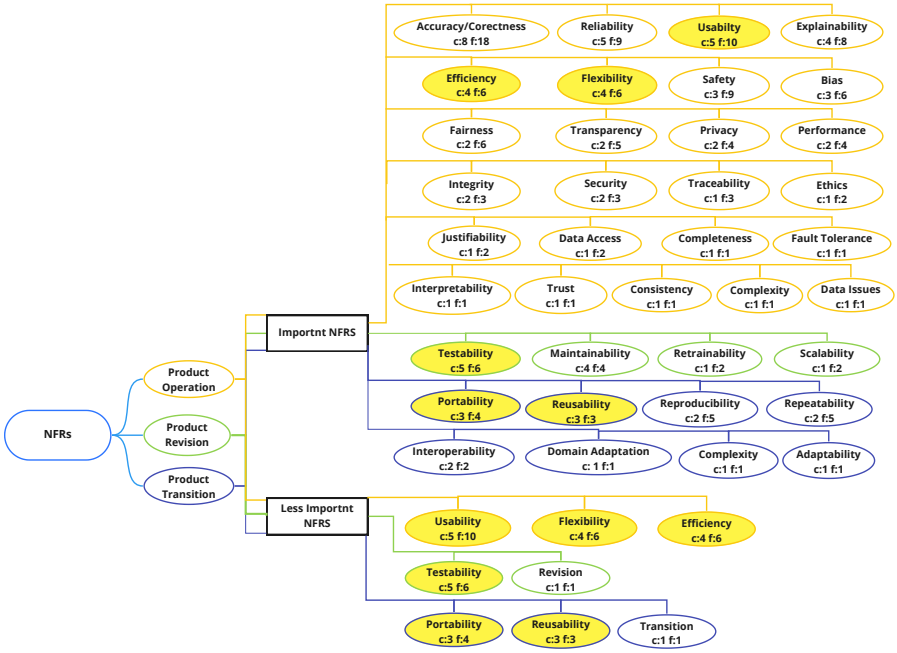

Figure 1.8: Important and Less Important NFRs for ML. **c:** counts of the number of the interviewees whose interview included, **f:** count of occurrences of the code across all transcripts, **Yellow background:** NFRs mentioned by some interviewees as important are identified as less important NFRs by other interviewees.

RQ4: *Which NFRs for ML systems have received the most—or least—attention in existing research literature?*

We conducted a literature search in the Scopus database to estimate the number of relevant publications on each of the selected NFRs for ML. The number of identified publications is presented in the second column of Table 1.2. We found that performance, accuracy, efficiency, security, complexity, privacy, and safety received the most attention in research. In contrast, retrainability, justifiability, testability, repeatability, traceability, and maintainability got the least number of publications. The detailed result is described in **Paper D**. The number of papers for accuracy is very high since researchers and practitioners are particularly interested in prediction accuracy. We also found more papers for usability than we expected, even when excluding papers using usability as a synonym for applicability, and find it encouraging that research is focusing on human-oriented aspects. Even while practitioners in the interview and survey study (**Paper A**) noted retrainability as an important NFR for ML systems, we were surprised that no literature was found for retrainability.

Table 1.2: NFRs with number of search results, number of relevant publications, kappa values (agreement on sample), and final paper volume estimation for select NFRs. We only examined a second sample in cases where we wanted to see if agreement would improve.

NFR	Search Results	Relevant (1)	Kappa (1)	Relevant (2)	Kappa (2)	Est. Pubs.
Performance	114853					
Accuracy	92669					
Efficiency	22247					
Security	19142					
Complexity	16997					
Privacy	6388					
Safety	5848					
Reliability	5620					
Bias	4118					
Scalability	3595					
Consistency	2936					
Flexibility	2764	23 (46%)	0.54			1271
Interpretability	2418					
Trust	1965					
Reproducibility	1796					
Domain Adapt.	1732	47 (94%)	0.63			1628
Usability	1270	21 (42%)	0.50	29 (58%)	0.44	635
Adaptability	1177	34 (68%)	0.50			800
Fairness	1089	45 (90%)	0.41			980
Correctness	1045	16 (32%)	0.53			334
Integrity	1015					
Transparency	851	44 (88%)	0.70			749
Explainability	706	44 (88%)	0.22			621
Fault Tolerance	553	26 (52%)	0.68			288
Interoperability	532	9 (18%)	0.45			96
Completeness	372	23 (46%)	0.40	25 (50%)	0.58	179
Portability	346	21 (42%)	0.45			145
Ethics	331	31 (62%)	-0.03			205
Reusability	321	24 (48%)	0.55			154
Maintainability	277	6 (12%)	0.30	9 (18%)	0.72	42
Traceability	214	4 (8%)	0.61	6 (12%)	0.61	21
Repeatability	171	17 (34%)	0.44			58
Testability	77	4 (8%)	0.54	2 (4%)	1.00	5
Justifiability	3	0 (0%)	1.00			0
Retrainability	0					0

RQ5: *Over what aspects of an ML system are NFRs defined and measured?*

Our results show that while traditional systems often associate NFRs with the system as a whole or at the module and component level, NFRs in ML systems present new challenges due to the presence of ML pipeline and the heterogeneous nature of such systems.

The interview study result described in (**Paper A**) shows that the participants expressed uncertainty in clearly identifying the scopes for both NFR specification and measurement. We observed inconsistency in how different practitioners interpreted and applied scoping of NFRs over different elements of ML systems. Some practitioners explicitly defined NFRs (e.g., performance or robustness) over ML models, while others framed them at the system level, especially when discussing safety, usability, or integrity. We also note that this question was not so easy to answer for many participants. We see even more disagreement on the scope of measurement than on the scope of NFR definition, with still a slight focus on measuring over the model rather than the data or whole system.

From the survey study described in (**Paper A**), we found that most

practitioners (72%) focused on defining NFRs over the whole system. While many interviewees and some survey respondents (17%) also define NFRs on models, a few practitioners (11%) have explicitly considered NFRs for ML-related data. Almost all respondents (93%) agreed that NFR measurements for ML systems are dependent on the context, while one participant added that measurement for NFRs in ML is dependent on the domain. For the statement, "NFR measurements for ML-enabled systems can be dependent on another NFR defined for the other parts of the same system, the whole system, the ML model, or the data," we received 26 responses; among them, 85% of the participants agreed with the statement , while one disagreed (4%) and three gave neutral responses (12%).

RQ6: *What structured approaches or framework(s) can support the identification, scoping, specification, and management of NFRs for ML systems?*

ML System Scoping: In **Paper D**, we performed an exploratory scoping of selected NFRs in terms of which elements of the system they can be defined and measured over (e.g., training data, ML algorithm, ML model, or results). To illustrate our determinations, we select several examples. For example, NFR usability can be defined over the ML algorithm, the ML model, the results, and the whole system; but may not be applicable over the training data. If we take the simple definition of usability from **Paper D**, "how effectively users can learn and use a system," this definition makes sense over the whole system. We can also define this NFR over specific ML elements. The usability of an ML algorithm depends on how effectively users can learn and use it to train an ML model as part of a system. The usability of an ML model is how effectively users learn to use an ML model at run-time to get results. The usability of the ML results is how effectively users can understand and apply ML results for some practical purpose. However, we struggled to create a definition for the usability of the training data. Does a user learn data? Although a user uses data, is some data more usable than others, or is that more a matter of data quality and data appropriateness?

To explore NFR scoping in greater depth, a more comprehensive and systematic treatment of NFR scoping is done and presented in **Paper F**. We argue that ML systems should not be viewed as monolithic entities but as assemblies of heterogeneous components with distinct roles, formats, and behaviors. We first propose that ML systems can be divided into "ML components"—responsible for supporting and performing ML operations—as well as components that interface with ML and components with no relationship to ML. These groups of components can then be broken down further.

In Fig. 1.9, we present typical components of an ML system based on supervised learning. However, it could be extended for unsupervised and reinforcement learning. The ML components include the trained models that perform predictions, as well as the data and training pipeline used to train and tune those models. ML-interfacing components include code that loads, invokes, and monitors the models. There is also data ingested at runtime, as well as code components that deliver traditional forms of functionality, separate from ML. ISO/IEC 23050:2022 also provided a breakdown of the elements of an ML system [97]. While the ISO/IEC 23053:2023 standard provides a high-level

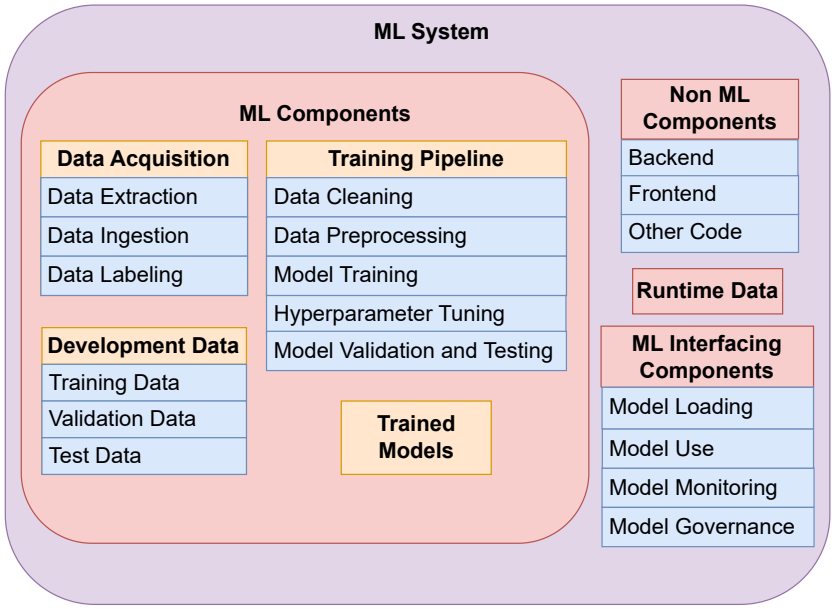

Figure 1.9: ML system components, separated into groups.

functional framework for the lifecycle of AI systems using machine learning, emphasizing phases such as data preparation, model learning, deployment, and operation—our decomposition adopts a system-level architectural perspective.

When scoping NFRs, the "type" of components contained within a prospective scope may influence specification and measurement. In Fig. 1.10, we also break down these components into four types of information represented, including data—input data to either the training process or to the system at runtime—trained models, structured code, and scripting.

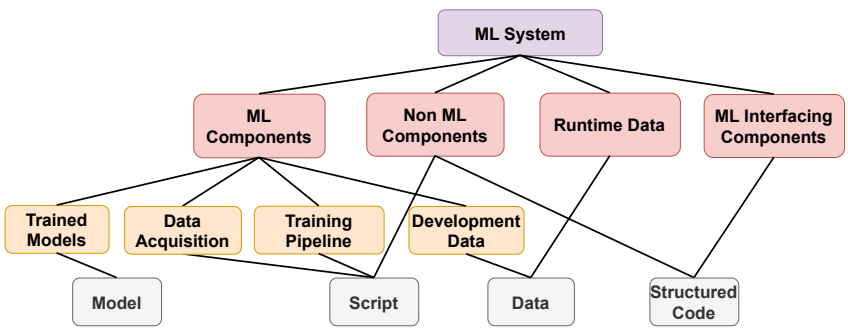

Figure 1.10: Components, grouped by the type of information represented.

In summary, our findings across empirical and conceptual studies indicate

that NFRs in ML systems cannot be uniformly defined or measured over a single scope or only over the whole system. Instead, the scoping must be explicit, granular, and tailored to the different system components.

A Framework for Managing NFRs for ML Systems: Toward addressing NFR-related challenges, we propose a *framework* to help practitioners manage NFRs for ML systems. The details of the framework can be found in **Paper E**. This framework, consisting of five steps, guides NFR specification from a high level (Steps 1–4)—identifying, prioritizing, defining, scoping, balancing trade-offs between *NFR types* (e.g., accuracy), and measuring—to template-based specification (Step 5) of low-level individual *specific NFRs* (e.g., "The lane identification model must have an accuracy of 99.99%"). Each of the five steps of our proposed NFR management framework consists of one or more concrete tasks—presented in Fig. 1.11. Although the steps are presented in an ordered sequence, practitioners can jump between steps and tasks, as it is expected that NFR identification, definition, scoping, trade-off, measurement, and specification will be iterative in nature.

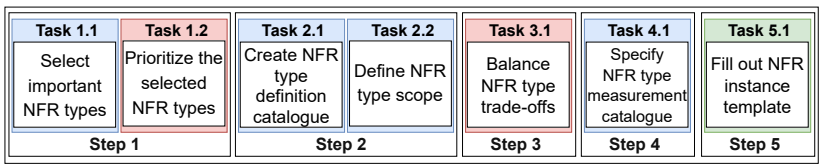

Figure 1.11: Overview of the framework. Tasks in blue are performed on NFR types, tasks in red compare NFR types, and tasks in green are performed on specific NFR instances.

As a first step (Task 1.1 and Task 1.2), practitioners are advised to identify and prioritize NFR types that are important for their context. By selecting important NFR types, practitioners can ensure research allocation (time, effort, and budget) more effectively, mitigate risks, prevent potential issues that could arise later, and make informed design decisions that prioritize aspects of system architecture, infrastructure, and implementation that contribute most significantly to meeting performance objectives. Practitioners can identify and prioritize important NFRs on an ad hoc basis or using established requirements engineering techniques. In Step 2 (Task 2.1), we recommend that practitioners create a definition catalogue by adapting the general definitions of the prioritized important NFRs to their particular system. The adjustment is needed as the definition of an NFR type may vary depending on context. Furthermore, in Task 2.2, we recommend practitioners to scope NFR types over different elements of ML systems. An ML system is typically a monolith; some components are related directly to ML, while others do not. Scoping is important, as the definition, importance, and measurement of NFR types may differ depending on the components considered. Practitioners may observe trade-offs among different NFR types—e.g., performance, usability, and security often conflict. In Step 3, we recommend practitioners balance trade-offs between conflicting NFR types, highlighting that these conflicts are often inevitable and context-dependent. In Step 4, we suggest practitioners specify an NFR measurement catalogue for the prioritized NFRs for their system. Practitioners can adapt

a measurement catalogue for the selected NFR types. Having a clear and comprehensive measurement catalogue is crucial, as it provides a systematic method to identify how to assess the attainment of each NFR. In the final step, we suggest practitioners fill out a template as part of the process of specifying the low-level individual NFRs of the types selected and defined in the previous steps. To uphold the overall system quality, each NFR type may require the fulfillment of many specific NFRs. It is important to have stable, unambiguous, and complete NFR documentation so that stakeholders can form a shared understanding of the NFR and ensure that the NFR is satisfied.

We summarize how the elements of the framework address the challenges in NFR management identified through literature studies and our interview studies in Table 1.3. For example, "new" NFR types like fairness can be discovered and considered as part of Task 1.1—where important NFR types are identified—with the help of a list of potentially relevant NFRs.

Table 1.3: Mapping between NFR challenges for ML systems and our framework.

Challenges	**Framework Aspect**
New NFR types are needed [13, 14, 91]	Task 1.1
NFR types change in priority for systems with or without ML [91, 94]	Task 1.2
NFR type definitions need to be adjusted [14, 91]	Task 2.1
NFRs type definitions need to be scoped over components [14, 91, 94]	Task 2.2
Trade-offs between NFR types are important and unclear [51, 91, 98]	Step 3
Difficulty measuring attainment of different types of NFRs [91]	Step 4
Lack of documentation [91]	Step 5
Lack of management guidance, practices, and knowledge [14, 91]	All
Lack of management solutions (e.g., frameworks or tools) [15, 51, 91]	All

We use autonomous perception systems—part of driving automation systems (DAS)—as a running example to describe our proposed framework. Autonomous perception systems use ML, trained on complex sensor data, to identify and analyze objects within a vehicle's environment. We preliminarily evaluated our proposed framework through four interviews with five practitioners. This evaluation demonstrates the potential of this framework and also provides feedback for future revisions.

RQ7: *How to develop ML-specific measurements of NFRs for ML systems?*

Maintainability Definition, Scoping, and Measurement for ML Systems: In **Paper F**, we focus on maintainability—and, specifically, modularity—as an example of how NFR definition and measurement can be adapted to ML systems. In Fig. 1.9, we present a breakdown of ML systems into common components that can be used to scope measurements and requirements. We have also introduced a modified definition of maintainability, along with metrics to assess cohesion and coupling, which take into account both structured and unstructured code, as well as dependencies on models and data.

We propose that the ISO/IEC 25010 and 25059 definition of maintainability [19, 99] should be adapted to consider this breakdown:

Maintainability refers to the degree of effectiveness and efficiency with which modifications, including corrections, adaptations, or improvements, can be applied to the ML system as a whole or to scopes (individual components

or subsets of components) within the system.

ML systems consist of complex heterogeneous components. Maintainability of ML systems is impacted by both individual components and subsets of components in ways that are still unclear. Though this is a simple change textually, we suggest that maintainability—as well as its five sub-characteristics (modularity, reusability, modifiability, analyzability, and testability)—must be reconsidered in a way that reflects a scoping-aware view of an ML system, as different scopes may have unique maintenance needs and challenges.

Modularity, a sub-characteristic of maintainability, plays a crucial role in software design and has a significant impact on maintainability and on other NFRs. In traditional software systems, modularity is typically assessed using coupling and cohesion. However, existing metrics and measurement approaches only consider structured code, potentially overlooking the diversity of components and interactions found in modern systems. As such, new or revised measurements and metrics may be needed at different scopes to capture interactions between different subsets and types of components. Two concepts are often used in quantitative modularity metrics—cohesion and coupling [100–102]. Coupling refers to the degree of interdependence between code-based components in a system, while cohesion refers to the degree to which grouped sub-components—e.g., functions collected within a single class—belong together [103]. Both concepts indicate how easily a system can be understood, modified, and extended. High coupling indicates low modularity, as components are more interdependent [104]. Similarly, low cohesion indicates low modularity, as components lack focused responsibility [105]. One of the most common coupling metrics is **"Coupling Between Object Classes" (**CBO**)** [106]. One of the most common cohesion metrics is **Loose Class Cohesion (**LCC**)** [107].

Modified Coupling Measurement: We introduce a modified form of the CBO metric, CBO^{ML}, that differs from the original in the following ways:

- Rather than calculating the metric over each class in the project, which assumes structured code, we calculate one value for each distinct code file in the project.
- Rather than the number of classes that the code file-under-assessment is coupled to, we count: (1) the number of other code files that the file-under-assessment is coupled to, and (2), the number of other data files or models that the file-under-assessment is coupled to.

Coupling between structured code and scripting is determined based on a reference to a variable or method in another code file. In our current implementation, coupling between code and external data files is determined by detection of invocations of `read`, `write`, and `load` functions referencing a particular filename[2]. CBO^{ML} can be calculated using the following formula:

$$CBO^{ML}(C) = \sum_{i=1}^{c} I(C, C_i) + \sum_{j=1}^{d} I(C, D_j) \tag{1.1}$$

[2]Our concrete implementation is based on Python, but could be adapted to similar functions in other languages.

Where C is the code file-under-assessment, c is the total number of code files in the system, and d is the total number of data or model files in the system. $I(C, C_i)$ and $I(C, D_j)$ are equal to 1 if C is coupled to the code or data/model file and 0 otherwise. We can take the average CBO^{ML} as an indicator of the modularity of the project as a whole.

Modified Cohesion Measurement: LCC considers a class to be cohesive if many of its methods read from or modify the same variables within the class. We introduce a modified version of LCC, which we refer to as LCC^{ML}:

- Rather than calculating for each class, we calculate LCC^{ML} for each code file.
- In addition to considering two methods to be cohesive if they have direct or incorrect connections to the same variables, we consider methods to be cohesive if they access the same data files or models.
- We also consider two methods to be cohesive if they invoke the same functions from a common ML library[3]. Many ML systems depend on such libraries, and methods contained in the same code file that use the same library functions are likely to be related in their purpose.

LCC^{ML} can be calculated using the following formula:

$$LCC^{ML}(C) = \frac{M_v \cup M_f \cup M_l}{\frac{n(n-1)}{2}} \tag{1.2}$$

Where M_v is the number of method pairs that are directly or indirectly connected by sharing at least one attribute, M_f is the number of method pairs that are directly or indirectly connected by accessing at least one common file, M_l is the number of method pairs that are directly or indirectly connected by accessing the same library function, and n is the total number of methods in the class. As with the traditional formula, LCC^{ML} values range between 0 and 1, with 1 indicating maximum cohesion. When $n = 0$—i.e., a class contains no methods—LCC^{ML} is undefined.

Our sample implementations perform static analysis to detect dependencies on code and data files. To select representative systems, we have searched for systems in GitHub with the "ML" tag and chosen the 10 systems with the most stars—that is, the most popular on GitHub.

Table 1.4 presents the average CBO^{ML} for each of the 10 ML systems, where the colored cells indicate an increase from the traditional metric, with the magnitude of the increase in parentheses.

Table 1.5 presents the average LCC^{ML} for each of the 10 systems. The average LCC for six of the systems are greater than 0.5, indicating that the systems tend to be relatively highly cohesive.

We hypothesize that LCC^{ML} is more accurate than LCC for assessing cohesion within ML systems, as it considers methods to be cohesive through sharing the same variables, data files, models, or library functions, and not just variables. Like with CBO, the magnitude of the increase indicates that cohesion through traditional code-based mechanisms is more common than cohesion

[3]The specific libraries are listed at `https://anonymous.4open.science/r/maintainability-for-MLSystems-3C12/cohesion_improved.py`.

Table 1.4: Average CBO^{ML} for each system and scope. Colored cells indicate difference from CBO.

System	Whole System	Data Acquisition	Training Pipeline	ML Interfacing	Non-ML
face recognition	2.32	✖	2.50	2.65	1.00
faceswap	46.16	✖	43.09	50.24	41.72
Open Assistant	15.25 (0.33%)	10.67 (2.79%)	28.61 (0.32%)	12.02	11.81 (0.08%)
DeepFaceLive	22.47	17.00	35.00	27.18	27.18
CLIP	1.60	2.00	2.00	1.50	1.00
EasyOCR	11.54 (0.26%)	13.00	11.62	12.58	✖
DocsGPT	8.80 (0.80%)	15.00	11.11 (9.89%)	7.96 (0.51%)	12.19
Chatterbot	8.91 (0.56%)	✖	13.75	11.47	12.83
DeepFace	24.94	✖	27.39	28.97	10.56
LaTeX-OCR	6.08	4.00	6.71	6.00	6.00
Average	**14.81 (0.14%)**	**10.28 (0.49%)**	**18.18 (0.61%)**	**16.06 (0.06%)**	**13.81**

Table 1.5: Average LCC^{ML} for each system and scope. Colored cells indicate difference from LCC.

System	Whole System	Data Acquisition	Training Pipeline	ML Interfacing	Non-ML
face recognition	0.70	✖	0.50	0.05	0.00
faceswap	0.69 (3.79%)	✖	0.64 (5.83%)	0.59 (4.82%)	0.61
Open Assistant	0.45 (5.95%)	0.64 (0.95%)	0.44 (9.75%)	0.20 (3.16%)	0.41 (1.25%)
DeepFaceLive	0.66 (2.00%)	0.54	0.97	0.87	0.88
CLIP	0.54	0.00	0.38	0.00	1.00
EasyOCR	0.55 (17.45%)	0.33 (51.36%)	0.52 (15.11%)	0.37 (133.75%)	✖
DocsGPT	0.49	0.00	0.65	0.20	0.59
Chatterbot	0.87	✖	0.21	0.57	0.73
DeepFace	0.53 (5.40%)	✖	0.39 (5.95%)	0.23 (13.00%)	0.68 (5.78%)
LaTeX-OCR	0.49 (6.15%)	0.00	0.40 (12.86%)	0.03 (180.00%)	0.49
Average	**0.60 (3.41%)**	**0.25 (8.35%)**	**0.51 (3.98%)**	**0.31 (10.53%)**	**0.60 (0.66%)**

through data files or library functions. However, LCC^{ML} still highlights dependencies that are missed by the traditional metric.

Overall, the values of CBO^{ML} and LCC^{ML} are higher than CBO and LCC because the revised metrics capture dependencies on data and library functions missed by traditional metrics. However, we see minimal differences in coupling between CBO and CBO^{ML} and a small—but more distinct—increase in LCC^{ML}, compared to LCC. This indicates that code-based dependencies remain more common than data or library-based dependencies. However, the revised metrics still highlight dependencies missed by traditional metrics, especially with regard to cohesion.

1.5 Threats to Validity

Construct Validity: In the work presented in **Paper A**, in terms of construct validity, some of our interviewees asked for examples of NFRs because they were not familiar with either the concept or terminology of NFRs. One potential explanation for this is that the interviewees are representative of the data science and ML field and may not have formal software engineering expertise. As a result, they might not be familiar with specific terms or concepts in software engineering. To exemplify NFRs, we showed a version of McCall's software quality hierarchy [108]. We could have used other available NFR hierarchies, as there are many. However, we chose this example because of its prominence in RE literature.

The questions concerning how NFR measurements were captured were difficult for the interviewees to understand. Therefore, they might have interpreted and understood each NFR differently. In retrospect, this question could have been written more clearly. Still, we believe that the collected results were interesting. In addition, we see that several survey respondents had experience with RE and NFRs of less than a year, so some questions may have been confusing to them because they were unfamiliar with the terminology. To reduce this threat, as part of each question, we included short definitions of terms. In the survey introduction, we also provided a description of the survey context and definitions of terms.

In terms of construct validity, in **PaperF**, we focused only on modularity, one sub-characteristic of maintainability among five— specifically on cohesion and coupling measurements. However, modularity is one of the most critical sub-characteristics of maintainability, and cohesion and coupling are the most common means of assessing modularity. Although these metrics are widely used, some studies have found that they are too narrowly focused on code-level dependencies, rendering them insufficient for characterizing maintainability [109,110].

External Validity: In the interview and survey study presented in Paper A, we had a large number of respondents from the Nordic countries, even though our participants came from different parts of the world. However, we found participants from a wide range of product domains, and we believe that the Nordic countries have a strong and international AI-oriented industry. Thus, our participants are fairly representative of the software development industry as a whole.

In **Paper D**, we have only used Scopus, which may mean that we might miss relevant articles in other databases. However, Scopus is a meta-database

that is rich in content on computer science research from multiple publishers. We searched for articles in Scopus up to September 2021, and there may be newer papers that were missed.

In the studies described in **Paper A**, **Paper B**, **Paper C**, and **Paper E**, we used a combination of purposive and snowball sampling to find participants. As our study needed a certain set of expertise to answer the research questions, we could not perform a random sampling. Still, due to the size of the study, with participants covering a wide variety of roles with varying experience levels and covering different companies, we believe our participants are fairly representative of the software development industry as a whole. Furthermore, we deliberately chose not to link participants to specific interviews or companies to protect their anonymity. Although this may affect the transferability of our results, we feel that this level of anonymity does not greatly hurt our results. We argue that we reached a sufficient point of saturation with our interview data, as we noticed a sharp decline in emerging codes after analyzing the last few interviews.

Though the results of our study in **Paper B** and **Paper C** are limited to autonomous perception systems in DAS, we argue that some findings can be applied to other safety-critical or perception systems. DAS represents ML systems that integrate heterogeneous components such as sensors, data pipelines, trained models, and embedded software—and relies heavily on large-scale data for perception and decision-making. Similar factors are found in other domains, such as robotics, industrial automation, and medical devices. Consequently, RE and SE practices and challenges identified in DAS, such as NFR management, traceability, and V&V complexity, are expected to also apply in these domains. Applicability to a wider variety of systems outside of the embedded and safety-critical domains should be explored in future studies.

In **Paper E**, the number of interview participants may affect the reliability of the evaluation. In addition, three out of five interviewees were from the automotive domain. However, we selected the interviewees based on deep knowledge of both NFRs and ML, and their suggestions are applicable to other domains. Importantly, the goal of this evaluation was simply to gain preliminary feedback.

Although the focus of the framework has been on ML systems, as mapped to findings in this work, it is likely that elements of the framework could be generally useful for managing NFRs for all types of systems. However, our focus was on ML systems.

In **Paper F**, we focused on maintainability as a representative NFR as maintainability is an important software quality attribute that is widely studied, important for ML systems, and can be measured using cohesion and coupling. Demonstrating an approach to measure maintainability for ML systems in an ML specific way provides a methodological foundation for the measurement of other NFRs. However, we acknowledge that not all NFRs behave similarly. For example, NFRs such as safety, security, or performance often require context-specific definitions and domain-dependent evaluation methods. Consequently, our focus on maintainability offers a starting point for NFR measurement in ML systems, its generalization to all NFRs should be validated in future work.

For evaluating our proposed maintainability measurement, we analyzed ten ML systems. This sample size was selected to balance the coverage of the

domain with practical constraints on time and resources. We chose systems based on popularity (GitHub stars). However, many of these systems are based on the use of images as a data source. Thus, this subset may not represent the full diversity of ML systems, including types of ML (e.g., we do not currently consider unsupervised or reinforcement learning) or application domains. Still, we believe this selection is adequate for illustrating and preliminarily evaluating our proposed component breakdown and metrics, while also offering direction for future research.

Additionally, our evaluation focused exclusively on ML systems implemented in Python. However, Python is currently the most popular language for ML system development, and the proposed measurements and component breakdown are not language-specific.

Internal Validity: In **Paper A**, **Paper B**, and **Paper C**, we applied thematic coding that may suffer from internal validity threats. Although qualitative coding always comes with some bias, we mitigated this threat by following established literature [90], performing independent coding over half the interviews and comparing results, finding sufficient agreement for **Paper A**; coding in multiple rounds, using inductive and deductive codes, and having multiple authors participate in each round of coding, with in-depth discussion on code meanings for **Paper B** and **Paper C**.

In **Paper A**, **Paper B**, and **Paper C**, we conducted a pilot interview and conducted an internal peer review of the interview guide to improve the guide and procedure. All interview participants received an email from us outlining the details and aim of the interview study. We can consider whether our interview findings were close to reaching saturation. We found towards the end of our analysis that the codes were generally converging to a stable set but did not reveal any new results. Thus, we believe further interviews could help to enrich our findings, but would not produce significant additions.

In **Paper A**, our sampling technique for the interview and survey study found several participants straddle the boundaries between industry and academia. This may be a result of our circle of contacts and reflective of the practitioners interested in responding to the studies. However, we also believe that those who are interested in the topics covered in this paper are often mid- to upper-level management and often have a strong academic or research-oriented background. Another potential issue is that the length of the survey may have discouraged people from participating. However, we sent the survey questionnaire to three other researchers to test whether they understood the questions before widely distributing the survey. We changed the wording and reduced the number of questions according to their suggestions.

In **Paper C**, there is potential bias in determining paper inclusion, and we defined shared inclusion criteria, each of the authors examined each title and abstract separately, and we made a collective decision in cases of disagreement to mitigate this risk.

The clusters we created in **Paper C** may be subjective to our experiences and opinions. NFRs could be arranged differently, but we believe our clusters provide a suitable foundation for organizing and guiding future research. Our evaluation of the NFR definition's scope may also be subjective. We made these judgments in agreement between all authors, discussing difficult cases.

We have tried to justify our selection for a sample of NFRs.

In **Paper F**, our maintainability metric implementations are based on static analysis and could miss dependencies, leading to potentially misleading values. However, we performed manual tests on two projects to ensure the accuracy of the measurements.

1.6 Summary of Contributions

This thesis contributes a set of empirical insights and artifacts with a focus on the identification, definition, and management of NFRs for ML systems. The contributions span empirical findings and concrete artifacts that support both research and industry practice. Table. 1.6 presents a mapping from the contributions of this thesis to the addressed RQs, and to the corresponding published papers. The studies were generally designed to answer specific RQs. However, some studies generated side contributions beyond their primary objectives. For example, the SE practices and challenges for ML systems in Paper B and the in-depth analysis of data-related specification, challenges, and quality management in Paper C emerged as supplementary results while addressing RQ1.

Table 1.6: Mapping among the contributions of this thesis, RQs, and published papers.

Contributions	**RQs**	**Papers**
A list of important NFRs for ML systems	RQ3, RQ4	A, B
A comprehensive set of NFR and NFR measurement-related challenges for ML systems	RQ2	A, B, C
Insights into RE practices and challenges for ML systems	RQ1	B
Insights into SE practices and challenges for ML systems	Supplementary results	B
In-depth analysis of data-related specification, challenges, and quality management	Supplementary results	C
NFR clustering and scoping over ML system elements	RQ5,RQ7	D, E, F
A quality framework for managing NFRs in ML systems	RQ6	E
Maintainability metrics specific to ML systems	RQ7	F

The major contributions of the thesis are:

- **A list of important NFRs for ML systems:** We identified important NFRs for ML systems based on extensive empirical data from interviews, surveys, and group studies involving ML practitioners across different sectors. Our findings included how traditional NFRs shift in relevance in the ML context and revealed the emergence of new NFRs such as fairness, explainability, retrainability, and justifiability. We also identified some NFRs (e.g., portability) that may be less important for ML systems. Our results open an opportunity for further research to be done on those NFRs with a newly increased focus in an ML context, e.g., fairness, explainability, transparency, bias, justifiability, and testability.
- **A comprehensive set of NFR and NFR measurement-related challenges for ML systems:** We identified NFR-related challenges—such as uncertainty, domain dependence, challenging regulations, and lack

of awareness among practitioners. We also identified NFR measurement-related challenges, including missing measurement baseline, complex ecosystem, expensive rigorous testing, unclear metrics, and lack of knowledge among practitioners and stakeholders. These findings serve as a reference point for researchers aiming to mitigate NFR challenges for ML systems and help practitioners benchmark their own organizational objectives.

- **Insights into RE practices and challenges for ML systems:** We identified RE-related topics and sub-topics for ML-based autonomous perception systems. Although perception systems have been the primary focus of this work, many of the RE practices and issues would be more broadly applicable to other areas that rely on ML. Practitioners experience RE challenges related to uncertainty, ODD detection, realistic scenarios, edge case specification, traceability, creating specifications for data and annotations, and quantifying quality requirements. We also collected quality requirements (NFRs) at different system levels. At the function level, the interviewees mentioned performance, accuracy, and suitability. We also identified quality trade-offs, such as safety vs. cost, accuracy vs. usability, and cost vs. comfort. By summarizing the views and challenges of different experts on RE for ML-enabled perception systems, our results are valuable for practitioners working to advance this area. Additionally, our findings contribute to improving RE knowledge more broadly in other domains reliant on ML. The results of this study offer guidance to practitioners and suggest future research directions in the intersection of requirements engineering, software quality, development methodologies, and machine learning to help mitigate the challenges practitioners are facing.

- **Insights into SE practices and challenges for ML systems:** We explored how the integration of ML influences software and systems engineering practices in the development of ML-based autonomous perception systems. Our findings reveal that traditional and agile methodologies are insufficient on their own for large-scale ML development, leading practitioners to adopt hybrid approaches that combine top-down engineering with iterative, bottom-up workflows supported by continuous feedback cycles. Effective V&V depends heavily on data selection, quality assessment, and in some cases the use of synthetic data, which raises additional quality and realism challenges. These results provide actionable insights for practitioners adapting engineering processes to the unique demands of ML-driven perception systems and contribute to the broader understanding of SE in ML-enabled domains. Although this work primarily focuses on perception systems, many of the SE practices and challenges are likely relevant to other domains that rely on ML.

- **In-depth analysis of data-related specification, challenges, and quality management:** Recognizing that ML systems are fundamentally data-driven, we investigated the challenges of specifying, selecting, and annotating data in industrial ML development. We found that unclear data collection processes, a lack of metrics for data quality, inconsistent

annotation practices, and poor transparency in annotation workflows all hinder effective requirements specification. Furthermore, we investigated current practices in the business environment and ecosystems deployed in the automotive industry, especially concerning a new trend toward emphasizing joint development projects over the traditional OEM supplier relationship in data-intensive developments. We provided several recommendations to the practitioners based on our observations. The results of our study suggest a number of further research topics: the problem of defining clear metrics for data quality and annotation aspects and how partners can agree on proper metrics is not solved.

- **NFR clustering and scoping over ML system elements:** We created six clusters of important NFRs for ML systems based on shared characteristics and meanings. These clusters provide practitioners with simplified views of which NFRs are closely related and which are unique. These clusters also facilitate informed decision-making about which NFRs to prioritize, define, and measure during development. In Paper D, we further introduce the notion of preliminary NFR scope—that is, the parts of an ML system (e.g., model, data, pipeline, result) to which a given NFR applies. In Paper F, we revised the scoping and introduced a breakdown of ML systems into typical components that could be used to scope requirements and measurements. This scoping concept helps clarify where in the system each NFR is defined and measured, and supports trade-off decisions between conflicting NFRs.

- **A quality framework for managing NFRs in ML systems:** We proposed a framework for managing NFRs for ML systems, which includes structured steps for identifying, prioritizing, defining, scoping, measuring, and specifying NFRs for ML systems. This framework can help practitioners and researchers to systematically deliver high-quality ML systems by offering step-by-step guidance for defining, prioritizing, scoping, measuring, and specifying NFR types and specific NFRs during the development process. Although our proposed framework could be useful for NFR management in any type of system, the challenges specific to ML systems indicate that this type of guidance and management is particularly needed, and the framework has been designed with these ML system challenges in mind.

- **Maintainability metrics specific to ML systems:** We proposed a component-based breakdown of ML systems, introduced a scoping-aware definition of maintainability, and ML-specific measurements of modularity, tailored to these ML components. Modularity is often measured for traditional systems using coupling and cohesion. A major contribution of this paper is the adaptation of traditional modularity metrics—specifically, coupling (CBO) and cohesion (LCC)—into new metrics (CBO^{ML} and LCC^{ML}) that include dependencies not only within structured code but also between code, data files, and ML models. We investigated how our proposed breakdown and adapted metrics can be used to assess the maintainability of 10 real-world ML systems. We found that our breakdown

is applicable, and that the modified metrics capture dependencies missed by traditional metrics. The contributions and observations in this paper offer a starting point for future research on measurement and improving the maintainability of ML systems. Our proposed scoping can also be applied to other NFRs important for ML systems.

1.7 Future Work

We gathered early feedback on our developed artifact – the quality framework— using interviews with the practitioners working with RE and ML. Based on the input from the domain experts, we will refine our proposed quality framework and perform a further evaluation in practice using a case study. In addition, we plan to conduct further evaluation of our quality framework using an interview and/or survey study. We also plan to develop a rigorous NFRs definition catalogue and NFRs measurement catalogue specific to ML systems as a part of the framework that will pose features such as NFR measurement techniques, tools, measurement baselines, measurement capturing techniques, measurement challenges, and so on. Furthermore, our future work includes framework tooling. A tool based on our framework could provide users with suggestions for potentially incomplete or overlooked aspects (e.g., trade-offs, measurements, templates), thereby offering a form of quality or completeness check. In addition, we will explore how our proposed framework will facilitate compliance with the EU AI Act [111].

Another important direction of future work involves evaluating our measurements and scoping on further projects and other sub-characteristics of maintainability. We plan to consider additional metrics in future work. In addition, our current approach looks only at code modularity, not at the modularity of data, models, or other parts of ML systems. Future studies should consider adapted or new measurements for those components and for other sub-characteristics of maintainability and other NFRs, which are important for ML systems.

1.8 Conclusion

This PhD thesis focuses on addressing the challenges related to NFRs and managing NFRs in the development process of ML systems. ML systems are composed of diverse and interdependent components, such as data pipelines, ML models, and ML interface components, which can influence the overall system quality differently compared to traditional systems. These unique characteristics, along with the non-deterministic nature of ML systems, introduce challenges in defining, measuring, and managing NFRs specific to ML systems. To tackle these issues, the research develops a structured approach to NFR management, grounded in empirical studies and guided by design science research methodology. At first, we conducted an interview and survey that identified important NFRs along with NFR- and NFR measurement-related challenges for ML systems. We also conducted an exploratory study and part of a systematic mapping study, where we clustered important NFRs based on shared characteristics, identified the initial scope of defining and measuring

NFRs for ML, and identified important NFRs for ML that are less explored in research. Furthermore, we conducted a group interview study and identified RE practices and challenges in ML-enabled autonomous perception systems.

In the solution space, we proposed a structured five-step framework for managing NFRs in ML systems. The framework supports NFR identification, prioritization, defining, scoping, trade-off analysis, measurement planning, and template-based specification. The framework was validated through interviews with practitioners from different domains. Finally, we proposed a breakdown of ML systems into granular-level components as well as a revised definition and measurements of maintainability that take into account these components. We focused on modularity, a sub-characteristic of maintainability, as an example of how NFR definition and measurement can be adapted to ML systems. We adapted traditional metrics and revised them to assess cohesion and coupling, considering both structured and unstructured code, as well as dependencies on models and data.

Overall, this thesis contributes to a deeper understanding of how NFRs can be managed in ML systems by addressing key challenges in their definition, scoping, and measurement. The proposed framework, along with its supporting insights, serves as a bridge between traditional software engineering practices and the evolving needs of ML system development.